



systems

Special Issue Reprint

Advancements in the Practical Applications of Agents, Multi-Agent Systems and Simulating Complex Systems

Edited by

Philippe Mathieu, Juan M. Corchado, Alfonso González-Briones
and Fernando De la Prieta Pintado

mdpi.com/journal/systems



Advancements in the Practical Applications of Agents, Multi-Agent Systems and Simulating Complex Systems

Advancements in the Practical Applications of Agents, Multi-Agent Systems and Simulating Complex Systems

Editors

Philippe Mathieu

Juan M. Corchado

Alfonso González-Briones

Fernando De la Prieta Pintado



Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Editors

Philippe Mathieu
Department of Computer
Science
Lille University
Villeneuve d'Ascq
France

Juan M. Corchado
Department of Computer
Science and Automation
University of Salamanca
Salamanca
Spain

Alfonso González-Briones
Department of Computer
Science and Automation
University of Salamanca
Salamanca
Spain

Fernando De la Prieta Pintado
Department of Computer
Science and Automation
University of Salamanca
Salamanca
Spain

Editorial Office

MDPI
St. Alban-Anlage 66
4052 Basel, Switzerland

This is a reprint of articles from the Special Issue published online in the open access journal *Systems* (ISSN 2079-8954) (available at: www.mdpi.com/journal/systems/special_issues/paams22).

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. <i>Journal Name</i> Year , <i>Volume Number</i> , Page Range.
--

ISBN 978-3-0365-9309-8 (Hbk)

ISBN 978-3-0365-9308-1 (PDF)

doi.org/10.3390/books978-3-0365-9308-1

© 2023 by the authors. Articles in this book are Open Access and distributed under the Creative Commons Attribution (CC BY) license. The book as a whole is distributed by MDPI under the terms and conditions of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) license.

Contents

About the Editors	vii
Preface	ix
Philippe Mathieu, Juan Manuel Corchado, Alfonso González-Briones and Fernando De la Prieta Advancements in the Practical Applications of Agents, Multi-Agent Systems and Simulating Complex Systems Reprinted from: <i>Systems</i> 2023 , <i>11</i> , 525, doi:10.3390/systems11100525	1
Alfredo Guzmán Rincón, Ruby Lorena Carrillo Barbosa, Nuria Segovia-García and David Ricardo Africano Franco Disinformation in Social Networks and Bots: Simulated Scenarios of Its Spread from System Dynamics Reprinted from: <i>Systems</i> 2022 , <i>10</i> , 34, doi:10.3390/systems10020034	7
Yuanjian Ye, Renjie Zhang, Yiqing Zhao, Yuanyuan Yu, Wenxin Du and Tinggui Chen A Novel Public Opinion Polarization Model Based on BA Network Reprinted from: <i>Systems</i> 2022 , <i>10</i> , 46, doi:10.3390/systems10020046	19
Ismo T. Koponen Agent-Based Modeling of Consensus Group Formation with Complex Webs of Beliefs Reprinted from: <i>Systems</i> 2022 , <i>10</i> , 212, doi:10.3390/systems10060212	37
Aiwu Zhao, Jingyi Wang, Zhenzhen Sun and Hongjun Guan Research on the Evolutionary Path of Eco-Conservation and High-Quality Development in the Yellow River Basin Based on an Agent-Based Model Reprinted from: <i>Systems</i> 2022 , <i>10</i> , 105, doi:10.3390/systems10040105	55
Jang Won Bae and Il-Chul Moon Practical Formalism-Based Approaches for Multi-Resolution Modeling and Simulation Reprinted from: <i>Systems</i> 2022 , <i>10</i> , 174, doi:10.3390/systems10050174	79
Ziyan Wang and Tianjian Yang Multi-Category Innovation and Encroachment Strategy Evolution of Composite E-Commerce Platform Based on Multi-Agent Simulation Reprinted from: <i>Systems</i> 2022 , <i>10</i> , 215, doi:10.3390/systems10060215	95
Manuel Castañón-Puga, Ricardo Fernando Rosales-Cisneros, Julio César Acosta-Prado, Alfredo Tirado-Ramos, Camilo Khatchikian and Elías Aburto-Camaclanqui Earned Value Management Agent-Based Simulation Model Reprinted from: <i>Systems</i> 2023 , <i>11</i> , 86, doi:10.3390/systems11020086	119
Athanasia Karalakou, Dimitrios Troullinos, Georgios Chalkiadakis and Markos Papageorgiou Deep Reinforcement Learning Reward Function Design for Autonomous Driving in Lane-Free Traffic Reprinted from: <i>Systems</i> 2023 , <i>11</i> , 134, doi:10.3390/systems11030134	139
Nikolaos I. Spanoudakis, Charilaos Akasiadis, Georgios Iatrakis and Georgios Chalkiadakis Engineering IoT-Based Open MAS for Large-Scale V2G/G2V Reprinted from: <i>Systems</i> 2023 , <i>11</i> , 157, doi:10.3390/systems11030157	167

Jonatan J. Gómez Vilchez and Roberto Pasqualino The Hidden Side of Electro-Mobility: Modelling Agents' Financial Statements and Their Interactions with a European Focus Reprinted from: <i>Systems</i> 2023 , <i>11</i> , 132, doi:10.3390/systems11030132	195
Jörg Bremer and Sebastian Lehnhoff Enhancing Local Decisions in Agent-Based Cartesian Genetic Programming byCMA-ES Reprinted from: <i>Systems</i> 2023 , <i>11</i> , 177, doi:10.3390/systems11040177	213
Peter Atrazhev and Petr Musilek It's All about Reward: Contrasting Joint Rewards and Individual Reward in Centralized Learning Decentralized Execution Algorithms Reprinted from: <i>Systems</i> 2023 , <i>11</i> , 180, doi:10.3390/systems11040180	235
Miguel Pincheira, Elena Donini, Massimo Vecchio and Raffaele Giaffreda An Infrastructure Cost and Benefits Evaluation Framework for Blockchain-Based Applications Reprinted from: <i>Systems</i> 2023 , <i>11</i> , 184, doi:10.3390/systems11040184	251
Ahmad Esmaeili, Zahra Ghorrati and Eric T. Matson Agent-Based Collaborative Random Search for Hyperparameter Tuning and Global Function Optimization Reprinted from: <i>Systems</i> 2023 , <i>11</i> , 228, doi:10.3390/systems11050228	265
Stéphanie Roussel, Gauthier Picard, Cédric Pralet and Sara Maqrot Conflicting Bundle Allocation with Preferences in Weighted Directed Acyclic Graphs: Application to Orbit Slot Allocation Problems Reprinted from: <i>Systems</i> 2023 , <i>11</i> , 297, doi:10.3390/systems11060297	287

About the Editors

Philippe Mathieu

Philippe Mathieu received his Ph.D. degree in computer science from the University of Lille, France, in 1991. He is now full professor at the University of Lille. He is the team leader of the SMAC team in CRISAL Lab, UMR 9189 CNRS since 22 years. His research focuses on Artificial Intelligence, Multi-agent Systems, Computational game theory and Computational economics. He is the author of numerous publications and communications on these research fields.

Juan M. Corchado

Juan Manuel Corchado (15 May 1971, Salamanca, Spain) is Professor at the University of Salamanca, Director of the BISITE Research Group (Bioinformatics, Intelligent Systems and Educational Technology) and President of the AIR Institute. He is also the Director of the IOT Digital Innovation Hub, President of the AIR Institute, and Principal Investigator of the Project funded by DIGIS3 (Digital Innovation Hub of Castilla y León). He is also Visiting Professor at the Osaka Institute of Technology and Visiting Professor at the Universiti Malaysia Kelantan.

Juan M. Corchado has been Vice-Rector for Research from 2013 to 2017 and Director of the Science Park of the University of Salamanca. He was elected twice as Dean of the Faculty of Science, has been president of the IEEE Systems, Man and Cybernetics association, and academic coordinator of the University Institute for Research in Art and Animation Technology of the University of Salamanca, as well as researcher at the Universities of Paisley (UK), Vigo (Spain), and the Plymouth Marine Laboratory (UK). He has also been a member of the Advisory Group on Online Terrorist Propaganda of the European Counter Terrorism Centre (EUROPOL) and has been a visiting professor at the University of Technology Malaysia.

Alfonso González-Briones

Alfonso González-Briones earned a Ph.D. in Computer Engineering in 2018 at the University of Salamanca; his thesis obtained the second place in the 1st SENSORS+CIRTI Award for the best national thesis in Smart Cities (CAEPIA 2018). At the same University, he had also obtained a Bachelor of Technical Engineering in Computer Engineering (2012), a Bachelor's Degree in Computer Engineering (2013), and a Master's Degree in Intelligent Systems (2014). Since 2014, Alfonso González Briones has worked in different public research centres, such as the BISITE Research Group, and at Complutense University of Madrid as a "Juan De La Cierva" Postdoc. Moreover, he has worked at private research centres, such as Virtual Power Solutions (VPS) in Coimbra, (Portugal), where he was involved in research on decision tree algorithms with a high percentage of efficiency in the recognition of devices that consumed energy, as well as parameters that helped to improve the accuracy of recognition and detection; in the ISLab Research Group (Synthetic Intelligence Lab) of Universidade do Minho, Braga (Portugal), he was involved in a research on the methodologies and algorithms that achieve energy efficiency through the use of Artificial Intelligence techniques. He has also worked as Project Manager in Industry 4.0 and IoT projects in the AIR Institute, and Lecturer at the International University of La Rioja (UNIR).

Currently, Alfonso González Briones is Associate Professor at the University of Salamanca in the Department of Computer Science and Automation.

Fernando De la Prieta Pintado

Fernando de la Prieta Pintado is Associate Professor at the University of Salamanca Department of Computer Science and Automation, where he currently is Deputy Secretary-General of the University of Salamanca. Dr. De la Prieta is equally well experienced in research and teaching, as evident in his curriculum.

Over the past years, he has followed a clearly defined line of research, focusing on the integration of multi-agent organisations, machine learning and advanced architectures in different fields. He applied the resulting knowledge toward both his doctoral thesis (for which he obtained an international PhD mention and an extraordinary PhD award) and in the projects he has been involved in. He has more than 50 publications in international journals, many of which have a JCR impact factor on the Web of Science database. His H index in Google Scholar is 27. Furthermore, he has published more than 100 articles in books and in the proceedings of prestigious international conferences, around thirty of these publications have been published in conferences indexed according to the CORE ranking. He has worked on more than 90 research projects (16 of them were international and in several he has been the principal investigator). In addition, he has participated in more than 30 research contracts (Art. 83), in some of them as the principal investigator. As a result of his work, around 40 intellectual properties have been registered. He has performed several stays abroad (pre- and post-doctoral) in Portugal, Japan and South Korea. He has also taken an active part in the organisation of international conferences, some of them included in the CORE ranking: IEEE-GLOBECOM (core B), ICCBR (Core B), CEDI, PAAMS (core C), ACM-SAC (core B), IEEE-FUSION (core C), and others.

Preface

The relationship between individuality and aggregation is an important topic in complex systems science, as both aspects are facets of emergence. This problem has generally been addressed by adopting a classical individual- versus population-level approach in which boundaries emerge in segregated communities. More specifically, boundaries delimiting and interconnecting aggregates are at play. It is, therefore, crucial to define the properties of complex systems correctly, such as generic agent-based models, with which to simulate communities situated in grid- and scale-free network environments. To do this, complexities may be resolved through simulation, modeling and analysis techniques, which help provide confidence regarding the behavior of such systems, especially of those operating in dynamic environments or under unexpected constraints. Moreover, modeling and simulation help reduce the risks and costs involved in the design and development of validation tests.

Understanding the emergent behaviors of complex systems, and ensuring their correct performance in different environments, will allow for their evolution, as well as that of the methodologies integrated in them.

Although various methodologies are being used for the development of Complex Systems Simulation, one of the most widely adopted approaches is based on the agent paradigm, which may be used to create simulations for dynamic continuous time systems and discrete event systems. Agent-based systems may be applied in all sorts of areas, including home automation, industry, smart cities and automotive sectors.

This Special Issue invited researchers to submit original, quality studies regarding the domain of Complex Systems Simulation and urged them to address its main sub-disciplines.

**Philippe Mathieu, Juan M. Corchado, Alfonso González-Briones, and Fernando De la Prieta
Pintado**
Editors

Editorial

Advancements in the Practical Applications of Agents, Multi-Agent Systems and Simulating Complex Systems

Philippe Mathieu ¹, Juan Manuel Corchado ^{2,3,4}, Alfonso González-Briones ² and Fernando De la Prieta ^{2,*}

¹ Department of Computer Science, Lille University, Cité Scientifique, 59650 Villeneuve-d'Ascq, France; philippe.mathieu@univ-lille.fr

² BISITE Research Group, University of Salamanca, 37007 Salamanca, Spain; jm@corchado.net or corchado@usal.es (J.M.C.); alfonso@usal.es (A.G.-B.)

³ Air Institute, IoT Digital Innovation Hub, 37188 Salamanca, Spain

⁴ Department of Electronics, Information and Communication, Faculty of Engineering, Osaka Institute of Technology, Osaka 535-8585, Japan

* Correspondence: fer@usal.es; Tel.: +34-677-522-678

Introduction

This Editorial summarizes the content of the Special Issue entitled *Advancements in The Practical Applications of Agents, Multi-Agent Systems and Simulating Complex Systems*, published in the “Complex Systems” section of *Systems* (ISSN 2079-8954).

Complex systems have played a fundamental role in the simulation, modeling, and analysis of information in dynamic environments and under unexpected constraints [1–3]. These agent-based systems have evolved significantly throughout history, providing increasingly sophisticated solutions to address the complex challenges encountered across multiple fields. The history of complex systems dates back to early research in systems theory and cybernetics in the 1940s [4]. These disciplines laid the foundation for understanding and addressing problems involving complex and emergent interactions between multiple components. As computer technology advanced, the first agent-based modeling and simulation approaches emerged, allowing complex systems to be represented through the interaction of multiple autonomous entities [5,6].

At the heart of complex systems are agents, which can be individuals, organizations, robots, or any entity with the ability to make decisions and respond to its environment. These agents interact with each other and with their environment, generating emergent patterns and collective behaviors that cannot be attributed solely to the individual characteristics of the agents. Agent-based systems technology has been advancing rapidly, enabling greater sophistication in the representation and simulation of complex systems [7]. The importance of complex systems lies in their ability to address real-world problems in a wide range of disciplines, including economics, biology, ecology, logistics, and supply chain management, among others. These systems can model and simulate complex phenomena such as crowd behavior, traffic flow, the spread of disease, climate change, and the evolution of ecosystems [8–10].

The simulation and modeling of complex systems offer several significant advantages. First, they enable the evaluation of different scenarios and strategies without incurring the costs and risks associated with real-world implementation. This is especially valuable when dealing with unpredictable or highly complex environments wherein it is difficult to obtain empirical data or conduct controlled experiments [11]. Complex systems provide a deeper understanding of the underlying mechanisms and interactions that shape the system being studied. This helps to identify emerging patterns, hotspots, and non-intuitive behaviors, which in turn can guide decision-making and strategic planning. By better understanding complex systems, it is possible to reduce the risks and costs associated with the design and development of real-world validation tests [12–14].

Citation: Mathieu, P.; Corchado, J.M.; González-Briones, A.; De la Prieta, F. Advancements in the Practical Applications of Agents, Multi-Agent Systems and Simulating Complex Systems. *Systems* **2023**, *11*, 525. <https://doi.org/10.3390/systems11100525>

Received: 6 September 2023

Accepted: 19 October 2023

Published: 21 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

An Overview of Published Articles

This Special Issue consists of fifteen practical papers covering key topics in the field of multi-agent systems and complex systems. These articles, presented during the 20th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS'22) (<https://www.paams.net>, 6 September 2023), are noteworthy for their highly innovative results and trends [15]. The conference was held in L'Aquila, Italy, and it was organized by the University of L'Aquila (Italy), Umeå University (Sweden), the University of Lille (France), and the University of Salamanca (Spain)

The first three articles address the relevance of using multi-agent systems for the analysis of information. Such an analysis is carried out through the application of natural language processing techniques. These articles aimed to understand public opinion, detect false information, or identify accounts that provide misleading information. Guzmán Rincon et al. (Contribution 1) present a mathematical model to simulate scenarios of disinformation propagation in social networks caused by bots, trolls, and others. The authors carried out simulations related to the increase in the rate of the activation and deactivation of disinformation agents and the disinformation caused by this mechanism. Ye et al. (Contribution 2) explored the specific attributes of individuals and opinion network nodes by incorporating parameters such as individual conformity and the strength of individual online relationships for the purpose of identifying an online opinion polarization of a group. Through simulations, the authors found that individual conformity and the difference in environmental attitude greatly influence the trajectory of opinion polarization events. Similarly, the analysis of shared beliefs, opinions, and views in groups is a topic of great interest that has been debated in sociology, political science, communication, and organizational science. Koponen (Contribution 3) performed an analysis of consensus group formation through an agent-based model. Agents' views were described as complex, and they have extensive structures, similar to semantic networks, i.e., belief networks. In the agent-based model presented by the author, the agents' interactions and their participation in the sharing of their views depend on the similarity of the agents' belief webs; the higher the similarity, the more likely the interaction and the sharing of webs of belief elements.

In the areas of economics, finance, and e-commerce, complex systems have also had a major impact. Zhao et al. (Contribution 4) present an agent-based model created using empirical data from a number of cities as sample data to simulate the evolutionary trajectory of eco-protection and high-quality development under different policy scenarios, such as green innovation, ecological constraints on the environment, ecological compensation, etc. The model shows how, depending on the existing development model, the economic development of cities will be subject to different degrees of ecological and resource constraints and that different policy scenarios significantly affect the evolutionary trends of economic development. Other authors, such as Bae et al. (Contribution 5), introduce a formalism or multi-resolution translational discrete event system specification (MRT-DEVS) intended to facilitate the implementation of simulations and reduce simulation execution costs. MRT-DEVS embeds state and event translation functions into the model's specifications so that it enables multi-resolution modeling with less complex mechanisms in terms of operations. Wang et al. (Contribution 6) studied the product encroachment behavior of composite e-commerce platforms with double-differentiated multi-product competition and constructed a game model of product innovation by an independent seller and product encroachment by the platform owner. Using multi-agent simulation, the authors simulated the bounded rational decision-making and interaction process of multiple agents in multiple periods and analyzed the influence of the main parameters. Moreover, Castañón-Puga et al. (Contribution 7) illustrate how earned value management (EVM) is an efficient method for measuring a project's performance by comparing actual progress against planned activities, thus facilitating the formulation of more accurate predicted estimations using an agent-based simulation model.

Researchers have also focused on applying complex system algorithms to facilitate problem solving in the field of transport. Karalakou et al. (Contribution 8) propose the

design of autonomous vehicles using deep reinforcement learning and the combination between various reward components that are able to gradually learn effective policies in environments with different levels of difficulty, especially when all the proposed reward components are appropriately combined. Spanoudakis et al. (Contribution 9) have designed an open system for the V2G/G2V power transfer problem domain using an agent-based architecture involving flexible microservices that are interconnected via an IoT platform. Gómez Vilchez et al. (Contribution 10) describe a simulation model that facilitates the analysis of potential emission penalties in the broader context of the financial position of original equipment manufacturers. Through their simulation, the authors aim to understand the channels through which money flows (e.g., to promote R&D in cleaner vehicles and to finance zero-emission powertrain sales) between market players.

On the other hand, agent systems have demonstrated successful performance in the application of Cartesian genetic programming to solve a series of use cases, such as complete enumeration in local agent decisions. In this context, Bremer et al. (Contribution 11) present the adaptation of a distributed optimization heuristic protocol for Cartesian genetic programming and an extension using CMA-ES (Covariant Matrix Adaption Evolution Strategy) to improve local agent decisions. By decomposing the evolution on an algorithmic level, it becomes possible to distribute the nodes and regard the evolution process as a parallel, asynchronous execution of an individual coordinate's descent.

Atrazhev et al. (Contribution 12) address the issue of choosing an appropriate reward function in multi-agent reinforcement learning. Among the traditional approaches to employing joint rewards for team performance, this one is questioned because of its lack of theoretical support. Thus, the authors explore the impact of changing the reward function from joint to individual on learning centralized–decentralized execution algorithms in a level-based foraging environment. The results show that different algorithms are affected differently, with value factorization and proximal policy optimization (PPO)-based methods taking advantage of the increased variance to achieve better performance. This study sheds light on the importance of considering the choice of a reward function and its impact on multi-agent reinforcement learning systems.

Within the area of optimization, Pincheira et al. (Contribution 13) present a framework for evaluating the infrastructure costs and benefits of blockchain applications. The framework includes a taxonomy that classifies relevant transactions, a model to evaluate the infrastructure costs and application benefits using public or private blockchains, and guidance on how to use the model. Another research work focusing on optimization comes in the form of the paper by Esmaelii et al. (Contribution 14). The authors of this paper introduce an agent-based collaborative technique for finding near-optimal values for any arbitrary set of hyperparameters (or decision variables) in a machine learning model (or a blackbox function optimization problem). The developed method forms a hierarchical agent-based architecture for the distribution of the searching operations at different dimensions and employs a cooperative searching procedure based on an adaptive width-based random sampling technique to locate the optima.

Finally, within this Special Issue, Roussel et al. (Contribution 15) address the issue of conflicting bundle allocation and weighted directed acyclic graphs. The authors propose several models for novel resource allocation problems where agents express their preferences over conflicting bundles of items as edge-weighted on a directed acyclic graph (directed path allocation problem, or DPAP), particularizing conflicts on vertices (V-DPAP) and conflicts on resources (R-DPAP). The multi-agent system proposed by the authors allows for the search of path allocation. Conflicting bundle allocation and weighted directed acyclic graphs are also commonly simulated using complex systems.

Conclusions

This Special Issue showcases a variety of research papers on practical approaches to the use of complex systems and complementary agent-based AI models, facilitating the parallel use of data treatment and knowledge processing algorithms.

List of Contributions

1. Guzmán Rincón, A.; Carrillo Barbosa, R.L.; Segovia-García, N.; Africano Franco, D.R. Disinformation in Social Networks and Bots: Simulated Scenarios of Its Spread from System Dynamics. *Systems* **2022**, *10*, 34.
2. Ye, Y.; Zhang, R.; Zhao, Y.; Yu, Y.; Du, W.; Chen, T. A Novel Public Opinion Polarization Model Based on BA Network. *Systems* **2022**, *10*, 46.
3. Koponen, I.T. Agent-Based Modeling of Consensus Group Formation with Complex Webs of Beliefs. *Systems* **2022**, *10*, 212.
4. Zhao, A.; Wang, J.; Sun, Z.; Guan, H. Research on the Evolutionary Path of Eco-Conservation and High-Quality Development in the Yellow River Basin Based on an Agent-Based Model. *Systems* **2022**, *10*, 105.
5. Bae, J.W.; Moon, I.-C. Practical Formalism-Based Approaches for Multi-Resolution Modeling and Simulation. *Systems* **2022**, *10*, 174.
6. Wang, Z.; Yang, T. Multi-Category Innovation and Encroachment Strategy Evolution of Composite E-Commerce Platform Based on Multi-Agent Simulation. *Systems* **2022**, *10*, 215.
7. Castañón-Puga, M.; Rosales-Cisneros, R.F.; Acosta-Prado, J.C.; Tirado-Ramos, A.; Khatchikian, C.; Aburto-Camaclanqui, E. Earned Value Management Agent-Based Simulation Model. *Systems* **2023**, *11*, 86.
8. Karalakou, A.; Troullinos, D.; Chalkiadakis, G.; Papageorgiou, M. Deep Reinforcement Learning Reward Function Design for Autonomous Driving in Lane-Free Traffic. *Systems* **2023**, *11*, 134.
9. Spanoudakis, N.I.; Akasiadis, C.; Iatrakis, G.; Chalkiadakis, G. Engineering IoT-Based Open MAS for Large-Scale V2G/G2V. *Systems* **2023**, *11*, 157.
10. Gómez Vilchez, J.J.; Pasqualino, R. The Hidden Side of Electro-Mobility: Modelling Agents’ Financial Statements and Their Interactions with a European Focus. *Systems* **2023**, *11*, 132.
11. Bremer, J.; Lehnhoff, S. Enhancing Local Decisions in Agent-Based Cartesian Genetic Programming by CMA-ES. *Systems* **2023**, *11*, 177.
12. Atrazhev, P.; Musilek, P. It’s All about Reward: Contrasting Joint Rewards and Individual Reward in Centralized Learning Decentralized Execution Algorithms. *Systems* **2023**, *11*, 180.
13. Pincheira, M.; Donini, E.; Vecchio, M.; Giaffreda, R. An Infrastructure Cost and Benefits Evaluation Framework for Blockchain-Based Applications. *Systems* **2023**, *11*, 184.
14. Esmaeili, A.; Ghorrati, Z.; Matson, E.T. Agent-Based Collaborative Random Search for Hyperparameter Tuning and Global Function Optimization. *Systems* **2023**, *11*, 228.
15. Roussel, S.; Picard, G.; Pralet, C.; Maqrot, S. Conflicting Bundle Allocation with Preferences in Weighted Directed Acyclic Graphs: Application to Orbit Slot Allocation Problems. *Systems* **2023**, *11*, 297.

Author Contributions: P.M., J.M.C., A.G.-B. and F.D.I.P. worked together throughout the entire editorial process of this Special Issue entitled “Advancements in The Practical Applications of Agents, Multi-Agent Systems and Simulating Complex Systems”, published by the *Systems* Journal. A.G.-B. and F.D.I.P. drafted this editorial summary. P.M., F.D.I.P. and J.M.C. reviewed, edited, and finalized the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the project “Coordinated intelligent Services for Adaptive Smart areaS (COSASS)”, Reference: PID2021-123673OB-C33, financed by MCIN/AEI/10.13039/501100011033/FEDER, UE.

Acknowledgments: First and foremost, we would like to thank all the researchers who submitted articles to this Special Issue for their excellent contributions. We are also grateful to all the reviewers who helped in the evaluation of the manuscripts and made very valuable suggestions to improve the quality of the contributions. We would like to acknowledge the editorial board of *Systems*, who invited us to guest edit this Special Issue. We are also grateful to the *Systems* Editorial Office staff, who

worked thoroughly to maintain the rigorous peer-review schedule and ensure the timely publication of this Special Issue.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guessoum, Z. Adaptive agents and multiagent systems. *IEEE Distrib. Syst. Online* **2004**, *5*. [[CrossRef](#)]
2. Wood, M.F.; DeLoach, S.A. An overview of the multiagent systems engineering methodology. In Proceedings of the International Workshop on Agent-Oriented Software Engineering, Limerick, Ireland, 10 June 2000; pp. 207–221.
3. Boes, J.; Migeon, F. Self-organizing multi-agent systems for the control of complex systems. *J. Syst. Softw.* **2017**, *134*, 12–28. [[CrossRef](#)]
4. Wiener, N. Cybernetics. *Sci. Am.* **1948**, *179*, 14–19. [[CrossRef](#)] [[PubMed](#)]
5. Panait, L.; Luke, S. Cooperative multi-agent learning: The state of the art. *Auton. Agents Multi-Agent Syst.* **2005**, *11*, 387–434. [[CrossRef](#)]
6. Wooldridge, M.; Jennings, N.R.; Kinny, D. The Gaia methodology for agent-oriented analysis and design. *Auton. Agents Multi-Agent Syst.* **2000**, *3*, 285–312. [[CrossRef](#)]
7. Parker, D.C.; Manson, S.M.; Janssen, M.A.; Hoffmann, M.J.; Deadman, P. Multi-agent systems for the simulation of land-use and land-cover change: A review. *Ann. Assoc. Am. Geogr.* **2003**, *93*, 314–337. [[CrossRef](#)]
8. Ulicny, B.; Thalmann, D. Towards interactive real-time crowd behavior simulation. In *Computer Graphics Forum*; Blackwell Publishing, Inc.: Saarbrücken, Germany, 2002; pp. 767–775.
9. Hamidi, H.; Kamankesh, A. An approach to intelligent traffic management system using a multi-agent system. *Int. J. Intell. Transp. Syst. Res.* **2018**, *16*, 112–124. [[CrossRef](#)]
10. Roche, B.; Guégan, J.-F.; Bousquet, F. Multi-agent systems in epidemiology: A first step for computational biology in the study of vector-borne disease transmission. *BMC Bioinform.* **2008**, *9*, 1–9. [[CrossRef](#)]
11. Janssen, M. *Complexity and Ecosystem Management: The Theory and Practice of Multi-Agent Systems*; Edward Elgar Publishing: Cheltenham, UK, 2002.
12. Oroojlooy, A.; Hajinezhad, D. A review of cooperative multi-agent deep reinforcement learning. *Appl. Intell.* **2023**, *53*, 13677–13722. [[CrossRef](#)]
13. Canese, L.; Cardarilli, G.C.; Di Nunzio, L.; Fazzolari, R.; Giardino, D.; Re, M.; Spanò, S. Multi-agent reinforcement learning: A review of challenges and applications. *Appl. Sci.* **2021**, *11*, 4948. [[CrossRef](#)]
14. Rosenfeld, A.; Richardson, A. Explainability in human-agent systems. *Auton. Agents Multi-Agent Syst.* **2019**, *33*, 673–705. [[CrossRef](#)]
15. Dignum, F.; Mathieu, P.; Corchado, J.M.; De la Prieta, F. Advances in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation. In *The PAAMS Collection: 20th International Conference, PAAMS 2022, L'Aquila, Italy, 13–15 July 2022, Proceedings*; Springer Nature: Berlin/Heidelberg, Germany, 2022; Volume 13616.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Disinformation in Social Networks and Bots: Simulated Scenarios of Its Spread from System Dynamics

Alfredo Guzmán Rincón ^{1,*}, Ruby Lorena Carrillo Barbosa ², Nuria Segovia-García ¹
and David Ricardo Africano Franco ²

¹ Center of Research of Asturias (Centro de Investigación de Asturias), Corporación Universitaria de Asturias, Bogotá 110221, Colombia; tecnologia.ns@iep.edu.es

² Commercial Engineer, Universidad de Ciencias Aplicadas y Ambientales U.D.C.A, Bogotá 110221, Colombia; rcarrillo@udca.edu.co (R.L.C.B.); dafricano@udca.edu.co (D.R.A.F.)

* Correspondence: alfredo.guzman@asturias.edu.co; Tel.: +57-321-254-0363

Abstract: Social networks have become the scenario with the greatest potential for the circulation of disinformation, hence there is a growing interest in understanding how this type of information is spread, especially in relation to the mechanisms used by disinformation agents such as bots and trolls, among others. In this scenario, the potential of bots to facilitate the spread of disinformation is recognised, however, the analysis of how they do this is still in its initial stages. Taking into consideration what was previously stated, this paper aimed to model and simulate scenarios of disinformation propagation in social networks caused by bots based on the dynamics of this mechanism documented in the literature. For achieving the purpose, System dynamics was used as the main modelling technique. The results present a mathematical model, as far as disinformation by this mechanism is concerned, and the simulations carried out against the increase in the rate of activation and deactivation of bots. Thus, the preponderant role of social networks in controlling disinformation through this mechanism, and the potential of bots to affect citizens, is recognised.

Keywords: disinformation; social networks; bots; model

Citation: Guzmán Rincón, A.; Carrillo Barbosa, R.L.; Segovia-García, N.; Africano Franco, D.R. Disinformation in Social Networks and Bots: Simulated Scenarios of Its Spread from System Dynamics. *Systems* **2022**, *10*, 34. <https://doi.org/10.3390/systems10020034>

Academic Editors: Philippe Mathieu, Fernando De la Prieta, Alfonso González-Briones and Juan M. Corchado

Received: 16 February 2022

Accepted: 4 March 2022

Published: 10 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The academic community has shown widespread interest in understanding how disinformation spreads in virtual media, including social networks, e.g., [1–7], due to the potential of disinformation to trigger various problems for governments, citizens, and other social actors [2]. Thus, the state approach has attributed multiple consequences to disinformation on social networks, such as: the polarisation of citizens' opinions [4], the destruction of the credibility of traditional media [8], the mobility of citizens to prevent the development of public policies [9], among others.

Recently, the spread of misinformation has been growing exponentially [1] as a result of the massive use of social networks. An example of this was the case of COVID-19, when the Russian media RT and Sputnik accused NATO and the United States of America of creating the virus in order to destabilise the Chinese economy, and this information was widely spread on social networks such as Facebook, Twitter and Tik Tok [3,10], or, in the case of the vaccines developed for COVID-19, where the anti-vaccine movement sought to attribute effects such as autism and possible genetic malformations to their use, triggering mistrust on the part of the population and preventing the control of the virus and the mitigation of its transmission [11]. In view of these examples, one of the main problems for social actors, in particular states, as well as the academic community, is the lack of awareness of the existence of this type of information and the lack of understanding of the strategies used by the disinformation agent to ensure the propagation of misinformation on social networks [12–14].

This article focuses on the second problem area. Thus, it is evident from the literature that disinformation can be propagated by the direct intervention of individuals either consciously or unconsciously, as well as by automated accounts known as bots [15]. These bots are present in all social networks, however, in some of them their presence is more noticeable. An example of this is Twitter, where, given the importance of this network in issues related to politics, it has been estimated that between 9% and 15% of active accounts are bots [15–18].

Due to the proliferation of bots used in social networks to misinform and the interest in this mechanism, the literature has focused on establishing the role of bots in the spread of misinformation, with a focus on describing case studies, e.g., [15,18–24], however, the development of models to understand the patterns of propagation of disinformation caused by this mechanism is still in its beginnings, with most of the advances being generalist and not specialised in bots such as the works of Lazer et al. [24], Shao et al. [25]; Vosoughi et al. [26] and Shao et al. [27]. The limited amount of work is due to the difficulty of finding the initial sources of disinformation [4], as well as the absence of robust and easily accessible tools for identifying bots, and thus correctly identifying their activities [15].

In this context, the aim of this article was to model and simulate scenarios of disinformation propagation in social networks caused by bots based on the dynamics of this mechanism documented in the literature. This will strengthen the understanding of the phenomenon of disinformation through bots by making it possible to establish patterns of behaviour in the system and to evaluate the effects of the various decisions made by the actors involved in disinformation, especially in relation to social network policies.

It is important to highlight that, although there is no unified meaning of disinformation due to the large number of definitions and intermediate terms found in the literature, particularly in the literature from Anglo-Saxon countries, (e.g., fake news, misinformation, etc.), this work assumes disinformation to be any deviant information that is intended to distort and mislead a target audience in a predetermined way [3,28]. In this way, disinformation refers to a wide range of content, including fake news, misinformation, misleading content, hate speech and deliberate misreporting, among others [29–31]. Additionally, disinformation is not only about the message itself, but as a practice it has the potential to discredit the messenger and true information due to its close relationship with multiple social sectors, especially politics.

This article is structured in five main sections. The first one broadens the conceptualisation of bots and their features; the second one establishes the behaviour of the disinformation caused by this mechanism in social networks and introduces the diagram of causal loops with their respective dynamic hypotheses; the third section shows the methodology used while the fourth section describes the results with emphasis on the mathematical model developed and the simulations defined in the methodology; finally, the fifth section presents the conclusions of the study.

2. Theoretical Framework and Dynamic Model

2.1. What Are the Bots?

The use of the word bots corresponds to an abbreviation of the Anglicism “software robot” [15], which has permeated the disinformation literature, including that of Latin American origin. Thus, the bots as machines are usually automated to some degree, either independently or with human intervention. The bots as machines are put at the service of external agents, in this case diplomacy. In this sense, this type of software can be categorised as either beneficial or malicious [32]. In the case of those that are beneficial, they are programmed by companies to improve the attention and service provided to their customers [33], or as in the case of the social network Twitter, some are used to inform the time, as documented by Yang et al. [32], or to support the raising of charitable resources [34].

For those called malicious, in some cases they oversee malware distribution or spam generation, but in social networks they have another purpose, which is to automatically produce content and interact with humans on social networks, trying to emulate them and

possibly alter their behaviour [35–37]. According to the Academic Society for Management and Communication [38] these bots have three strategies for influencing the behaviour of online users on social networks:

- Smoke screening: bots use context-related hashtags on social networks to distract online users from the main point of debate in a discussion.
- Misdirecting: the use of hashtags related to the topic of discussion to talk about other non-corresponding plots to divert the attention of the target audience from the disinformation.
- Astroturfing: the bot tries to influence public opinion, e.g., in a political debate, by creating the impression that a large majority is in favour of a certain position.

Although the strategies described above are the basis of the strategies used by the malicious bots, it is necessary to recognise that, along with the emergence of new technologies, these tend to evolve and new strategies emerge, which is why a general view of the behaviour of the system generated by this type of software is required, rather than the study of each of the strategies themselves.

2.2. Disinformation, Bots and Causal Loop Diagrams

Disinformation as a social phenomenon that occurs in social networks is not the result of chance, but of a strategic analysis developed by the disinforming agent, as related by Guzmán and Rodríguez-Canovas [2]. Therefore, the disinforming agent seeks to attract the largest possible target population in order to convert it into a population susceptible to being disinformed [39], which is where the various mechanisms for disinformation, such as bots, emerge [5]. With the linking of the population susceptible to disinformation, we proceed to the propagation of the message [40] to consolidate the disinformed population, which for the purposes of this study will be understood as the individual or subject who was exposed to the message of the disinforming agent.

Considering the above as a basis, the role of bots as a mechanism for spreading disinformation begins with their activation, where the disinformation agent establishes how many he or she wishes to have, however, maintaining a fixed number over time is difficult, given the detection mechanisms that social networks have to eliminate or block this type of account [16,41]. However, the interaction of these fake accounts is usually done with publications based on powerful hashtags, comments and by sharing content, being standardised in terms of the number of characters when providing a machine, as well as in the frequency and time of publications [42]. In view of the number of the target population, which will become a susceptible and post-informed population, bots are characterised by making exclusive use of the organic reach of the account, so that as the number of followers grows, this reach tends to decrease, requiring a greater number of bots to impact a greater number of the population [2]. Figure 1 shows the causal loop diagram that represents the behaviour of the studied phenomenon.

Finally, it should be noted that the spread of disinformation is like the way a disease is spread, where there is a population of infected people (bots), who seek to spread the disinformation message in a susceptible population, hence the previous models developed are based on the SIR model (Susceptible—Infected—Recovered), e.g., [43,44]. Thus, previous studies have sought to complement the SIR model by analysing specific disinformation mechanisms, as well as models that integrate several of these mechanisms, (e.g., bots, trolls, paid outreach, etc.). These models include the SIRaRu model, which allowed us to understand the behaviour of disinformation in homogeneous and heterogeneous communities [45], the SEIR model (Susceptible—Exposed—Infectious—Recovered), the SIR model for dynamic and complex social networks [46], among others. Hence, the model presented here both in Figure 1 and in the subsequent sections is based on the SIR model.

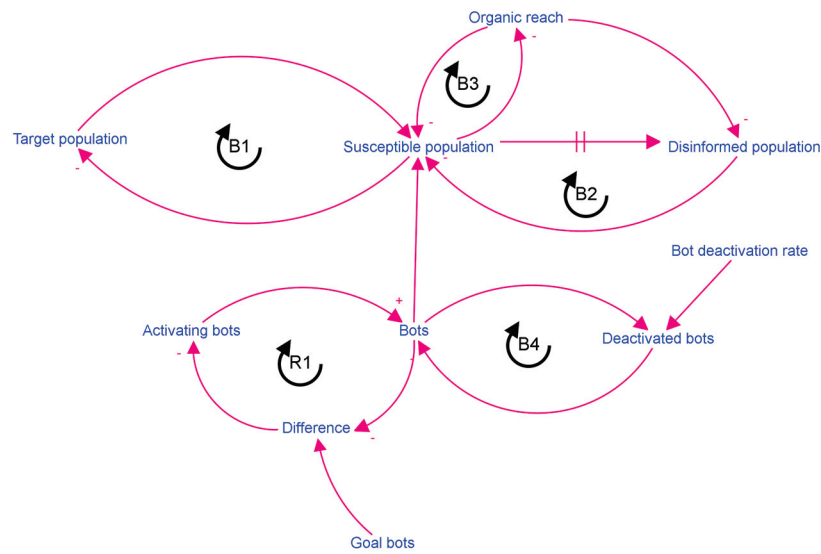


Figure 1. Diagram of causal loops. Note: B represents the balance loops of the system and R represents the reinforcement loops.

3. Methodology

The aim of this article was to model and simulate scenarios of disinformation propagation in social networks caused by bots based on the dynamics of this mechanism documented in the literature, so the main technique used for the development of the model was System Dynamics, considering Bala et al. [47] and Bianchi [48] as theoretical references. In this sense, the choice of modelling technique is based on the complexity of the disinformation caused by bots, in which various elements are involved and whose behaviour is characterised by a non-linear, multicausal and time lagged behaviour [47]. The model is based on the existing literature on the problem under study, for which the steps established by Bala et al. [47] were followed and summarised below:

- Construction of the flow diagram and levels of the model: this corresponds to the physical structure of the system in a defined period and allows the generation of the model’s patterns. In this step, the variables that allow the system’s behaviour to be represented are defined.
- The writing of differential equations: they represent the causes and effects of the system and allow its operationalisation.
- Parameter estimation: they assign the values of the variables that make it possible to simulate and obtain the system’s behaviour. For the present work, this stage was based on multiple sources, especially the study developed by Himelein-Wachowiak et al. [15].
- The testing of the internal consistency of the model: this corresponds to the evaluation of the behaviours generated by the model and which, for the present case, is based on the theory exposed in Section 2.2.

With the proposed model, we proceeded to develop the simulations presented in Table 1, for which modifications were made to the parameters established in the initial model (Table 2); it should be noted that in the execution of the simulations, only the parameter indicated in Table 1 was modified, and the other parameters retained their initial values. The description of the modified parameters is presented in Table 2.

Table 1. Computer simulations.

Code	Simulation	Modified Parameters
Sim—1	Reference simulation	NA
Sim—2	Increased activation rate	TAB = 0.8
Sim—3	Increased deactivation rate	TDB = 0.1
Sim—4	Beginning of disinformation	RD = 30

Table 2. Variables required for model development and initial parameters.

Variables	Conceptualización	Unidades	Parámetro Inicial
$PobO_t$	Target population: a population with specific characteristics that was defined by the agent to be susceptible to disinformation	People	1,000,000
$PobS_t$	Susceptible population: the population that is subscribed to one of the disinformation agent's accounts.	People	0
$PobD_t$	Disinformed population: the population that visualised or had some contact with the disinforming message.	People	0
$Bots_t$	Bots: number of bots in the social network used by the disinformation agent at a given time.	Bots	5
$BotsD_t$	Deactivated bots: corresponds to the number of bots that were deactivated by the algorithm or social network staff.	Bots	0
$TCpobC$	Target population growth rate: is the ratio of growth of the population $PobO$ for a specified t .	%	0.1%
AO	Organic reach: percentage of publications viewed by the Bots through the distribution of the algorithm. This rate is defined according to the number of $PobS_t$ contacted.	%	GRAPH (TIME) (0, 0.0017), (10001, 0.0004), (100001, 0.00015), (300000, 0.00015), (400000, 0.00015), (500000, 0.00015), (600000, 0.00015), (700000, 0.00015), (800000, 0.00015), (1000000, 0.00015), (10000001, 0.00008)
TEC	Effective contact rate: defined as the percentage of people who subscribe to the disinformation agent's account.	%	0.01%
RD	Disinformation delay: This corresponds to the initial t at which message starts its propagation. This lag was developed by using the delay function.	Days	90
TAB	Bot activation rate: the percentage of bots that are activated in each t .	%	50%
$METB$	Goal bots: number of bots that the uninformed agent wishes to have active.	Bots	300
TDB	Bot deactivation rate: the percentage of bots that are deactivated by the social network at a given t .	%	1.973%
$RTDB$	Delay deactivation of bots: This corresponds to the initial t at which the deactivation of the bots is initiated. This lag was developed by using the delay function	Days	98

The analyses of the simulations were carried out descriptively, and in order to determine the existence of statistically significant differences in the behaviour of the system, the medians of the $PobD$ level were compared (see Table 2). The Kolmogorov-Smirnov statistic was applied to check whether the data fit a normal distribution (p -value > 0.05), and it was found that the data did not follow a normal distribution. Thus, to establish the difference in averages between the behaviour of the system with the initial parameters and the modified parameters, the Wilcoxon test was used, considering this difference with a p -value < 0.05.

Finally, the computational work on the model and the simulations were implemented in Stella Architect software version 1.9.5. The following model settings were considered: $t_i = 0$, $t_f = 360$, $\Delta t = 4$, where t represents time in days; additionally, Euler was used as the integration method. SPSS software version 25 was used for the statistical analyses.

4. Results

The results are presented in two sections. The first one describes the model proposed with the capacity to replicate the behaviour of the system based on the evidence from

the literature review; and the second one describes the simulations obtained and the corresponding statistical analyses.

4.1. Model

Figure 2 represents the flow and level diagram [1], which is based on the SIR model. Thus, it consisted of five level variables, five flow variables and 10 auxiliary variables. The green section represents the process of disinformation of the target population, and the blue section how the activation process and deactivation of bots behaves.

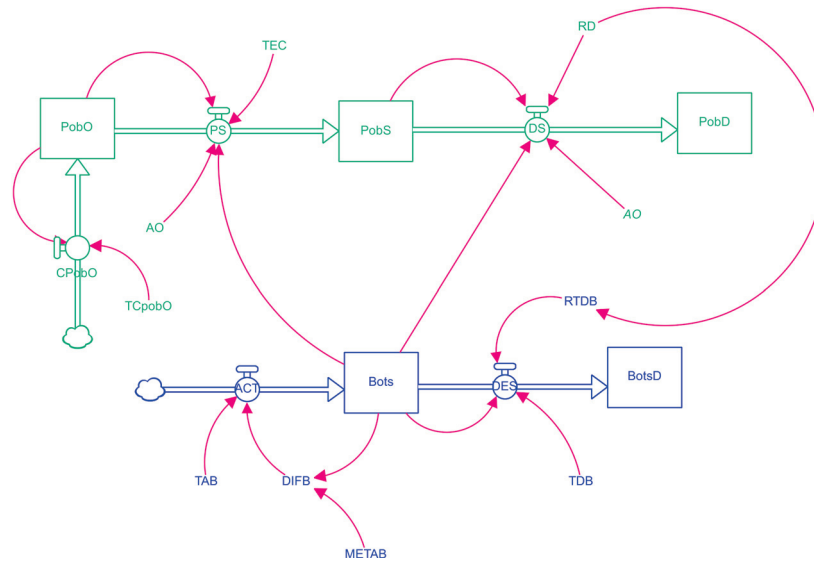


Figure 2. Diagram of flows and levels of disinformation in social networks by means of bots.

The proposed model explains the behaviour of the phenomenon studied here, as long as the following assumptions are met:

- Between the moment of the constitution of the population susceptible to disinformation (PobS) and the beginning of the spread of disinformation there is a delay.
- There is a limited number of bots that the disinformation agent wishes to place on the social network.
- The existence of a delay for the deactivation of bots.

Under the technical conditions of non-negativity of the level variables, (i.e., their domain is restricted to 0 or positive numbers) and that $t = 0, 1, 2, \dots, 360$, the system was represented by the following differential equations.

Target population:

$$PobO_t = [PobO_{t-1} + (PobO_{t-1} \times TCpobC) - (PobO_{t-1} \times AO \times TEC \times Bots)] dt \quad (1)$$

Susceptible population:

$$PobS_t = [PobS_{t-1} + (PobO_{t-1} \times AO \times TEC) - (PobS_{t-1} \times f(X_t, X_{t-\tau}, dt; t \geq t_0))] dt \quad (2)$$

where:

$$X_t = [PobS_{t-1} \times Bots \times AO] dt \quad (3)$$

Misinformed population:

$$PobD_t = [PobD_{t-1} + (PobS_{t-1} \times f(X_t, X_{t-\tau}, dt; t \geq t_0))] dt \quad (4)$$

Bots:

$$Bots_t = \left[Bots_{t-1} + \left(Bots_{t-1} \times TAB \times \frac{METB - Bots}{METB} \right) - (Bots_{t-1} \times f(Y, Y_{t-\tau}, dt; t \geq t_0)) \right] dt \quad (5)$$

where:

$$Y_t = [Bots_{t-1} \times TDB \times RTDB] dt \quad (6)$$

Deactivated bots:

$$Bots_t = [Bots_{t-1} + (Bots_{t-1} \times f(Y, Y_{t-\tau}, dt; t \geq t_0))] dt \quad (7)$$

Table 2 presents the description of the variables and the initial parameters for the operationalisation of the model. Because of the generalist nature of the proposed model, which is applicable to any social network, the initial parameters are susceptible to modification. In this particular case, those of Twitter were used, so parameters such as organic reach, paid reach, effective contact rate, among others, must be modified for its use in other social networks.

4.2. Simulations

Compared to the results obtained in SIM-1, it was found that under the initial parameters for $t = 360$ *PobO* increased by 230,000 people, with 159,000 being effectively uninformed, which represented 15.9% of the initial *PobO*. On the other hand, for $t = 90$, period in which the disinformation of *PobS* begins, this was close to 28,200 people, increasing until $t = 111$, after this day the *PobS* begins to decrease until it reaches the value of 0. In the case of the bots for $t = 133$ the highest number of activated automata was $54.8 \approx 55$ and for $t = 360$ the number of deactivated bots was 158. Figure 3 shows the behaviour of the system for SIM-1.

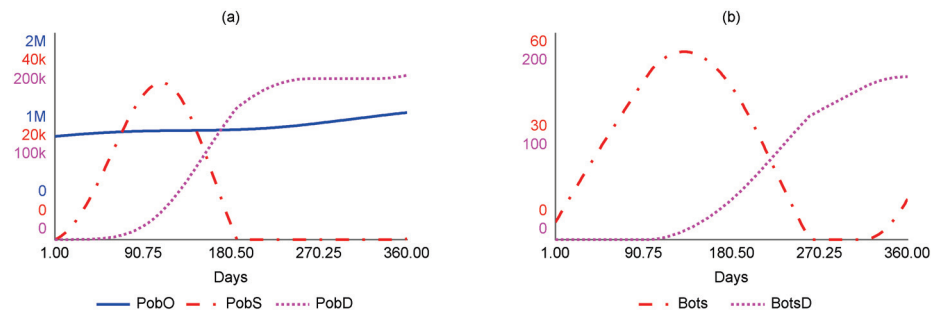


Figure 3. Results SIM-1. Note: (a) behaviours for the levels of the disinformation process and (b) behaviour of the activation and deactivation of bots.

The results for SIM-2 show that with a higher activation rate *bot*, the *PobO* for $t = 360$ would be equal to 1,150,000 people, with *PobD* being 228,000, which represented an increase of 43.39% compared to SIM-1. For the bots in this simulation for $t = 132$ the highest number of activated automata would be reached with a total of $81.2 \approx 82$. Similarly, due to the increase in the activation rate for $t = 360$, a total of 238 bots would have been deactivated, showing an increase of 50.63% in relation to SIM-1. On the other hand, a comparison between the *PobD* between SIM-1 and SIM-2 showed statistically significant differences with $z = -16.42$, p -value = 0.00. Figure 4 shows the behaviour of the system for SIM-2.

For SIM-3, which sought to simulate a higher rate of bots' deactivation by social networks, a change in system behaviour was evident (see Figure 5). Thus, the disinformation target population *PobO* for $t = 360$ would be equal to 1,260,000 people with a total of 109,000 people being disinformed, decreasing the *PobD* by 31.44% in relation to SIM-1. Similarly, in the case of the susceptible population, there are two peaks at $t = 107$ and $t = 360$ (see Figure 5a), correlated with the number of active bots (see Figure 5b). Regarding

the *PobD* level between SIM-3 and SIM-1, it was determined that there are statistically significant differences with $z = -14.00$ p -value = 0.00 .

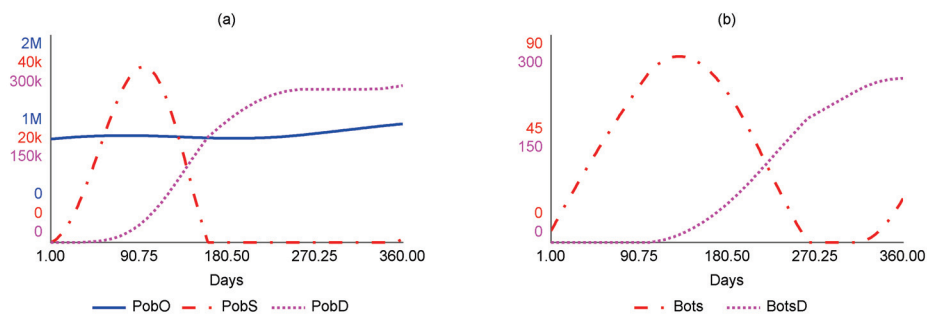


Figure 4. Results SIM-2. Note: (a) behaviours for the levels of the disinformation process and (b) behaviour of the activation and deactivation of bots.

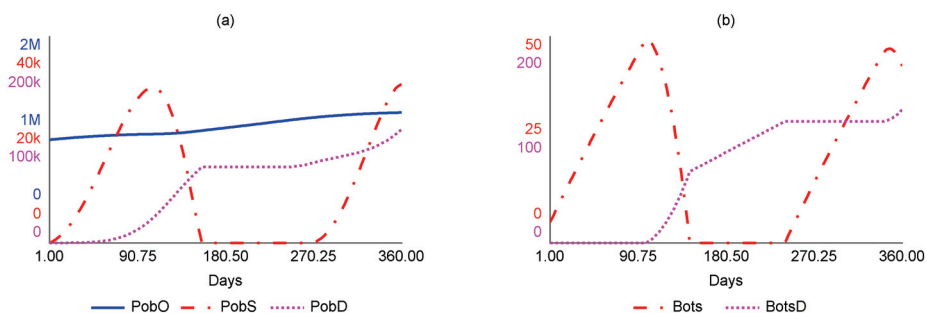


Figure 5. Results SIM-3. Note: (a) behaviours for the levels of the disinformation process and (b) behaviour of the activation and deactivation of bots.

Now, in the case that disinformation starts to circulate at $t = 30$, it was observed that the target population of disinformation *PobO* for $t = 360$ would be equal to 1,240,000, being similar to the behaviour derived from SIM-1. For the population that managed to be uninformed, it was determined that for $t = 360$ the *PobD* was 145,000. Similarly, in the case of bots, and due to the ability of social networks to deactivate them, after $t = 170$ the number of active automatons tends to stabilise. Thus, for $t = 360$ a total of $23.3 \approx 24$ active bots and 147 deactivated bots were evident. On the other hand, the comparison between the *PobD* between SIM-1 and SIM-2 established statistically significant differences with $z = -14.66$, p -value = 0.00 . Figure 6 shows the behaviour of the system for SIM-4.

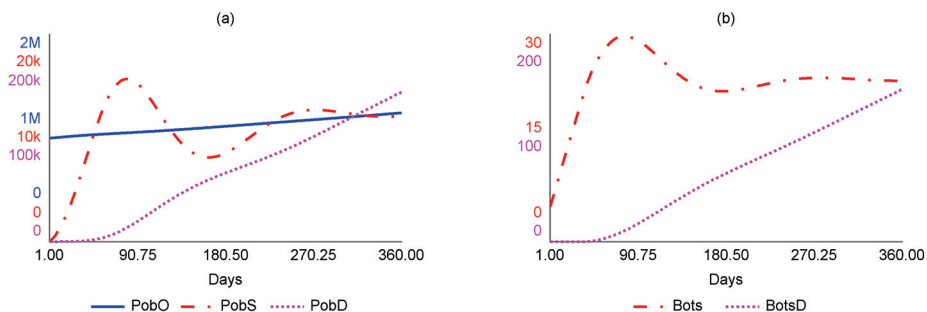


Figure 6. Results SIM-4. Note: (a) behaviours for the levels of the disinformation process and (b) behaviour of the activation and deactivation of bots.

5. Conclusions

The objective of this study, which was to model and simulate scenarios of the propagation of disinformation in social networks caused by bots based on the dynamics of this mechanism documented in the literature, was achieved. The model presented has the capacity to replicate the behaviour of the system, being consistent with the dynamic

hypotheses set out in Figure 1, complementing previous studies such as those developed by de Lazer et al. [24], Shao et al. [25]; Vosoughi et al. [26] and Shao et al. [27], in relation to the use of bots as a mechanism to propagate false information.

This model allows actors involved in disinformation to analyse in a more objective way the behavioural patterns of disinformation caused by bots for decision making, based on three assumptions. The first one relates to the delay in the start of disinformation; the second one to the limited number of bots that the disinformation agent can put in place; and the third one to the limitations of social network systems to detect and deactivate automated accounts.

With that said, the simulations developed clarify that the system of disinformation using bots is susceptible to policies that are conducive to better detection, blocking and elimination of these types of accounts. In this sense, the uninformed population is smaller, hence the responsibility of social networks to design better detection mechanisms and of citizens to report these types of accounts, to increase the effective rate of deactivation of bots. However, if the disinformation agent starts its activity early, the impact over time will not be on the amount of the disinformed population, but on the number of bots required to achieve its purpose, since the number of bots tends to stabilise over time.

From the proposed model and the simulations developed, it is necessary to recognise the role of bots in aggravating existing social problems as a result of the propagation of false information, hence the need to delve deeper into various analyses such as the evolution of this type of mechanism, the new technologies they incorporate to circumvent the security systems of social networks, the use of artificial intelligence in these, among other aspects. On the other hand, it is necessary to urge the academic community to make use of the model, to complement it and, above all, to eliminate the current barriers to the study of disinformation, such as the difficulties of access to declassified information that would allow the model to be operationalised under conditions different from those expressed in this article and which are based on secondary data from other studies.

Author Contributions: Conceptualization, A.G.R., R.L.C.B., D.R.A.F. and N.S.-G.; methodology, A.G.R.; software, A.G.R. and N.S.-G.; validation, R.L.C.B.; formal analysis, A.G.R.; investigation, A.G.R. and D.R.A.F.; resources, A.G.R.; data curation, A.G.R. and R.L.C.B.; writing—original draft preparation, A.G.R.; writing—review and editing, N.S.-G.; supervision, R.L.C.B.; project administration, A.G.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding, and the APC was funded by Corporación Universitaria de Asturias.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data and the model are available at: <https://exchange.iseesystems.com/models/player/alfredoguzmanrincon/disinformation-and-bots> (accessed on 4 March 2022).

Acknowledgments: To Cecilia Carabajal who, with her unconditional support, made the style correction and translation of this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jussila, J.; Suominen, A.H.; Partanen, A.; Honkanen, T. Text Analysis Methods for Misinformation-Related Research on Finnish Language Twitter. *Future Internet* **2021**, *13*, 157. [CrossRef]
2. Guzmán, A.; Rodríguez-Cánovas, B. Disinformation Propagation in Social Networks as a Diplomacy Strategy: Analysis from System Dynamics. *JANUS. NET e-J. Int. Relat.* **2021**, *11*, 32–43. [CrossRef]
3. Agarwal, N.K.; Alsaeedi, F. Understanding and Fighting Disinformation and Fake News: Towards an Information Behavior Framework. *Proc. Assoc. Inf. Sci. Technol.* **2020**, *57*, e327. [CrossRef]
4. La Cour, C. Theorising Digital Disinformation in International Relations. *Int. Polit.* **2020**, *57*, 704–723. [CrossRef]
5. Buchanan, T.; Benson, V. Spreading Disinformation on Facebook: Do Trust in Message Source, Risk Propensity, or Personality Affect the Organic Reach of “Fake News”? *Soc. Media Soc.* **2019**, *5*, 205630511988865. [CrossRef]

6. Bjola, C. The Ethics of Countering Digital Propaganda. *Ethics Int. Aff.* **2018**, *32*, 305–315. [[CrossRef](#)]
7. Gerrits, A.W.M. Disinformation in International Relations: How Important Is It? *Secur. Hum. Rights* **2018**, *29*, 3–23. [[CrossRef](#)]
8. Bennett, W.L.; Livingston, S. The Disinformation Order: Disruptive Communication and the Decline of Democratic Institutions. *Eur. J. Commun.* **2018**, *33*, 122–139. [[CrossRef](#)]
9. Acosta-Quiroz, J.; Iglesias-Osores, S. COVID-19: Desinformación en Redes Sociales. *Rev. Cuerpo Med. HNAAA* **2020**, *13*, 218–219. [[CrossRef](#)]
10. Gottlieb, M.; Dyer, S. Information and Disinformation: Social Media in the COVID-19 Crisis. *Acad. Emerg. Med.* **2020**, *27*, 640–641. [[CrossRef](#)]
11. Grimes, D.R. Medical Disinformation and the Unviable Nature of COVID-19 Conspiracy Theories. *PLoS ONE* **2021**, *16*, e0245900. [[CrossRef](#)] [[PubMed](#)]
12. Spradling, M.; Straub, J.; Strong, J. Protection from ‘Fake News’: The Need for Descriptive Factual Labeling for Online Content. *Future Internet* **2021**, *13*, 142. [[CrossRef](#)]
13. Helmstetter, S.; Paulheim, H. Collecting a Large Scale Dataset for Classifying Fake News Tweets Using Weak Supervision. *Future Internet* **2021**, *13*, 114. [[CrossRef](#)]
14. Shu, K.; Bhattacharjee, A.; Alatawi, F.; Nazer, T.H.; Ding, K.; Karami, M.; Liu, H. Combating Disinformation in a Social Media Age. *WIREs Data Min. Knowl. Discov.* **2020**, *10*, e1385. [[CrossRef](#)]
15. Himelein-Wachowiak, M.; Giorgi, S.; Devoto, A.; Rahman, M.; Ungar, L.; Schwartz, H.A.; Epstein, D.H.; Leggio, L.; Curtis, B. Bots and Misinformation Spread on Social Media: Implications for COVID-19. *J. Med. Internet Res.* **2021**, *23*, e26933. [[CrossRef](#)] [[PubMed](#)]
16. Samper-Escalante, L.D.; Loyola-González, O.; Monroy, R.; Medina-Pérez, M.A. Bot Datasets on Twitter: Analysis and Challenges. *Appl. Sci.* **2021**, *11*, 4105. [[CrossRef](#)]
17. Badawy, A.; Ferrara, E.; Lerman, K. Analyzing the Digital Traces of Political Manipulation: The 2016 Russian Interference Twitter Campaign. In Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Barcelona, Spain, 28–31 August 2018; pp. 258–265.
18. Jin, F.; Wang, W.; Zhao, L.; Dougherty, E.; Cao, Y.; Lu, C.-T.; Ramakrishnan, N. Misinformation Propagation in the Age of Twitter. *Computer* **2014**, *47*, 90–94. [[CrossRef](#)]
19. Xu, W.; Sasahara, K. Characterizing the Roles of Bots on Twitter during the COVID-19 Infodemic. *J. Comput. Soc. Sci.* **2021**, 1–19. [[CrossRef](#)]
20. Lanius, C.; Weber, R.; MacKenzie, W.I. Use of Bot and Content Flags to Limit the Spread of Misinformation among Social Networks: A Behavior and Attitude Survey. *Soc. Netw. Anal. Min.* **2021**, *11*, 32. [[CrossRef](#)]
21. Wang, P.; Angarita, R.; Renna, I. Is This the Era of Misinformation yet: Combining Social Bots and Fake News to Deceive the Masses. In Proceedings of the Web Conference 2018—WWW ’18 Companion, Lyon, France, 23–27 April 2018; ACM Press: Lyon, France, 2018; pp. 1557–1561.
22. Dunn, A.G.; Surian, D.; Dalmazzo, J.; Rezazadegan, D.; Steffens, M.; Dyda, A.; Leask, J.; Coiera, E.; Dey, A.; Mandl, K.D. Limited Role of Bots in Spreading Vaccine-Critical Information Among Active Twitter Users in the United States: 2017–2019. *Am. J. Public Health* **2020**, *110*, S319–S325. [[CrossRef](#)]
23. Tandoc, E.C. The Facts of Fake News: A Research Review. *Sociol. Compass* **2019**, *13*, e12724. [[CrossRef](#)]
24. Lazer, D.M.J.; Baum, M.A.; Benkler, Y.; Berinsky, A.J.; Greenhill, K.M.; Menczer, F.; Metzger, M.J.; Nyhan, B.; Pennycook, G.; Rothschild, D.; et al. The Science of Fake News. *Science* **2018**, *359*, 1094–1096. [[CrossRef](#)] [[PubMed](#)]
25. Shao, C.; Ciampaglia, G.L.; Varol, O.; Yang, K.-C.; Flammini, A.; Menczer, F. The Spread of Low-Credibility Content by Social Bots. *Nat. Commun.* **2018**, *9*, 4787. [[CrossRef](#)] [[PubMed](#)]
26. Vosoughi, S.; Roy, D.; Aral, S. The Spread of True and False News Online. *Science* **2018**, *359*, 1146–1151. [[CrossRef](#)] [[PubMed](#)]
27. Shao, C.; Hui, P.-M.; Wang, L.; Jiang, X.; Flammini, A.; Menczer, F.; Ciampaglia, G.L. Anatomy of an Online Misinformation Network. *PLoS ONE* **2018**, *13*, e0196087. [[CrossRef](#)]
28. Innes, M. Techniques of Disinformation: Constructing and Communicating “Soft Facts” after Terrorism. *Br. J. Sociol.* **2020**, *71*, 284–299. [[CrossRef](#)]
29. Rodríguez, A.R. Fundamentos del Concepto de Desinformación Como Práctica Manipuladora en la Comunicación Política y Las Relaciones Internacionales. *Hist. Comun. Soc.* **2018**, *23*, 231–244. [[CrossRef](#)]
30. McGonagle, T. “Fake News”: False Fears or Real Concerns? *Neth. Q. Hum. Rights* **2017**, *35*, 203–209. [[CrossRef](#)]
31. Fallis, D. What is Disinformation? *Libr. Trends* **2015**, *63*, 401–426. [[CrossRef](#)]
32. Yang, K.; Varol, O.; Davis, C.A.; Ferrara, E.; Flammini, A.; Menczer, F. Arming the Public with Artificial Intelligence to Counter Social Bots. *Hum. Behav. Emerg. Tech.* **2019**, *1*, 48–61. [[CrossRef](#)]
33. Rossmann, A.; Zimmermann, A.; Hertweck, D. The Impact of Chatbots on Customer Service Performance. In *Advances in the Human Side of Service Engineering*; Spohrer, J., Leitner, C., Eds.; Springer International Publishing: Cham, Switzerland, 2020; Volume 1208, pp. 237–243, ISBN 9783030510565.
34. Savage, S.; Monroy-Hernandez, A.; Höllerer, T. Botivist: Calling Volunteers to Action Using Online Bots. In Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing, San Francisco, CA, USA, 27 February–2 March 2016; ACM: San Francisco, CA, USA, 2016; pp. 813–822.

35. Zhang, J.; Zhang, R.; Zhang, Y.; Yan, G. The Rise of Social Botnets: Attacks and Countermeasures. *IEEE Trans. Dependable Secur. Comput.* **2018**, *15*, 1068–1082. [[CrossRef](#)]
36. Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; Tesconi, M. The Paradigm-Shift of Social Spambots: Evidence, Theories, and Tools for the Arms Race. In Proceedings of the 26th International Conference on World Wide Web Companion—WWW '17 Companion, Perth, Australia, 3–7 April 2017; ACM Press: Perth, Australia, 2017; pp. 963–972.
37. Ferrara, E.; Varol, O.; Davis, C.; Menczer, F.; Flammini, A. The Rise of Social Bots. *Commun. ACM* **2016**, *59*, 96–104. [[CrossRef](#)]
38. Academic Society for Management and Communication. How Powerful Are Social Bots? Understanding the Types, Purposes and Impacts of Bots in Social Media. 2018. Available online: https://www.akademische-gesellschaft.com/fileadmin/webcontent/Publikationen/Communication_Snapshots/AGUK_CommunicationSnapshot_SocialBots_June2018.pdf (accessed on 18 December 2021).
39. Hollenbaugh, E.E.; Ferris, A.L. Facebook Self-Disclosure: Examining the Role of Traits, Social Cohesion, and Motives. *Comput. Hum. Behav.* **2014**, *30*, 50–58. [[CrossRef](#)]
40. Jahng, M.R. Is Fake News the New Social Media Crisis? Examining the Public Evaluation of Crisis Management for Corporate Organizations Targeted in Fake News. *Int. J. Strateg. Commun.* **2021**, *15*, 18–36. [[CrossRef](#)]
41. Pozzana, I.; Ferrara, E. Measuring Bot and Human Behavioral Dynamics. *Front. Phys.* **2020**, *8*, 125. [[CrossRef](#)]
42. Kooti, F.; Moro, E.; Lerman, K. Twitter Session Analytics: Profiling Users' Short-Term Behavioral Changes. In *Social Informatics*; Spiro, E., Ahn, Y.-Y., Eds.; Springer International Publishing: Cham, Switzerland, 2016; Volume 10047, pp. 71–86, ISBN 9783319478739.
43. Zhao, X.; Wang, J. Dynamical Model about Rumor Spreading with Medium. *Discret. Dyn. Nat. Soc.* **2013**, *2013*, 586867. [[CrossRef](#)]
44. Rapoport, A.; Rebhun, L.I. On the Mathematical Theory of Rumor Spread. *Bull. Math. Biophys.* **1952**, *14*, 375–383. [[CrossRef](#)]
45. Wang, J.; Zhao, L.; Huang, R. SIRaRu Rumor Spreading Model in Complex Networks. *Phys. A Stat. Mech. Appl.* **2014**, *398*, 43–55. [[CrossRef](#)]
46. Zhu, H.; Wang, Y.; Yan, X.; Jin, Z. Research on Knowledge Dissemination Model in the Multiplex Network with Enterprise Social Media and Offline Transmission Routes. *Phys. A Stat. Mech. Appl.* **2022**, *587*, 126468. [[CrossRef](#)]
47. Bala, B.K.; Arshad, F.M.; Noh, K.M. *System Dynamics*; Springer Texts in Business and Economics; Springer: Singapore, 2017; ISBN 9789811020438.
48. Bianchi, C. Dynamic Performance Management. In *System Dynamics for Performance Management*; Springer International Publishing: Cham, Switzerland, 2016; Volume 1, ISBN 9783319318448.

Article

A Novel Public Opinion Polarization Model Based on BA Network

Yuanjian Ye ¹, Renjie Zhang ², Yiqing Zhao ³, Yuanyuan Yu ⁴, Wenxin Du ³ and Tinggui Chen ^{1,*}

¹ School of Statistics and Mathematics, Zhejiang Gongshang University, Hangzhou 310018, China; 15178645299@163.com

² School of TaiLong Finance, Zhejiang Gongshang University, Hangzhou 310018, China; 15157711825@163.com

³ School of Economics, Zhejiang Gongshang University, Hangzhou 310018, China; zhaoyiqing0228@163.com (Y.Z.); hae2627286095@163.com (W.D.)

⁴ School of Finance, Zhejiang Gongshang University, Hangzhou 310018, China; yuyy1285467604@163.com

* Correspondence: ctgsimon@mail.zjgsu.edu.cn

Abstract: At present, the polarization of online public opinion is becoming more frequent, and individuals actively participate in attitude interactions more and more frequently. Thus, online views have become the dominant force in current public opinion. However, the rapid fermentation of polarized public opinion makes it very easy for actual topic views to go to extremes. Significantly, negative information seriously affects the healthy development of the social opinion ecology. Therefore, it is beneficial to maintain national credibility, social peace, and stability by exploring the communication structure of online public opinions, analyzing the logical model of extreme public attitudes, and guiding the communication of public opinions in a timely and reasonable manner. Starting from the J–A model and BA network, this paper explores the specific attributes of individuals and opinion network nodes. By incorporating parameters such as individual conformity and the strength of individual online relationships, we established a model of online group attitude polarization, then conducted simulation experiments on the phenomenon of online opinion polarization. Through simulations, we found that individual conformity and the difference in environmental attitude greatly influence the direction of opinion polarization events. In addition, crowd mentality makes individuals spontaneously choose the side of a particular, extreme view, which makes it easier for polarization to form and reach its peak.

Keywords: online public opinion; group polarization; influencing factors; power relations

Citation: Ye, Y.; Zhang, R.; Zhao, Y.; Yu, Y.; Du, W.; Chen, T. A Novel Public Opinion Polarization Model Based on BA Network. *Systems* **2022**, *10*, 46. <https://doi.org/10.3390/systems10020046>

Academic Editors: Philippe Mathieu, Fernando De la Prieta, Alfonso González-Briones and Juan M. Corchado

Received: 13 March 2022

Accepted: 7 April 2022

Published: 9 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, with the rapid development of China's self-media platforms, the polarization of online public opinion has become more frequent, for example, the Tesla car brake failure incident, the self-explosion "0 sugar" incident in Genki Forest, and the China Express blind box pet incident, all of which have aroused widespread social concern. It can be noted that in the process of spreading online opinions, due to the reduction of transmission cost, the amount of information received by individuals increases. At the same time, information homogenization and fragmentation are serious, which makes it difficult for individuals to maintain a neutral and objective attitude toward their actual output. As a result, the information views of the surrounding environment tend to be consistent and then become a driver of polarizing events in online public opinion. In fact, in the process of information sharing and decision making, individuals' actual behaviors are generated by their objective cognition together with psychological activities. Consequently, it is very easy to collide with the surrounding environment and group views. Furthermore, the original decision is biased, leading to different degrees of polarization. Moreover, individuals have differences in age, occupation, family, education level and other various aspects. These differences make them have various sensitivities to the polarization of public opinion and

fluctuations of their own opinions and attitudes, resulting in a complex trend of public opinion polarization events.

Existing studies on opinion polarization usually use a relatively small and simple network structure to analyze changes in the attitudinal values of social groups and focus on the influence of individual heterogeneity on the connections between individuals. In fact, the polarization process of individual attitudes relies on a complex network structure in which complex mechanisms of opinion polarization arise.

Distinguishing from existing research tools, this paper considers the heterogeneity properties of nodes and defines them specifically based on the J–A model, as well as the threshold changes between individuals, adding parameters such as individual consistency and strength of network relationships, forming a complex network structure of the group, and using its basic characteristics of growth and preferential attachment, the BA scale-free network is selected for the study, which can be used to investigate the sensitivity problem of the model correction.

We denote the number of edges formed by individuals and their neighbors as ‘degree’ (P) and use the BA scale-free network to present the power law of the distribution of complex networks. After matching the degree, relationship strength, and attitude values with the relevant data, it is confirmed that the group communication behavior between individuals will make the opposing parties continuously reinforce their own views, and the trend of bifurcation of online opinions is obvious.

Further, we try to put the connections between nodes into the set network model, fully discuss the relationship between the polarization process of individual attitudes and the complex network structure, and comprehensively consider the network public opinion propagation mechanism and polarization prediction laws. Based on the opinion polarization model of the BA network, we simulated the collision process of individual opinions on the network and predicted the values of public attitudes after more than 400 collisions. In the evolutionary process, it was found that most individuals would actively choose extreme views to battle in the evolution of time. Moreover, when they are in a network group with the same interests as their own, they will keep looking for similar views to their own in the mutual communication with members to reinforce their original ideas.

In terms of social organization, the negative impact of public opinion polarization reversal will intensify contradictions in a disguised way. This makes the output effect of superimposed views one-sided and extreme, with less output space for the positive and effective viewpoint information. Based on the BA network’s opinion polarization model, this paper predicts the trend of public attitudes toward online events. In general, it is conducive to control and govern online opinion polarization events by clearly understanding the dynamics and process of online opinion polarization, simulating public opinion and attitude, and controlling the heat of online public opinion in a timely fashion.

2. Literature Review

2.1. Literature Review Based on J–A Model

The J–A model refers to a new social attitude judgment model proposed by Jager and Amblard [1], which shows that the subject’s attitude structure determines the occurrence of assimilation and alienation effects, which in turn lead to the phenomenon of consensus and polarization. Since then, there have been many studies related to the J–A model. Barash et al. [2] found that the complex infection model could produce highly nonlinear infection diffusion dynamics, and its critical mass had potential practical significance for the prediction of the early stage of transmission activities. Li and Tang [3] proposed the threshold model of group behavior and considered group spatial factors and the strength of social influence relationships among individuals. Based on the group polarization effect, Gabbay et al. [4] added a new explanation, that was, the interaction between individuals with the same interests will trigger the change of attitude to extremes in disguise. Chen et al. [5] used the J–A model to study the rumor diffusion process with

the consideration of individual heterogeneity. Subsequently, they took the imported food safety issue as an example during the COVID-19 pandemic and testified to the efficiency of the proposed model.

From the above analysis, it can be seen that the existing literature uses a relatively small and simple network structure to analyze the changes in social group attitude value, which is different from the structure in complex social networks. In addition, many articles do not give node heterogeneity attributes and do not consider the change of threshold between individuals, which is also far from reality. Based on this, this paper integrates the parameters such as individual conformity and network individual relationship strength into the classical J–A model, which makes the model well adapted to complex, real-world events.

2.2. Literature Review of BA Models

The BA model, or scale-free model, was proposed by Barabasi and Albert in 1999 [6]. They pointed out that the network produced by the BA model had the characteristic of no scale, and the distribution of its network degree values followed a power-law distribution, which was closer to most actual networks. Liu et al. [7] conducted further research and found that the BA model could only generate a network model in which the distribution of degrees follows a power index of 3, while the value in the actual network was usually between 1 and 3. Chen et al. [8] explored a multi-dimensional public opinion process based on a complex network dynamics model in the context of derived topics, and they found that information intensity was the most important influence factor. Zhou et al. [9] found that the network generated by the BA model did not have obvious small-world characteristics, while the actual network usually had both unscale and small-world characteristics. In addition, a large number of scholars have found that the BA model is prone to isolated nodes in the application process and has the characteristics of only “first rich” and not “later rich”, which are not in line with the evolutionary characteristics of the actual network.

Combined with existing research, we find that most scholars focus on the interaction between nodes, emphasizing that the heterogeneous characteristics of individuals themselves will affect the connections between individuals. However, we notice that the connections between nodes are not only related to the properties of the nodes but also the network structure. Taking the BA model as the background, the model degree distribution is generally similar to the power-law distribution, and the connection between nodes has the characteristics of merit. This study will try to give a specific definition of the node’s own attributes according to the characteristics of the actual network and emphasize the node characteristics in the network background in order to improve the adaptability of the model in the existing research.

2.3. The Prediction Law of Online Public Opinion Dissemination and Polarization

More and more people have been connected to the world through digital technology in recent years. As a result, public opinion can spread quickly. It is difficult for the public to identify and judge what they want from a large amount of data. At present, more scholars have already conducted in-depth studies on social network structures and the phenomenon of opinion polarization. For example, Wang [10] dissected the dynamic relationship between the factors influencing group attitudes. Chen et al. [11] analyzed the panic emotion propagation process and further identified the emergence process of group panic buying behavior under the COVID-19 pandemic. Wang et al. [12] considered the components of group polarization formation of online public opinion, quantitatively analyzed the mechanism of public opinion polarization dynamics and regulation strategies, and strongly argued the relevance of the main factors of public opinion development through an example simulation. Zhang et al. [13] proposed the intertextual characteristics of the process of generation, diffusion, and polarization in self-media online public opinion. Hatton [14] proposed that preference and significance are related to different individual-level characteristics through the analysis of the European Social Survey and European barometer data. Heizler and Israeli [15] proposed that the tragedy of a specific individual

is more likely to cause the polarization of public opinion than the tragedy of a group. Blake et al. [16] believe that the neutrality and polarization of people's views vary according to sociodemographic characteristics, including age, gender, and education.

The above-mentioned literature has summarized the general law of the polarization phenomenon of online public opinion groups. However, the influencing factors and network structure in the polarization process are seldom analyzed. Based on this, this paper relies on a specific network structure to study the complexity of the polarization mechanism and the process of individual attitude polarization. By fully discussing the relationship between the two, we can understand the communication mechanism of network public opinion and the law of polarization prediction.

3. A Novel Public Opinion Polarization Model Based on BA Network

3.1. Basic J–A Model

Much of the existing research is discussed based on the D–W or J–A models. Both originate from social judgment theory. Social judgment theory analyzes the phenomenon of how an individual's position changes when confronted with different points of view. It is founded on the idea that a person's attitude changes depending on the information that causes the change. If the positive information is close to the individual's initial position, then the information is within the individual's range of acceptance. The view is that the individual is likely to move to the advocated position. That is, individuals are more likely to assimilate similar information. We brought this perspective to the J–A model as an example and obtained the following conclusions.

Individual i and individual j interact with information. The attitude values are based on the distance between them. The rule of attitude value change is related to the difference between the two attitude values. Individuals tend to prefer information close to themselves and reject information farther away, although the quality of attitudes affects the degree of individual interaction. The specific rules are as follows [1].

$$\text{If } |x_i - x_j| < u_i dx_i = \mu \cdot (x_j - x_i) \quad (1)$$

$$\text{If } |x_i - x_j| > t_i dx_i = \mu \cdot (x_i - x_j) \quad (2)$$

where u_i is the threshold when individual i decides to accept the message, t_i is the threshold when individual i rejects the message, and μ is the intensity of the control influence.

3.2. Improved Ideas

The J–A model provides a theoretical basis for information exchange simulation. However, the model does not consider factors such as environmental climate, individual affinity, and individual subordination. This deviates from the actual situation. For example, when the individual's herding is strong, the individual will move towards the stronger party. If the individual's herding is weak, they will adjust and move in a specific direction according to their own and the environmental attitude value. Obviously, the J–A model does not consider the population characteristics and individual attributes, and it does not have practical application value.

At the same time, we assigned the corresponding initial network structure, which aims to meet the environmental conditions in the process of individual interaction. Society is intricate and complex, with varying views on opinion events. In existing studies, small-world networks and BA scale-free networks (from now on referred to as BA networks) are often invoked to simulate realistic social networks to restore real individual attitudinal interaction processes. Small-world networks are derived from the regular network model, in which N nodes relate to probability p on broken edges. Its "degree" distribution is in line with normal distribution. The BA network has a power-law distribution of degrees characterized by a growth mechanism and meritocratic connectivity. The BA network grows while the nodes move to the nodes with a higher degree. In general, both network

structures are closer to reality, and both preserve the diversity in real networks. They both guarantee faster convergence of the algorithm and meet the requirements of the model.

Based on the above considerations, the network group attitude polarization model is improved based on the J–A model. For the attributes of individuals and networks, parameters such as individual followership and strength of personal network relationships are added to the J–A model. The model can be adapted for actual complex events. Moreover, in real society, the network distribution law is mostly reflected in the power-law distribution, and the BA network is used as the agent adjacency model. In addition, we set the effect interval parameters d_1 and d_2 to illustrate the positive or negative effects of relationship strength distribution and followership parameters on group attitude polarization.

3.3. Methodology

3.3.1. J–A Model

The J–A model refers to the new model of social attitude judgment proposed by Jager and Amblard. The main conclusions of the J–A model are as follows: first, the attitude structure of the subject determines the inevitability of its assimilation effect and alienation effect; second, the assimilation effect and the alienation effect have a counter-effect, which will lead to the subject reaching consensus, polarization, and other phenomena. The core idea of the J–A model is based on the theory of social judgment, whereby a person's attitude changes depending on the location of the persuasive information he receives. For example, commentators will be more inclined to make statements with similar views. The idea of this study is to explore the polarization of network public opinion, and the idea is to create a model adapted to different group characteristics and individual attributes, specifically by integrating parameters such as individual conformity and network individual relationship strength into the classical J–A model, so that the model is more suitable for complex, real-world events. The method of model simulation can more intuitively see the assimilation and alienation effects that occur in individual attitudes and the final polarization results.

3.3.2. BA Network

The BA network refers to the scaleless network proposed by Barabasi and Albert that follows power-law distribution. The BA network is based on the growth mechanism and the preferential connection; that is, the size of the BA network shows an increasing trend, and the network nodes will be connected to the nodes with higher proximity. In this study, under the rules of individual attitude interaction, the corresponding initial network structure is assigned to meet the simulation environment. Compared with the intricate interactive networks in reality, the BA network not only retains the diversity of the actual network but also standardizes and simplifies the individual interaction process.

3.3.3. Multi-Agent System

A multi-agent system is a collection of multiple agents that coordinate and serve each other to complete a task together. Its goal is to build large, complex systems into small, easily managed systems that communicate and coordinate with each other and has wide uses in many fields such as platform management [17], the effect of policy implementation [18,19], and so on. A multi-agent system has the following characteristics: first, each agent is independent, autonomous, and can solve a given sub-problem and affect the environment in a specific way; second, agents communicate and coordinate with each other.

The reason why a multi-agent system is selected for this study is precisely because it is suitable for complex and open distributed systems and meets the setting conditions of this paper.

3.4. The Novel Public Opinion Polarization Model

3.4.1. Model Construction

The individuals and connections between the individuals form a population-complex network structure. We define the parameters and features in the network. The model parameters are shown in Table 1 as follows.

Table 1. Model parameters.

Parameters	Definition
P_i	Degree
k_{ij}	Strength of relationship between individuals
$X_i(t)$	Individual attitude value
$S_i(t)$	Environmental attitude value
C_i	The clustering coefficient of individuals
C	The clustering coefficient of the network
M_i	Impact threshold
d_1	Assimilation effect interval
d_2	Exclusion effect interval
β	Assimilation degree coefficient
γ	Exclusion degree coefficient
L	Average distance length

(1) Degree (P_i)

The number of edges formed by individuals and their neighbors is called the degree. The size of the degree reflects the number of individuals in the nearby area. The higher the number of nearby individuals, the higher the importance of individuals. In social relationships, the higher the importance of the individual, the higher the level of information, with considerable power of speech and definition.

(2) Strength of relationship (k_{ij})

The strength of the relationship describes the closeness of the relationship between individual i and individual j . The model assigns a value to k by a random function. The k -value reflects the extent to which individuals influence each other. The range of the k -value is between integers 1 and 4. The strength of the relationship increases sequentially as the value increases.

(3) Individual attitude value ($X_i(t)$)

The individual attitude value is a quantitative indicator of the individual's attitude at the moment t . $S_i(t)$ is the average of all individual attitude values near individual i at the moment, also known as the integrated environmental attitude value. The expression for $S_i(t)$ is as follows.

$$S_i(t) = \sum_{j=1}^n \frac{2k_{ij} - 1}{4(n-1)} X_j(t) \quad (3)$$

where $S_i^+(t)$ is the summation of positive attitude values, and $S_i(t)$ is the summation of negative attitude values. The distribution of $X_i(t)$ conforms to the Gaussian distribution.

(4) The clustering coefficient of individuals (C_i)

The clustering coefficient of individuals is the ratio of the actual number of edges formed by individual i and neighboring individuals to the maximum number of possible edges. The maximum number of possible sides is $(n^2 - n)/2$. C_i reflects the aggregation of individuals. In general, individuals tend to build groups with a high degree of collection. The expression is as follows.

$$C_i = \frac{2n}{n(n-1)} \quad (4)$$

(5) The clustering coefficient of the network (C)

The clustering coefficient of the network C is the average of the clustering coefficients of all individuals in the network, which quantifies the degree of individual aggregation. The expression is as follows.

$$C = \frac{1}{n-2} \sum_{i=1}^n C_i \quad (5)$$

(6) Impact threshold (M_i)

The impact threshold M_i determines whether an individual's attitude has changed, directly responding to the level of information in the neighborhood. If $M_i \geq 1$, the individual attitude value has changed. Otherwise, the individual does not change. α is the adjustment parameter. The expressions are as follows.

$$\text{If } S_i(t) \geq 0 \quad M_i = \alpha S_i^+(t) + C_i \tag{6}$$

$$\text{If } S_i(t) < 0 \quad M_i = \alpha S_i^-(t) + C_i \tag{7}$$

The interpretation of M_i is as follows. According to the rule, whether an individual's attitude value changes depends on its subordination and the degree of environmental influence. There are three main scenarios. In the first case, the individual is highly submissive, entirely influenced by the environment. The individual will always follow the environment and adjust their attitude. In the second case, the environment around the individual is unbalanced, and there will be a view recognized and dominated by more individuals. In this case, the individual will also favor the strong side. In the third case, the individual's subordination combined with the environment drives the individual to move towards a particular side of the camp.

(7) Effect interval parameters (d_1/d_2)

Effect interval parameters specify the range of individual attitude value changes. If the distance between $X_i(t)$ and $S_i(t)$ is less than d_1 , the individual does not follow the rule of exclusion. Otherwise, individuals do not follow the rules of assimilation.

(8) Assimilation/exclusion degree coefficient (β/γ)

The assimilation/exclusion degree coefficient is the degree of control over the value of individual attitude change. β is the degree coefficient of the assimilation rule, and γ is the degree coefficient of the exclusion rule: both range between 0 and 1.

(9) Average distance length (L)

The average distance length is the average number of distances between individuals in the network [20]. The distance between individuals is the sum of the edges connecting both. The maximum distance is the diameter of the network. The L -value reflects the ability and efficiency of information transfer between individuals. Let the path length between individual i and individual j be l_{ij} . The expression of l_{ij} is as follows.

$$L = \frac{2}{n(n-2)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n-1} l_{ij} \tag{8}$$

3.4.2. Simulation Process

To reveal the mechanism of individual attitude polarization, we established a social networking platform. Research has shown that most complex, real-world networks exhibit power-law distribution, which indicates that most individuals have a small degree, and only a few offer a large degree. Barabasi and Albert proposed BA scale-free networks to study this class of networks that exhibit power-law distributions. The basis of the network is Growth and Preferential attachment. Growth means that the complex network structure will continue to expand. Preferential attachment means that the additional individuals are more inclined to connect with individuals of a higher degree. The specific construction method is as follows.

Step1. Growth: We randomly construct the initial network containing m_0 individuals. Next, we constantly increase the number of individuals, and individuals are randomly connected to the original model.

Step2. Preferential attachment: The probability (π_i) that an individual is connected to the network is positively correlated with the degree (p_i) of nearby individuals. The expression is as follows [6].

$$\pi_i = \frac{\sum_{j=1}^n p_i}{\sum_{j=1}^n p_j} \tag{9}$$

BA scale-free networks conform to the characteristics of self-organization, synchronization, and emergence mechanisms in actual society. Therefore, we choose the BA scale-free network to study and make corrections for issues such as model sensitivity.

3.4.3. Interaction Rules

In individual interaction, the impact threshold M_i is calculated by first considering the environmental attitude value, relationship strength, and clustering coefficient. Next, a judgment is made: if $M_i \geq 1$, the interaction takes place; otherwise, the individual attitude value does not change in any way.

We set the effect interval d_1/d_2 as the discriminate condition. A discussion of the interaction process follows.

(1) Assimilation rules

If the distance between $X_i(t)$ and $S_i(t)$ is less than d_1 , it is considered that assimilation of individual and environmental attitudes occurs. The rules of attitude value evolution follow the following rules.

$$X_i(t + 1) = (1 - \beta)X_i(t) + \beta S_i(t) \tag{10}$$

(2) Exclusionary rule

If the distance between $X_i(t)$ and $S_i(t)$ is greater than d_2 , the individual and the environmental attitude values are considered in exclusion. The rules of attitude value evolution follow the following rules.

$$X_i(t + 1) = (1 - \gamma)X_i(t) + \gamma S_i(t) \tag{11}$$

(3) Neutrality rules

If none of the above conditions are met, the individual is considered not to make any changes.

The following flow chart (shown in Figure 1) outlines the discriminatory process of the polarization model.

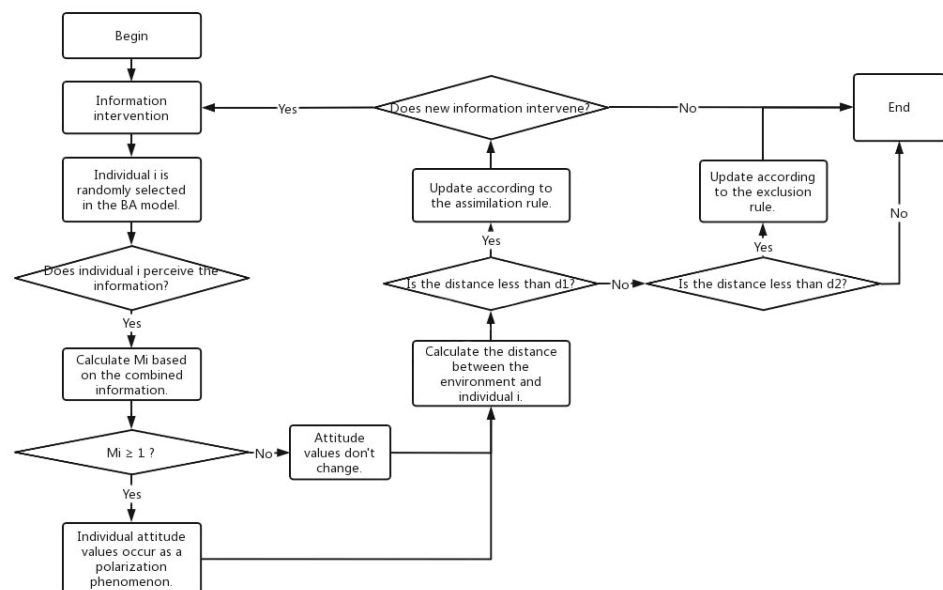


Figure 1. Network population attitude polarization model discrimination process.

4. Experiment Simulation

Because the BA network can present the social network well, this paper defines the BA network as the basis of evolution. By setting different parameter values, this paper makes an intensive study of the evolution process. First, this paper sets the scale of network nodes as 100 and takes $d_1 = 0.3$, $d_2 = 0.7$, $\beta = 0.1$, $\gamma = 0.2$. Through practical operations, this paper finds that after 400 interactions, the individual's attitude will tend to polarize with the surrounding environment, and their attitude value will gradually shift to the two extreme directions of -1 and 1 . However, some individuals will still maintain their original attitude. Furthermore, some individuals will constantly adjust their attitude value in the range of -1 to 1 to achieve a balanced state by adapting to the external environment. Specifically, in the process of attitude evolution, the quantitative distribution of different attitude values under different interaction times is shown in Figure 2 below:

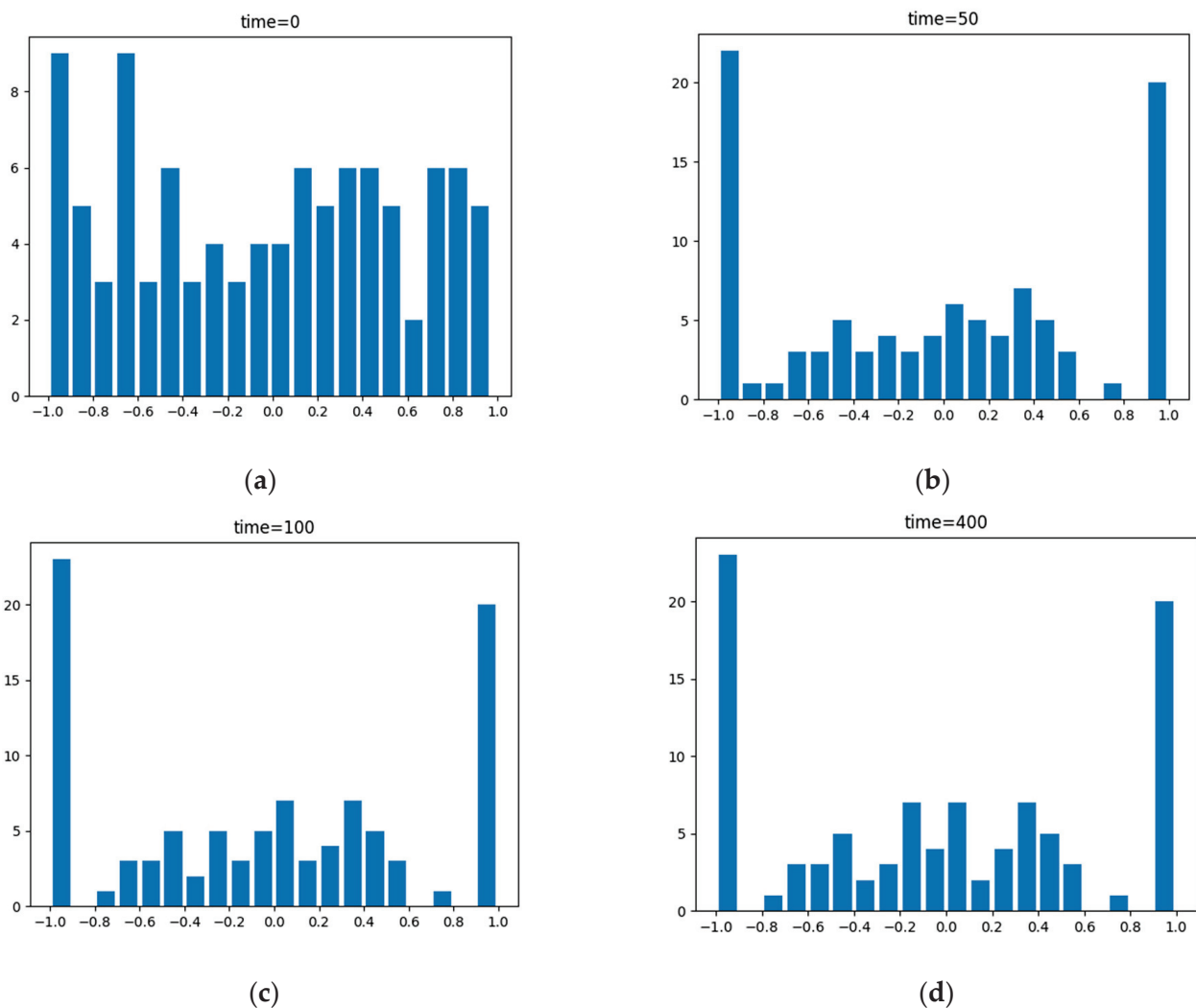


Figure 2. Quantitative distribution of attitude values under different interaction times.

In the initial state, time = 0: the individual attitude value distribution diagram is shown in Figure 2. The abscissa in the diagram represents the individual attitude value, and the ordinate represents the number of individuals corresponding to the attitude value. The simulation results show that in the initial state, the individual attitude value is relatively scattered and evenly distributed. In the initial state, individuals in the group hold their views on events, and there is no clear view of which is right or wrong, or there is a relatively unified opinion. Everyone makes judgments and forms attitude values purely through their views on events. Therefore, in the early stage of event development, there will be no

obvious extreme phenomenon in the attitude value of the group towards an event. With continuous interaction between individuals, when the time is 50, 100, and 400, the attitude value of individuals begins to show a differentiation trend. The specific simulation results are shown in Figure 2.

The number of individuals with a neutral view decreases, while the number of individuals close to 1 and -1 attitude values increases. These changes can obviously show a polarization phenomenon. In the process of increasing the number of interactions, it can be found from the four simulation results that the attitude distribution diagram presented in Figure 2 has been relatively stable. Even a few individuals did not change their attitude values. This paper lists two reasons:

- (1) The low conformity of individuals leads to the failure to reach the threshold of $R > 1$ set by the model. Therefore, the attitude value of other individuals has not influenced them, so their attitude value has been maintained as their initial attitude value.
- (2) Due to the network structure, the gap between the positive and negative sides is very close, making it difficult for the individual to make a choice under the influence of this evenly matched environment. As a result, a few individuals remain neutral from beginning to end, so they never change their attitude value.

In a real event, after each event is polarized, some people will always define the event according to their judgment to maintain their original point of view. Similarly, some individuals will hold a wait-and-see attitude because they cannot understand the truth of the event. However, as the simulation results show, driven by herd mentality, most individuals actively choose an extreme point of view to stand in line, which shows that most individuals show a phenomenon of joining the powerful party to seek security in the face of group events to avoid isolation.

5. An Empirical Case

In this paper, the public opinion polarization model based on the BA network is used to predict the trends in public attitudes towards network events. Based on the 4.1 simulation study, this paper selects the network event of “Hua Chenyu and Zhang Bichen having children unmarried” as a research sample to predict the attitude of online groups. The original data of the case sample is shown in Table 2.

Table 2. Case Sample Public Raw Attitude Values.

Interval	Count	Interval	Count	Interval	Count	Interval	Count
(−1.0,−0.9]	972	(−0.5,−0.4]	488	(0.0,0.1]	551	(0.5,0.6]	549
(−0.9,−0.8]	633	(−0.4,−0.3]	538	(0.1,0.2]	680	(0.6,0.7]	1399
(−0.8,−0.7]	608	(−0.3,−0.2]	521	(0.2,0.3]	541	(0.7,0.8]	768
(−0.7,−0.6]	514	(−0.2,−0.1]	507	(0.3,0.4]	658	(0.8,0.9]	849
(−0.6,−0.5]	519	(−0.1,0.0]	1511	(0.4,0.5]	690	(0.9,1.0]	3083

Data source: Zhang Bichen’s long article posted on Weibo at 17:51 on 22 January 2021.

On 21 January 2021, an unknown netizen broke the news on the Internet: a top male star in the entertainment industry married and had children, the woman was also an insider, and the child was registered when he was one year old. Another netizen revealed that the male star was Hua. On the same day, Hua’s cousin posted a denial. At 17:45 on 22 January 2021, Hua admitted to having a child with Zhang. At 17:51, Zhang also confirmed this by posting a long article on Weibo under his real name.

The incident of “Hua and Zhang having a child out of wedlock” caused an uproar on the Internet. With the continuous revelation of news related to the incident, netizens had a heated discussion, and the public view gradually became distinct and polarized. In this paper, the BA network simulates the state of a real social network, and we use the polarization model of public opinion to simulate and predict the evolution of this event. Through web crawlers, this article obtained the original data set of public attitudes under Zhang’s long post on Weibo at 17:51 on 22 January 2021. In this paper, Python

NLP natural language processing and machine learning are used to obtain 16,579 valid data, thereby determining the size of the instance network nodes. According to the actual situation of the case, this paper determines that the assimilation degree coefficient is 0.005, the repulsion degree coefficient is 0.01, the assimilation effect band distance is 0.3, and the repulsion effect band distance is 0.7. Based on existing stop word rules and machine learning recognition methods, this article assigns a positive or negative attitude value to the initial valid comment. With the soaring heat of the incident, the matter has aroused heated discussion among the public. In the environment of constantly revising the direction of public opinion, the views of network individuals collide, resulting in different degrees of change in their attitudes. This article regards this transformation as a process of individual interaction. Based on the polarization model of public opinion based on the BA network, this paper simulates the process of the collision of individual views on a network and predicts the value of public attitude after the occurrence of 10, 50, 100, and 400 such situations. The forecast statistics are shown in Table 3.

Table 3. Polarization predictions of public attitude values after interaction.

Time	Attitude Value			
	(−1.0,−0.9]	(−0.1,0.0]	(0.0,0.1]	(0.9,1.0]
10	1570	972	1113	3477
50	2926	976	1152	4121
100	3058	965	1154	4230
400	3071	913	954	4269

We captured 16,579 valid comments from Zhang’s statement at 17:51 on 22 January 2021. Figure 3 is valid comments from eight hours after the long article was published. This paper uses certain rules to assign different attitude values to different comments, and the distribution of individual attitude values is shown in Figure 3. The original public attitude value from the case sample was analyzed as follows: 2062 people held a neutral attitude towards the incident, and 4055 people held an absolute positive or negative attitude. At this time, the distribution of public attitude values was relatively even. Network individuals expressed their opinions on the event, and there was no obvious polarization tendency in network public opinion and no clear and unified view. For the incident, many network users still held a wait-and-see attitude and looked forward to the follow-up development of the event; at the same time, there were also a considerable number of netizens who held a “blessing” support attitude or a “not optimistic” opposition attitude.

Based on the model above, the prediction results of the attitude value distribution of the sample dataset after different interactions are shown in Figure 4. After 5 h, 1 day, 2 days, and 8 days of simulated interaction, the polarization trend of public attitudes gradually became obvious, and the number of neutral network individuals began to decrease. From the forecast results, it can be seen that from the beginning of Zhang’s statement at 17:51 on 22 January 2021 to 8 days after the statement was released, the proportion of network users with an absolute positive or absolute negative attitude rose from 24.46% to 44.27%, while the proportion of neutral internet users decreased from only 12.44% to 11.26%. Obviously, the proportion of network individuals who show an absolute attitude has increased significantly, and the polarization trend of network public opinion has become more and more obvious, while the numbers of netizens who indicate a neutral attitude has remained at a low level, and the range of changes is small. With the clarity of the incident, the views of netizens have become more distinct. However, there are still some who hold a neutral attitude, such as “eating melons”, and do not express personal views with a clear attitude.

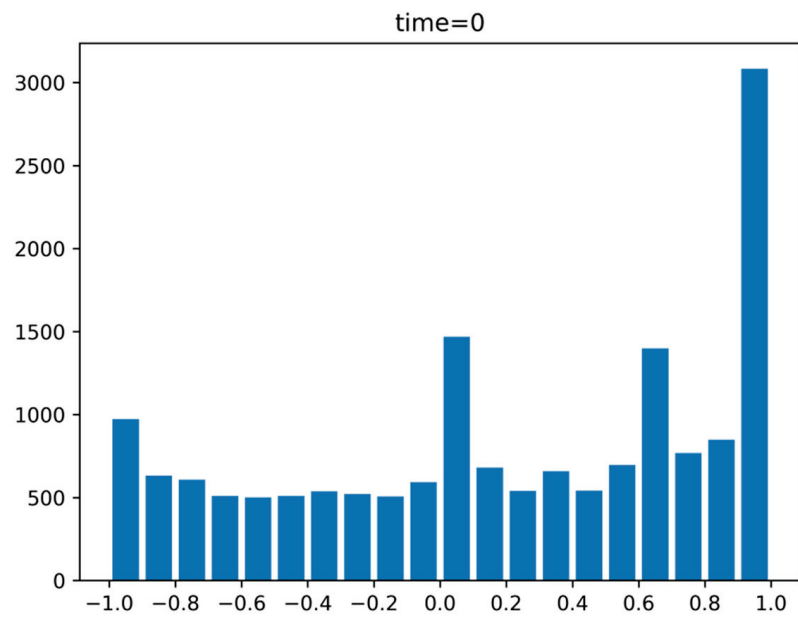


Figure 3. Distribution of public raw attitude values.

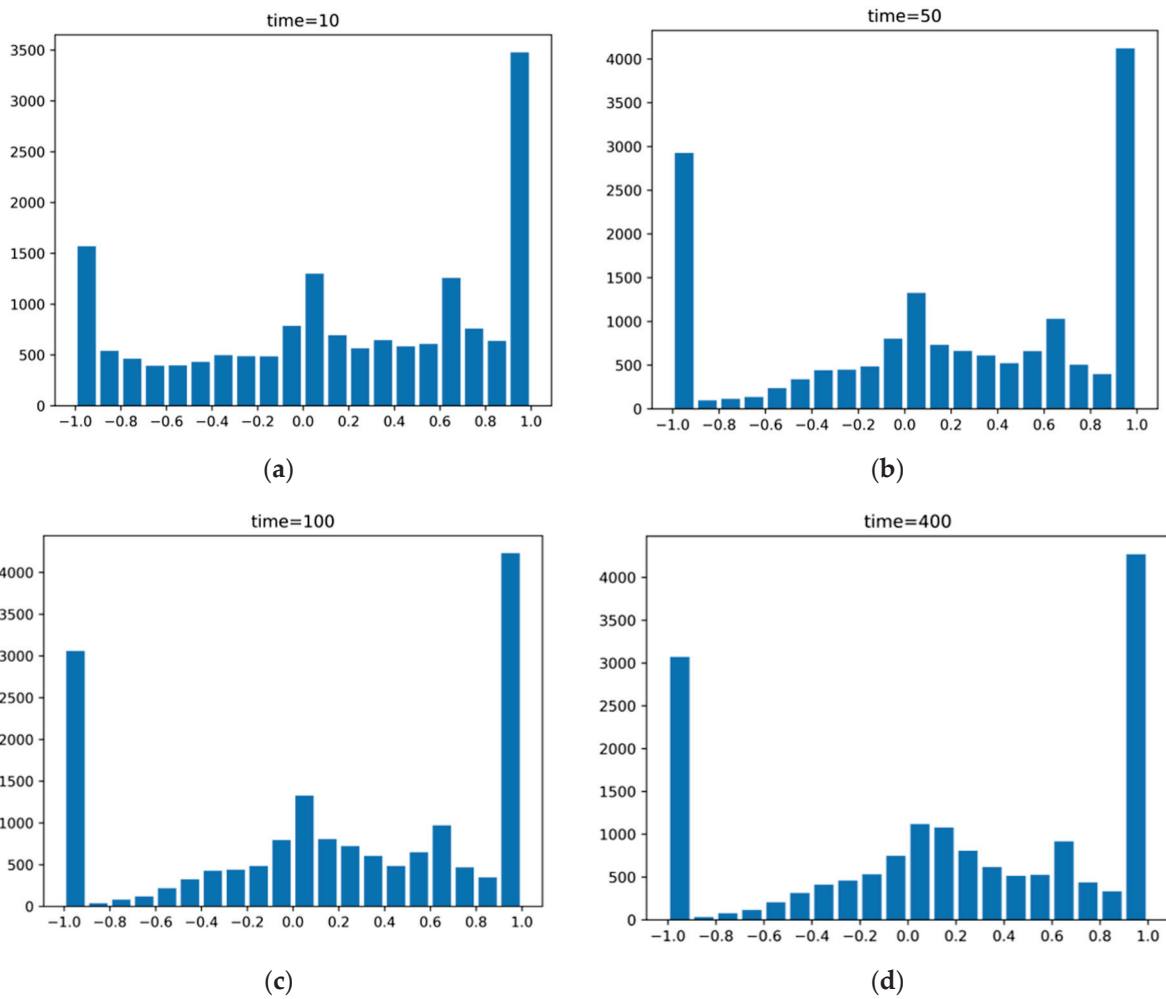


Figure 4. Prediction of attitude value distribution under different interactions.

From the prediction results of attitude value distribution, it was found that with the continuous occurrence of interaction, the growth rate of the proportion of network

individuals with an absolute attitude is from fast to slow, while the proportion of someone with a neutral attitude does not change much. From the simulated interaction results of the four time nodes of 5 h, 1 day, 2 days, and 8 days, it can be seen that from the beginning of Zhang's long article at 17:51 on 22 January 2021 to one day after the incident, the growth rate of the proportion of users who showed an absolute attitude increased from 24.45% to 39.64%, and after the incident, the growth rate slowed down from 39.64% to 0.71%. In summary, after one day of simulated interaction, the trend of the attitude distribution map stabilized.

Individuals participating in the evolution of network public opinion have the characteristics of subjective judgment and labeling processing, as well as passive acceptance and loss of subjectivity. There are countless exchanges of opinions between individuals, which eventually form group behavior, and individual behavior is affected by group behavior. The phenomenon of polarization of online groups mostly occurs in the field of opinions, and the result is mostly that the views are further differentiated and opposed, and the opposing parties continue to strengthen their views in the group discussion, and it is obvious that they cannot merge. At the same time, when a person is in a network group with similar interests or views as a link, he will exchange common ideas and understandings with other members of the group or constantly look for views like his, trying to obtain psychological comfort and strengthen his original concepts.

6. Conclusions

6.1. Summary

In order to explore the propagation structure of online public opinion and analyze the logic of extreme public opinion in the social context of the big data era and the developed self-media network, we constructed a BA network model, simulated and analyzed the trend of public attitudes toward online events and the polarization mechanism of individual attitudes, verified the propagation mechanism and polarization prediction law of online public opinion through experiments, confirmed the validity of the BA model, and obtained the following conclusions through simulation experiments.

- (1) Individuals' attitudes toward public opinion are related to their surroundings. When an individual's attitude changes toward an event, it is often due to the influence of other perspectives in communicating with other individuals. Through the J–A model, we can understand that the value of an individual's attitude at a particular moment depends on their attitude and the surrounding environment at the last moment. Based on this principle, we investigated the specific changes in attitude values.
- (2) The discrimination of attitudinal values depends on distance. Based on the difference in attitude values, the model specifies interaction rules to determine the attitude preference for the next moment. However, there are different positive effects between two individuals. The degree of influence is also inconsistent between individuals. It is worth discussing in what form the surrounding environment impacts the individual. The J–A model provides an idea. We consider parameters such as individual followership, the strength of network relationships, etc., and assign the corresponding values by specific rules. To obtain the final polarization algorithm, we need to combine the law of related network distribution and choose the BA network as the agent adjacency model.
- (3) The group communication behavior between individuals makes the opposing sides continuously reinforce their views and gradually form the polarization of online opinions. The evolutionary results show that there is a clear polarization phenomenon at the beginning of the evolutionary stage. As the polarization process proceeds, the fluctuations level off, and the level of inter-individual following is low. It fails to reach the influence threshold, causing the attitudes of several individuals to stay in the initial state. Moreover, the difference in network structure makes the change of individuals always within a local interval, even when some individuals have difficulty making a choice and remain neutral. With the deepening of polarization, the proportion of

network individuals with absolute attitudes increases significantly, and the trend of polarization of double-linked opinions becomes more and more obvious. On the other hand, the proportion of Internet individuals who expressed neutral attitudes remained low and slightly changed. With the development of the event, the Internet users' ideas about the event become more and more distinct. Based on the results of this simulation, we give policy recommendations and discuss the problems in the experimental process in the following sections.

6.2. Policy Recommendations

With the prevalence of the trend of network intelligence and the expansion of network coverage, the predicament of information blocking has changed, and human networked society has risen rapidly. At the same time, a large amount of true and false information causes confusion, and false information spreads to the public, misleading moral values, laying down hidden danger for the maintenance of a harmonious environment for online public opinion. The government should take timely measures for the real-time, changing, networked environment to create a just and harmonious network environment for citizens and eradicate some unsettling hidden social dangers. To better cope with the polarization of network public opinion, this paper puts forward the following suggestions:

- (1) Improve the public opinion monitoring mechanism and build a harmonious network order

Internet public opinion is easy to use to guide the views of the masses, and if it is not properly supervised, it is easy to mislead the masses. Even the evolution of online public opinion may cause the masses to fall into a vicious circle of emotional or even group polarization. At the same time, freedom of speech on the Internet promotes the interaction of people's information and the collision of thinking and produces a situation in which false information and misinformation affect the emotions and thoughts of viewers to achieve the publisher's personal, bad goals. Therefore, the normality of online public opinion requires the cooperation of a strong monitoring mechanism to ensure the safety and order of cyberspace to some extent.

Although there are some online information reporting platforms, the government's use of them is inefficient, and even the processing and feedback of reporting information is not timely. The government should further improve the monitoring mechanism of network public opinion, not only relying on computer keyword recognition and big data processing technology but also mobilizing social forces to help network monitoring. Reporting information is more accurate than keyword recognition technology. Only by screening false information and reasonably guiding the direction of public opinion can some netizens who have difficulty judging information be protected and not misled or suffer some losses due to being deceived. The government actively participates in the governance of cyberspace. It will contribute to the harmonious co-construction and sharing of the network.

- (2) Rebuild the accountability mechanism for public opinion and crackdown on online anomie

Every occurrence of online public opinion polarization is a test of the government's credibility. Whether the government's accountability mechanism is sound and whether other aftermath measures are appropriate and timely will affect the government's image. As a public servant of the people, the government should actively investigate disharmonious factors or improper regulation by the government itself after the negative impact of online public opinion and safeguard the legitimate interests of the people. Only in this way can the credibility of the government be maintained.

The reconstruction of the network public opinion accountability mechanism is a necessary part of the government's governance of the network environment. Relevant government departments can start by conducting satisfaction surveys on network individuals related to the governance of the network environment. Through this post-mortem investigation, the government can clarify its image positioning in the eyes of the public and

understand its own problems. Only when the crisis is handled properly and the masses are satisfied will the negative influence of online public opinion be weakened, and the rebound will be avoided. At the same time, relevant government departments should also be held accountable for illegal acts that maliciously affect public order and damage the interests of others. Only by thoroughly cracking down on online anomie can we give a warning to criminals and, at the same time, put an end to attempts to conduct anomie because of luck.

(3) Guide netizens' values and transmit positive energy of public opinion

Internet public opinion has both positive and negative effects, and in the contemporary era, when the Internet closely links everyone, positive and negative emotions are more likely to spread and affect the public. Therefore, it is particularly important to guide netizens' values in a timely and positive manner and transmit positive energy of public opinion.

For different network groups, different measures should be taken to guide and pass on the "right medicine". Neutral internet users with many fans and high membership levels not only have a stable stance but also have a relatively large fan base, a high degree of activity, and a strong potential to control public opinion. Therefore, such network users can be used as a key group for public opinion dissemination guidance and polarization intervention, and the background of social platforms should increase efforts to maintain key groups, promote content that is conducive to guiding the development of netizens' values in a positive direction, and transmit positive public opinion. For some network users who pay less attention to hot events because their sources of information are relatively closed and single and passive, they can push comprehensive information to this group in a targeted manner, which helps the group form an objective and comprehensive understanding of hot events. At the same time, increasing the frequency of pushing positive content to enhance the positive experience of network users helps to transmit positive public opinion. In addition to paying attention to the above two parts of network users, high-impact and highly active groups can be found through background big data, and advanced technology can be used to seize the opportunity of positive information exposure and play and enhance leadership.

(4) Enhance the image of the government and maximize the interests of society

Internet public opinion is usually inextricably linked to civil rights, people's livelihood, and real society and the problems it exposes or the focus of discussion are related to this. Therefore, the government plays an important role in the management of network public opinion, which is conducive to enhancing the image and playing a more decisive role in maximizing the interests of society.

The government can use advanced technology and big data platforms to strengthen the management of the two major sources of information dissemination, official media and self-media. First, to standardize the operation and management mechanism of official media, we should put social benefits in the first place, standardize and restrain professional journalists, and correct the one-sided pursuit of traffic realization by some bad official media; at the same time, increase support for official media in terms of policies, funds and talent introduction. Second, through the public or industry associations to regulate the development of self-media in the right direction, guide them to carry out activities to produce and disseminate positive energy information, and create a positive atmosphere of public opinion and emotion among the public.

The government focuses on governing online public opinion to further consolidate its position and enhance its image. At the same time, in the process of standardization and guidance, the interests of all parties maximize social interests after the game.

This paper discusses the extreme model of public opinion based on the BA network, enriches the theory and method of polarization of online group attitudes, and predicts the network public opinion of hot events through empirical analysis, providing practical guidance for the intervention and guidance of network public opinion, which is of great significance for promoting the modernization of national governance capabilities.

6.3. Limitations and Future Research Directions

The present study has limitations in some respects. First, the model is a poor fit for the phenomenon of group attitude reversal. Internet public opinion changes rapidly, and as events develop, the final direction may not always be consistent with the initial state. People's attitudes will undergo drastic shifts in the process, which is often difficult to simulate by polarization models. Later, we will enrich and extend the model to address the conditions and trends of public opinion reversal. Second, the example simulation process includes only one public opinion event with a small sample size. This study can conduct practical simulation experiments by collecting different events and a larger sample size. By simulating multiple occasions, we can effectively improve the model's generalizability. Third, there is some bias in analyzing attitude values during the experiment. This study uses a machine-learning algorithm to assign attitude values to event comments. It is crucial to extract group attitudes from the text effectively. The algorithm's limitations primarily influence the encoding operation of the training set. If the algorithm can accurately extract attitude values from buzzwords, expressions, and punctuation, the error of the model will be significantly reduced. There is still room for improvement in the example algorithm piece.

Author Contributions: Conceptualization, T.C.; data curation, R.Z.; formal analysis, Y.Z.; software, W.D.; visualization, Y.Y. (Yuanyuan Yu); writing—original draft, Y.Y. (Yuanjian Ye). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Zhejiang Provincial Natural Science Foundation of China (Grant No. LY22G010003), the 2020 National Innovation and Entrepreneurship Training Program (No.202010353008), the 2020 Science and Technology Innovation Activity Program for College Students and New Seedling Talent Program of Zhejiang Province (No.2021R408050), and the 2019 University-level College Student Innovation and Entrepreneurship Project of Zhejiang Gongshang University (No.CX202002021).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jager, W.; Amblard, F. Uniformity, bipolarization and pluriformity captured as generic stylized behavior with an agent-based simulation model of attitude change. *Comput. Math. Organ. Theory* **2005**, *10*, 295–303. [\[CrossRef\]](#)
2. Barash, V.; Cameron, C.; Macy, M. Critical phenomena in complex contagions. *Soc. Netw.* **2012**, *34*, 451–461. [\[CrossRef\]](#)
3. Li, Z.; Tang, X. Threshold model of collective action. *J. Syst. Sci. Math. Sci.* **2014**, *34*, 550.
4. Gabbay, M.; Kelly, Z.; Reedy, J.; Gastil, J. Frame-induced group polarization in small discussion networks. *Soc. Psychol. Q.* **2018**, *81*, 248–271. [\[CrossRef\]](#)
5. Chen, T.; Rong, J.; Yang, J.; Cong, G. Modeling Rumor Diffusion Process with the Consideration of Individual Heterogeneity: Take the Imported Food Safety Issue as an Example during the COVID-19 Pandemic. *Front. Public Health* **2022**, *10*, 781691. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Barabasi, A.L.; Albert, R. Emergence of scaling in random networks. *Science* **1999**, *286*, 509–512. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Tao, L.; Zhong, C.; Chen, X. Overview of Rehydration Network Theory and Application Research. *Syst. Eng.* **2005**, *23*, 1–7.
8. Chen, T.; Yin, X.; Yang, J.; Cong, G.; Li, G. Modeling Multi-Dimensional Public Opinion Process Based on Complex Network Dynamics Model in the Context of Derived Topics. *Axioms* **2021**, *10*, 270. [\[CrossRef\]](#)
9. Zhou, T.; Bai, W.J.; Wang, B.H.; Liu, Z.J.; Yan, G. A brief review of complex networks. *Physics* **2005**, *34*, 31–36.
10. Wang, G. Dynamic Model and Simulation Analysis of Group Polarization of Online Public Opinion. *J. Intell.* **2012**, *3*, 20–24.
11. Chen, T.; Jin, Y.; Yang, J.; Cong, G. Identifying emergence process of group panic buying behavior under the COVID-19 pandemic. *J. Retail. Consum. Serv.* **2022**, *67*, 102970. [\[CrossRef\]](#)
12. Wang, F.; Zhang, S.; Wei, Y. Simulation of the Dynamics of Group Polarization Phenomenon in Online Public Opinion and Strategies for Regulating It. *Police Technol.* **2017**, *4*, 37–40.

13. Zhang, J.; Zhang, Y.; Zhao, F. Inter textuality: An Energy Logic Model for the Generation, Diffusion and Polarization of Self-media Network Public Opinion. *Inf. Sci.* **2020**, *38*, 3–9.
14. Hatton, T.J. Public opinion on immigration in Europe: Preference and salience. *Eur. J. Political Econ.* **2021**, *66*, 101969. [[CrossRef](#)]
15. Heizler, O.; Israeli, O. The identifiable victim effect and public opinion to ward immigration; a natural experiment study. *J. Behav. Exp. Econ.* **2021**, *93*, 101713. [[CrossRef](#)]
16. Blake, K.D.; Gaysynsky, A.; Mayne, R.G.; Seidenberg, A.B.; Kaufman, A.; D'Angelo, H.; Roditis, M.; Vollinger, R.E., Jr. US public opinion to ward policy restrictions to limit tobacco product placement and advertising at point-of-sale and on social media. *Prev. Med.* **2022**, *155*, 106930. [[CrossRef](#)] [[PubMed](#)]
17. Chen, T.; Peng, L.; Yang, J.; Cong, G.; Li, G. Evolutionary Game of Multi-Subjects in Live Streaming and Governance Strategies Based on Social Preference Theory during the COVID-19 Pandemic. *Mathematics* **2021**, *9*, 2743. [[CrossRef](#)]
18. Chen, T.; Qiu, Y.; Wang, B.; Yang, J. Analysis of Effects on the Dual Circulation Promotion Policy for Cross-Border E-Commerce B2B Export Trade Based on System Dynamics during COVID-19. *Systems* **2022**, *10*, 13. [[CrossRef](#)]
19. Chen, T.; Peng, L.; Yin, X.; Jing, B.; Yang, J.; Cong, G.; Li, G. A policy category analysis model for tourism promotion in china during the COVID-19 pandemic based on data mining and binary regression. *Risk Manag. Healthc. Policy* **2020**, *13*, 3211. [[CrossRef](#)] [[PubMed](#)]
20. Fang, J.; Wang, X.; Zheng, Z.; Bi, Q.; Di, Z.R.; Xiang, L. New Interdisciplinary Science: Network Science. *Prog. Phys.* **2007**, *27*, 239.

Article

Agent-Based Modeling of Consensus Group Formation with Complex Webs of Beliefs

Ismo T. Koponen

Department of Physics, University of Helsinki, 00014 Helsinki, Finland; ismo.koponen@helsinki.fi

Abstract: Formation of consensus groups with shared opinions or views is a common feature of human social life and also a well-known phenomenon in cases when views are complex, as in the case of the formation of scholarly disciplines. In such cases, shared views are not simple sets of opinions but rather complex webs of beliefs (WoBs). Here, we approach such consensus group formation through the agent-based model (ABM). Agents' views are described as complex, extensive web-like structures resembling semantic networks, i.e., webs of beliefs. In the ABM introduced here, the agents' interactions and participation in sharing their views are dependent on the similarity of the agents' webs of beliefs; the greater the similarity, the more likely the interaction and sharing of elements of WoBs. In interactions, the WoBs are altered when agents seek consensus and consensus groups are formed. The consensus group formation depends on the agents' sensitivity to the similarity of their WoBs. If their sensitivity is low, only one large and diffuse group is formed, while with high sensitivity, many separated and segregated consensus groups emerge. To conclude, we discuss how such results resemble the formation of disciplinary, scholarly consensus groups.

Keywords: consensus groups; agent-based model; web of beliefs

Citation: Koponen, I.T. Agent-Based Modeling of Consensus Group Formation with Complex Webs of Beliefs. *Systems* **2022**, *10*, 212. <https://doi.org/10.3390/systems10060212>

Academic Editors: Philippe Mathieu, Juan M. Corchado, Alfonso González-Briones and Fernando De la Prieta

Received: 11 October 2022

Accepted: 8 November 2022

Published: 9 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The formation of groups with shared beliefs, opinions, and views has been and continues to be a topic of great interest, discussed in sociology, political science, communication, and organizational science, as well as studies focusing on structure of science (see, e.g., [1–5] for the diversity of topics and areas of studies). In all these cases, one key issue is to understand the dynamics which drive the segregation and consolidation of groups, even in conditions where communication and sharing of beliefs is common and frequent (for reviews, see [3–5]).

The computational modeling of opinion group formation [5–7], the formation of collaborative groups and collective decision making [2,8,9], as well as disciplinary fragmentation and progress in science [10], has shed light on the social dynamics behind group formation, segregation, and consolidation, often revealing the unexpectedly simple but self-reinforcing interactions behind such complex phenomena. Consequently, consensus group formation and opinion adoption, and their change and evolution, have been modeled using a variety of idealized models, many of them founded on one or another theoretical view about social influence and interaction or social learning, some of them being computational renderings of empirical findings. The models of opinion dynamics are as diverse as are the their theoretical underpinnings and intended scopes of applications. However, many of the models of opinion dynamics, where formation of consensus groups is of interest, can be classified in three groups [3]: (1) models of assimilative social influence; (2) models with similarity biased influence, and; (3) models with repulsive influence. In addition, it is possible to recognize models that are hybrids of these three classes [3]. Different well-known models in each class, their theoretical background and motivation, and most well-known computational implementations and empirical applications and justification (when it exists) are reviewed in detail elsewhere [3–5], and therefore we provide here only a brief summary

of the most important aspects of models with similarity bias to the extent needed to put in perspective the model introduced here.

Models with similarity bias do not assume a structurally fixed connection between agents, but instead, agents can interact if they have sufficiently similar opinions (or beliefs) or if they are sufficiently similar regarding some other pertinent feature. If the similarity (opinions or other preferred feature) is too distant, interactions are no longer possible. Such a threshold of interaction can be interpreted as a confidence to interact and modify one's opinion and can be assumed to arise from a variety of psychological or sociological reasons [3–5]. Therefore, many similarity bias models are referred to as bounded confidence models [11,12] owing to the existence of a kind of a confidence threshold to interact. These models typically give rise to persistent opinion clusters, where agents are similar and dissimilar to agents in other clusters. In these models, the cluster formation is an outcome of similarity bias homophily; the more stringent the threshold for required similarity, the more numerous are the segregated non-interacting clusters. In most bounded confidence models, the threshold is sharp and deterministic, and stochastic variation is included by adding a stochastic noise [11,12]. Interestingly, the addition of noise may fundamentally affect the consensus cluster formation [3–5]. The role of noise and its non-trivial effects in deterministically thresholded bounded confidence models suggests that to model the bounded regions of interactions due to similarity bias, genuinely stochastic and probabilistic rules to decide whether or not the interactions happen are sometimes preferable [13,14]. Finally, important and interesting recent generalizations of bounded confidence models are models where opinions or beliefs are multidimensional [15–17], or in one way or another, more complex structures of beliefs and related opinion elements [2,7,9,18]. Such models allow more realistic modeling of complex opinions, give rise to richer dynamics than one-dimensional bounded confidence models, and, moreover, the emerging clusters of opinions are in these cases more diverse than those found in one-dimensional models.

Paralleling the above briefly summarized recent generalizations of similarity-biased models of consensus formation, the agent-based model (ABM) proposed here is meant to be another step towards generalizations applicable to model complex sets of beliefs, where interactions between agents and dynamics of the change in beliefs are modeled as a genuine stochastic process. The model, however, adopts the simple view where agents modify their views by acquiring and accommodating their sets of views present in to a collection of all agents; no new elements emerge.

The ABM proposed here, given its restriction to describe the creation of new elements of opinion beliefs, and instead describing the evolution of consensus groups with complex sets of beliefs, has two possible areas of applications. One area of interest is the formation of disciplinary scholarly groups and schools, along with their characteristic ways of using scientific terms and forming various research programs [19–22]. Research fields always contain disciplinary groups where key scientific terms differ, and the same terms may be used differently in discussing and framing the key problems within the field [21,22]. In this case, within the established paradigms of research, new disciplinary groups are often formed within the existing fields. Such strong disciplinary fragmentation seems to be particularly apparent and typical in the human and behavioral sciences [23–25]. Another situation of interest, where the creation and generation of new knowledge is not necessarily of primary interest, but where differing disciplinary views about thematic topics can be recognized, is related to the disciplinary views of science education scholars [26,27] as well as science students, where student groups may have consensus views that differ from those of other student groups views, even when they have used the same study materials [28,29]. To address situations in which the knowledge or meaning structures of interest are complex systems of terms, concepts, or conceptions characteristic of the disciplinary group, it seems appropriate to use the expression “webs of beliefs” (compare e.g., ref. [30]), to be referred to briefly as WoBs in what follows. In this study, we approach such a disciplinary group formation through convergence and consolidation of WoBs, where the dynamics are driven by similarity and consensus seeking, without posing explicit bounds of confidence to

constrain interactions (as in so-called bounded confidence models, [1–3]). In the ABM presented here, agents possess generic WoBs in the form of a complex network. Agents compare their WoBs and exchange bits and pieces of them, guided by similarity-seeking dynamics (i.e., homophily). The comparison-triggered adjustment of WoBs leads to their convergence but also divergence, and thus to the formation of segregated disciplinary consensus groups. The model is highly idealized, generic, and simple, but as discussed in the final section, it has features that resemble the real situations of disciplinary group formation. Thus, the model presented here is a first attempt towards agent-based modeling of consensus group formation where views and opinions are complex webs of beliefs.

2. Materials and Methods

The computational model presented here is an agent-based multi-optional model for the formation, consolidation, and segregation of consensus groups based on agents' webs of beliefs (WoBs). The dynamics of the model are driven by agents' repeated comparison of their WoBs, guided by the utility of adjusting the WoB for better mutual similarity. We describe first the constructions of agents' WoBs, second the multi-state probabilistic model of selection of partners for comparison, and third, update dynamics for the change in WoBs. Simulations are carried out using an event-based roulette-wheel method [31,32]. Symbols and their meaning in the study are summarized in Table 1.

Table 1. Summary of symbols and abbreviations used recurrently in the text and figures. In the sub-indexes, ξ and γ refer to a pair of agents, Ens. avrg. denotes ensemble averages, and $\text{Dim}[\{\cdot\}]$ refers to a dimension of set.

Symbol/Abbreviation		Symbol/Abbreviation	
$\{e\}$	Set of edges ($E = \text{Dim}[\{e\}]$)	P_{\pm}	Prob. to add/delete elements
$\{v\}$	Set of vertexes ($V = \text{Dim}[\{v\}]$)	β	Sensitivity to similarity
$P_{\xi\gamma}$	Probability of interaction	β^*	Confidence to add/delete
$S_{\xi\gamma}$	Similarity function	N	Ens. avrg. of cluster number
S_E	Similarity, edges	R	Ens. avrg. of relat. occupancy
S_V	Weighted adjac. matrix	S	Ens. avrg. of similarity
Δ_{\pm}	Utility to add/delete elements	H	Entropy of cluster distribution

2.1. Model of Webs of Beliefs (WoBs)

The WoBs of interest, which mimic real scholarly conceptual or semantic structures, can be characterized as complex in the sense that they consist of several elements (vertexes or nodes) connected (by edges or links) in complex ways to their other elements. Such WoBs have a distribution of vertexes with broad difference in their connectivity; some vertexes have a high connectivity, while many are loosely connected (see e.g., [28,29,33,34]). Consequently, appropriate WoBs can be characterized as sparse complex networks with about 100 vertexes, with low average degrees of about 3 or 4 but with very few nodes having larger degrees up to 10.

Generic initial WoBs of agents are obtained by sampling a larger template network with desired properties, generated by a generative model previously introduced to produce networks with broad distributions of connectivities of vertexes [35]. In constructing the generic networks, they are pruned by removing auxiliary vertexes possessing only one edge, so that to only two cores (nodes with at least two edges) are taken into account. The template network has approximately 200 vertexes (i.e., nodes) and 400 edges (i.e., links), with the degrees (i.e., number of attached links) of the vertexes distributed according to inverse power law, with an inverse power close to 3.0 (see Appendix A for details of rationalization of this choice). With these parameters, the average degree of vertexes is about 3 to 4, with very few vertexes having a degree of about 8 to 12. From the template networks, a set of initial edges $E_0 = 120 \pm 20$ are drawn at random, and connected networks (discarding unconnected parts) are formed to serve as individual WoBs for each agent. With these choices, WoBs contain about 40–100 vertexes (in some rare cases only about 20)

and are always sub-networks of the template. The WoBs thus obtained from the template are stochastically generated and have in each case a slightly different detailed structure. The details of the generative model, briefly summarized in Appendix A, are reported elsewhere [35] and are of no further interest here.

2.2. Interaction Dynamics of Agents

The interaction between agents in the agent-based model (ABM) introduced here comprises N agents, where one agent ξ selects an agent γ to interact among $N - 1$ other agents. The selection is based on the similarity $S_{\xi\gamma}$ of lexicons between the agents. The probability $P_{\xi\gamma}$ that agent ξ selects agent γ is assumed to follow the (Gibbs-like) probability distribution (compare with refs. [36–38])

$$P_{\xi\gamma} = \frac{S_{\xi\gamma} \exp[\beta S_{\xi\gamma}]}{\sum_{i \neq \xi} S_{\xi i} \exp[\beta S_{\xi i}]}, \quad \xi \neq \gamma \tag{1}$$

where β is a parameter related to the *sensitivity* to similarity, with $\beta < 1$ (here, in practice $\beta \approx 1$) indicating low sensitivity (i.e., high noise or randomness) and $\beta \gg 1$ high sensitivity (i.e., low noise or randomness). The prefactors in Equation (1) ensure that at the limit $\beta \rightarrow 0$, the decision probabilities attain values corresponding to the ordinary rational, non-probabilistic choice (for details, see [37,38]).

The similarity of the agents is defined simply as the ratio of the number of shared elements (vertexes and edges) to all elements. Denoting the set of elements as $\{X\} = \{v\}$ for vertexes and $\{X\} = \{e\}$, the similarity of agents ξ and γ based on either edges or vertexes is given simply as a ratio of non-shared to shared elements,

$$S_{\xi\gamma}[X] = \frac{\text{Dim}[\{X\}_{\xi} \cap \{X\}_{\gamma}]}{\text{Dim}[\{X\}_{\xi} \cup \{X\}_{\gamma}]}, \tag{2}$$

where X denotes either edges (e) or vertexes (v) and $\text{Dim}[\cdot]$ means dimension (number of elements) in a given set. In defining the similarity, we chose to keep it symmetric and as simple as possible, although more elaborate definitions are possible, for example, taking into account the role of asymmetry and different number of non-shared elements. However, the definition of similarity is always a matter of choice, and no unambiguous definition seems to be possible because choices need to be made about what features are taken into account in similarity, as discussed in detail in, e.g., ref. [39].

We assume that both vertex and edge similarities need to have high values for a high similarity, and thus, in the effective similarity used in simulations, the effective similarity is taken as a geometric mean of vertex and edge similarities.

$$S_{\xi\gamma} = \sqrt{S_{\xi\gamma}[e] S_{\xi\gamma}[v]} \tag{3}$$

This choice of defining the similarity is a trade-off between simplicity and taking enough detail into account to characterize the WoBs. Here, given the generic nature of the WoBs, a simple symmetric similarity $S_{\xi\gamma} = S_{\gamma\xi}$ based on the counting of elements and constrained in the range of values from zero to one is satisfactory for the present purposes. In fact, we tried out more elaborated similarity definitions, but in the present ABM, they had little effect on the final results.

The selection criterion in Equation (1) with similarity defined as in Equation (2) prefers similar agents (i.e., homophilic preference) and, in that sense, it closely resembles the selection criteria in bounded confidence models and their variants [1,3], where homophilic cut-off criterion constrains the possibility of interactions. The present model, however, belongs to a class of probabilistic similarity-biased models [3,13], where no interaction possibilities are ruled out a priori, but partners with high similarity are prioritized.

2.3. Updating the WoBs

In the interaction events, agents update their WoBs, seeking to improve their consensus. When agent ζ has decided on the partner γ it will interact with, the update of WoBs for optimized communication takes place in two steps. First, agents ζ and γ pick out a common vertex, which is a center element in their interaction (i.e., shared term or concept for their belief). Second, they check the neighborhood of that selected vertex and check the number of available new edges to neighborhood vertexes their partner possesses but the agent itself does not yet possess (i.e., possible new elements and connections the agent can acquire to adopt a new edge and possibly a new vertex connected to it). Both agents have then two choices of how to increase mutual similarity: either to add a new edge, or to delete an edge not possessed by the interaction partner. The decision to add or delete an edge is made on the basis of the utilities, based on the changes in the similarity, which are taken to be changes in similarity by addition or deletion of an edge or a vertex (contained in the neighborhood of the common vertex). The advantage of such a simplified procedure is that utilities of addition can be estimated through changes in similarities, when number of elements for one agent changes while remaining intact for the other agent. Now, four possibilities are available, either agent γ changes its WoB or agent ζ does, but only one event happens at a time. In both cases, however, the changes are computable as a difference $\Delta_{\pm}[X] = S[X \pm 1] - S[X]$ for addition (+) and deletion (−), respectively, where indexes referring to agents are dropped because of symmetry in regard to the agents. For faster computation, the differences are approximated by the first linear terms, which are obtained through direct calculation, in the form

$$\Delta_+[X] = (1 - S[X]) \text{Dim}[\{X\}_{\zeta} \cup \{X\}_{\gamma}]^{-1} \tag{4}$$

$$\Delta_-[X] = S[X] \text{Max}[1, \text{Dim}[\{X\}_{\zeta}^C] + \text{Dim}[\{X\}_{\gamma}^C]]^{-1} \tag{5}$$

where X denotes either vertexes or edges and notation $\{\cdot\}^C$ complements of sets. Note that in all expressions containing S and Δ_{\pm} or related to them, indexes referring to agents are dropped in what follows. In Equation (5), the minimum value allowed in the denominator is 1, in order to prevent division by zero (this occurs rarely and has no consequences on results). In case only an edge is added/deleted but no new vertex becomes added/deleted, utilities are simply for edges ($X = e$). In the case that the added/deleted edge contains a new vertex, the effective utility Δ_{\pm} corresponding to addition (+) or deletion (−) of a vertex and its neighborhood is taken to be the geometric average of utilities for $X = e$ and $X = v$.

$$\Delta_{\pm} = \sqrt{\Delta_{\pm}[e] \Delta_{\pm}[v]} \tag{6}$$

Probabilities of addition and deletion are then given as

$$P_{\pm} = Z^{-1} \Delta_{\pm} \exp[\beta^* \Delta_{\pm}] \tag{7}$$

where parameter β^* has the role of confidence of the decision to add or delete on the basis of utility. The factor $Z = \Delta_- \exp[\beta^* \Delta_-] + \Delta_+ \exp[\beta^* \Delta_+]$ is the normalization factor. The rationalization to include prefactors is similar to the case of probability in Equation (1) selecting the partner for interactions. The choice is always between addition and deletion; staying intact (which is not counted as an event) is not included. This is related to the choice that simulations are event-driven; only events that make a difference (i.e., change the state of the agent) are taken into account. Note that in each interaction event, only one outcome of the four possibilities is allowed.

2.4. Simulations

Simulations of the ABM are based on the probabilities of selection $P_{\zeta\gamma}$ in Equation (1) an P_{\pm} in Equation (7), with each event of an agents' interaction consisting of both updates. Simulations are event-based, consisting of sequences of events $\tau = 1, 2, \dots, \tau_{\text{MAX}}$ events. At

each instant, when the value of τ increases by 1, it is decided: (1) which agents are going to interact and (2) whether an edge (and possibly, a vertex) is added or deleted. Each of the event selections is carried out by the roulette-wheel method [31,32]. In the roulette-wheel method, a discrete set of N possible events k with probabilities p_k is arranged with cumulative probability

$$\Phi_k = \sum_{i=1}^k p_i / \sum_{i=1}^N p_i . \quad (8)$$

The event k is selected if a random number $0 < r < 1$ falls in the slot $\Phi_{k-1} < r < \Phi_k$. In case (1), the probabilities p_k are given by $P_{\zeta\gamma}$ in Equation (1), while in case (2), one has only two probabilities $p_k = P_{\pm}$ defined by Equation (7). The roulette-wheel method is thus entirely event-based, where the occurrence of events is predicted on the basis of cumulative distribution. Consequently, there are no constant time-like intervals between events. All simulations are carried out for $N = 25$ agents, with $\tau_{\text{MAX}} = 50 \cdot 10^3$ events, corresponding to about 30 updates for each edge in a set of WoBs of 25 agents. However, only a fraction, usually about 2–3%, eventually leads to addition or deletion of an edge, since after the onset of formation of consensus groups, many neighborhoods are already identical. In practice, this means from 40 to 60 changes per agent in the course of simulations, and as is seen, a stabilized situation is reached well before τ_{MAX} is reached. In the simulations, only one affinity distribution is used, corresponding to an inverse power of 2.9, close to the marginal value of 3 (see Appendix A for details). The simulations are repeated for the ensemble of $36 = 6 \times 6$ different initial states, sampled for six different initial templates and each six times for different initial WoBs. Simulations and numerical computations are realized using Mathematica [40].

2.5. Representation of Data

The simulations track the evolution of agents' WoBs, and on that basis, agents' similarities based on shared edges and vertexes are the outcome of the simulations. The agents are classified in clusters according to the effective similarity $S = \sqrt{S_E S_V}$ and geometric average of size $\sqrt{E V}$, where $E = \text{Dim}[\{e\}]$ and $V = \text{Dim}[\{v\}]$. In finding the clusters, we use Mathematica's [40] DBSCAN algorithm, which partitions the datasets into clusters using density-based classification with noise [41]. DBSCAN proved to be a reliable and fast method for finding clusters not constrained by a prefixed number of clusters for the data generated by simulations. When clusters are detected, the number of clusters N and agents in a given cluster are counted to obtain the relative occupancy R as the average value of the fraction of agents that belongs to a given cluster.

In addition to cluster statistics condensed in average values N and r and their standard deviations, the Shannon entropy H of cluster distribution is calculated, given by

$$H = - \sum_{ij} p_{ij} \log[p_{ij}] \quad (9)$$

where $p_{ij} \neq 0$ is the probability density of agents with discretized values of similarity and size, $(S, \sqrt{E V})$. The entropy H is useful to monitor the consolidation of the cluster distribution.

In what follows, cluster formation, consolidation, and segregation are monitored by using the quantities N , R , S , and H . Of these, N , R , and H are of the greatest interest, because owing to the choice of simulation parameters to maintain average similarity close to a constant, similarity S is always nearly the same in the stabilized states, and thus, contains little information about the differences of cluster distributions in the stabilized state.

3. Results

The simulations are carried out for 25 agents, which possess different but partially overlapping WoBs about $E_0 \pm \Delta E_0$ (here $E_0 = 120$ and $\Delta E_0 = 20$) edges, which are randomly

selected parts of the same template network consisting of 200 vertexes (i.e., nodes) and 400 edges (i.e., links). However, only 2-core (i.e., each vertex has at least two edges attached) and fully connected WoBs are included as initial states, and thus the initial number of vertexes V_0 in the WoBs, when pruned to connected networks, varies roughly from 40 to 100 (with a few exceptional cases of 20 to fewer than 40 vertexes). The parameter affecting mostly the evolution of WoBs during the simulations is the sensitivity β to similarity of partners (i.e., sensitivity to homophily). Note that simulations are carried out for this parameter in the range from 10 to 300, but results are reported for scaled values $\beta \rightarrow \beta/10$ to allow easy presentation in logarithmic scale. The parameters E_0 and ΔE_0 are the next most important in affecting the outcome of simulations. The effect of these parameters is tested by keeping ratio $E_0/\Delta E_0$ fixed and changing the absolute values by $\pm 20\%$ by scaling them with a factor of $\eta = 0.8$ and 1.2 .

In simulations with unfolding interaction events τ , the agent selects another agent to compare and adjust its WoB to better match the other agent's WoB. The selection probability of the agent to interact depends on the similarity between the agents and the agent's sensitivity β to similarity in that selection. In comparison events of WoBs, interacting agents either adopt or delete an edge to increase their mutual similarity.

In Figure 1, examples of initial and final WoBs are shown. In all cases, the number of vertexes V (and edges) increase because the utility function favors the addition of edges and vertexes, thus favoring the growth of WoBs. However, initial and resulting final WoBs may have quite different sizes. Note that the initial WoBs are always 2-core networks, but during their evolution, singly connected vertexes may appear. In the final state ensemble, averaged values characterizing WoBs are not changing anymore, although slight changes in details of the structure may take place.

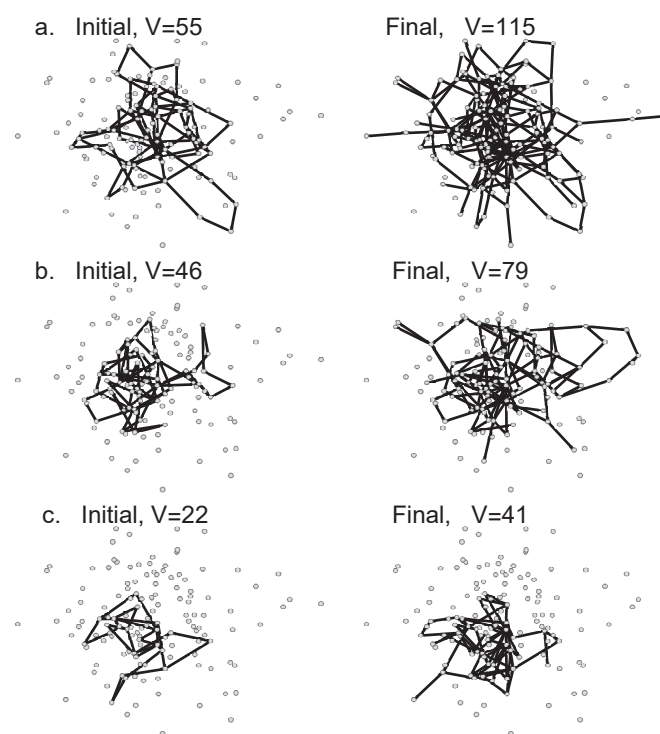


Figure 1. Three examples of agents' webs of beliefs (WoBs) (a–c) are shown for initial WoBs (at left), which in interaction events evolve to the final WoBs (at right). In all cases, WoBs are projected on a common aggregated template (with unconnected vertexes shown).

The dynamic changes in WoBs, driven by the similarity bias of interactions, eventually leads to a formation of consensus clusters. In Figure 2, an example of a similarity cluster in the initial stage is shown, and then in the stabilized final stage for agents making confident choices of interaction partners with $\beta = 30$ (recall that this is a scaled value $\beta/10$). The

stabilization of the cluster distributions with $\beta = 30$, as shown in Figure 2b, is obtained with about 30,000 interaction events in a group of 25 agents, each of them possessing about 100 edges on average (originating from a sampled set 120 edges on average). Roughly, this means that agents need about 10 interaction events for each edge if they are to reach consensus states. Note that in what follows, we report interaction events τ scaled by a factor of 1000.

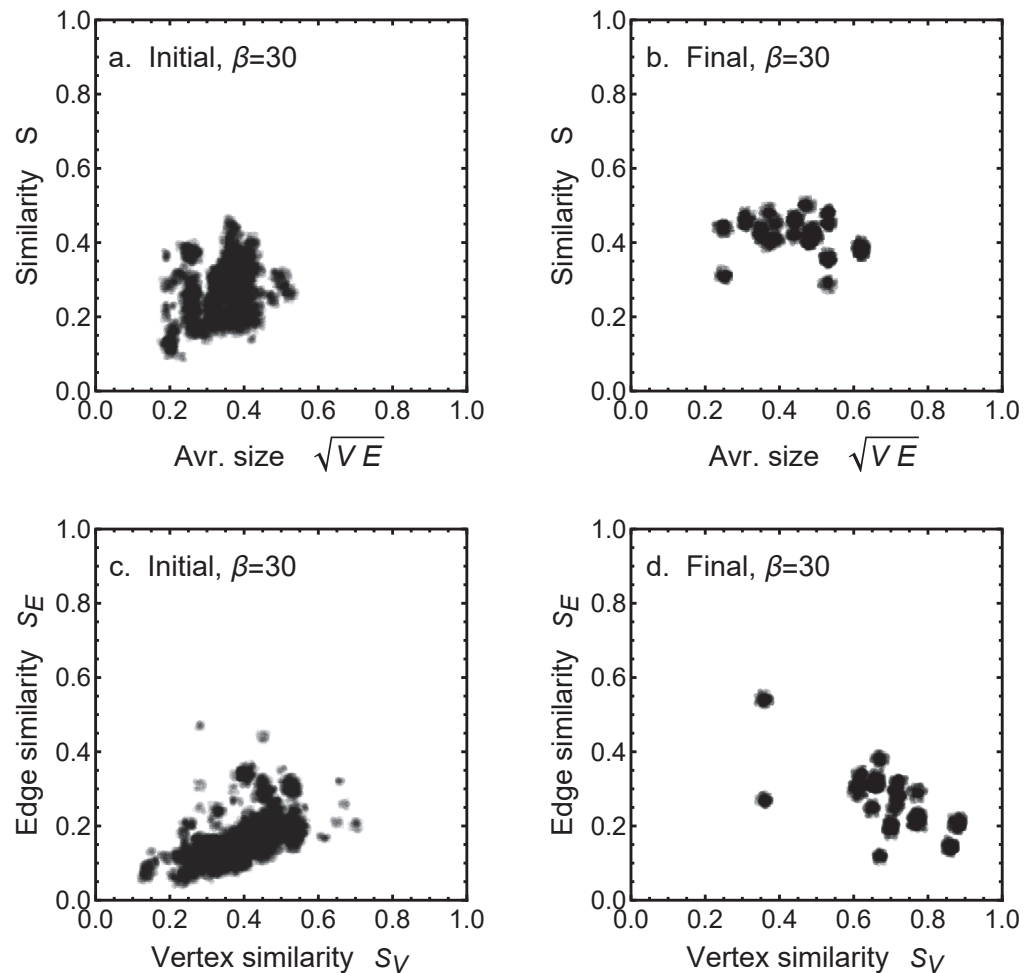


Figure 2. An example of formation of consensus clusters in the case of highly confident agents with $\beta = 30$ (with scaled β , see main text). In (a,b), the horizontal axis shows the average size \sqrt{EV} normalized to maximum $\sqrt{E_0V_0}$, and the vertical axis shows the effective similarity S . In (c,d), the horizontal axis shows the similarity of vertexes S_V , while the vertical axis shows the similarity of edges S_E . The similarities shown are aggregated over the last few stable ensembles corresponding to the range $45 < \tau < 50$.

In Figure 2a,b, clusters are shown as a density plot of effective similarity S versus the (geometric) average size \sqrt{EV} of the WoB, normalized to a maximum $\sqrt{E_0V_0}$, with both values discretized into bins of 0.01 ranging from 0 to 1. It is seen how the initial diffuse cluster (Figure 2a) segregates and consolidates to several smaller ones; in practice, six separate clusters (Figure 3a), as resolved by DBSCAN. In the case shown in Figure 2b, the DBSCAN routine finds seven or eight clusters (depending on parameters), but due to thresholding, which ignores very low-populated clusters, six remain to be counted as significant clusters. In Figure 2a, DBSCAN detects two clusters. In Figure 2c,d, the same case is shown, but now as a density plot of edge similarity S_E and vertex similarity S_V of the WoB. The similarities and sizes shown in Figure 2 are aggregated over the last few stable ensembles corresponding to the range $45 < \tau < 50$.

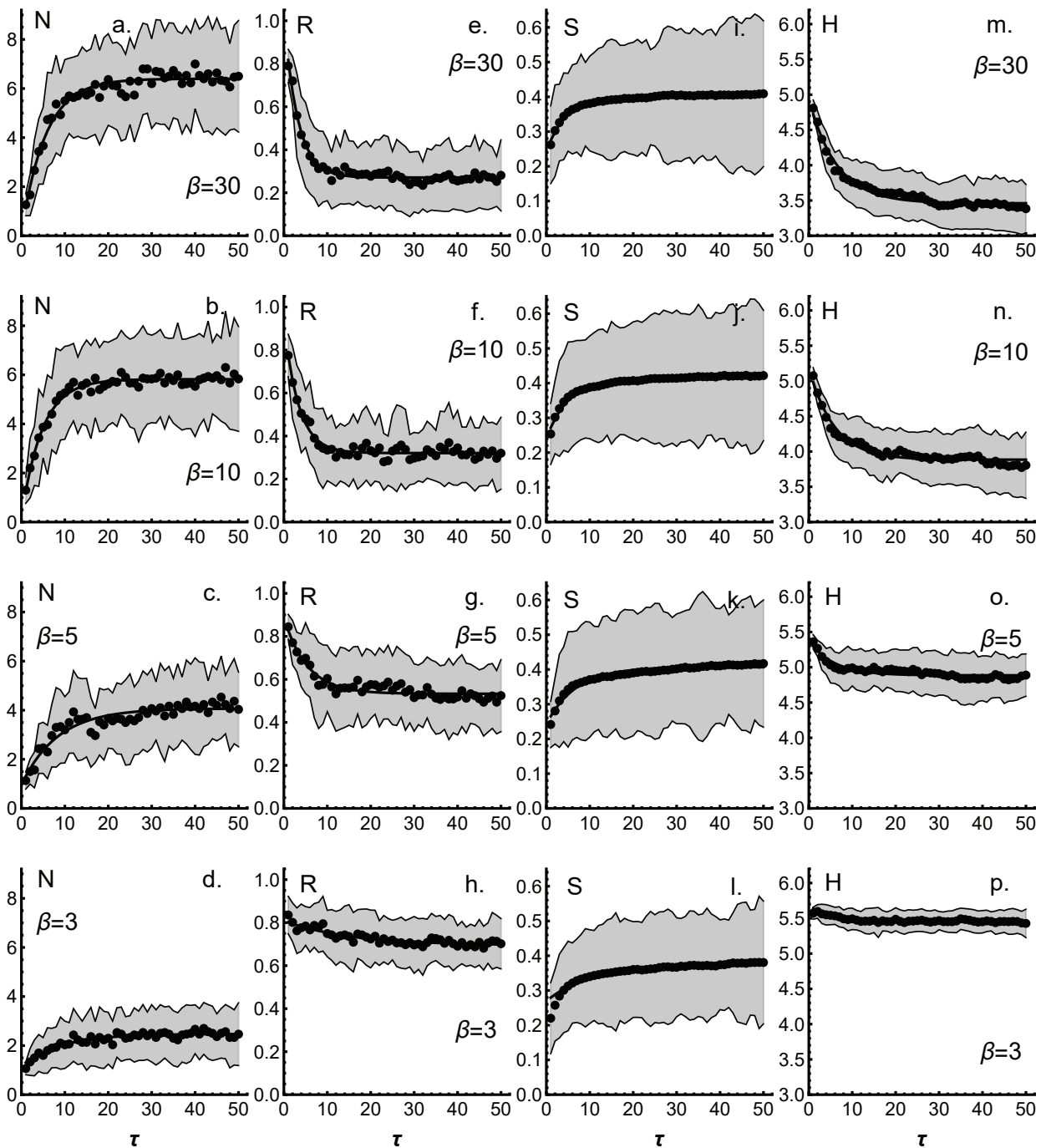


Figure 3. Average number of clusters N (a–d), relative occupancy R (e–h) and similarity S (i–l) of clusters, and entropy H (m–p) of cluster distribution for different strengths of sensitivity β to similarity (scaled by a factor of 10) from low ($\beta = 1$) to high ($\beta = 30$) sensitivity as a function of update events τ (scaled by a factor of 1000). Average values are given in black data points, and gray borders denote the standard deviations. Thin lines (not well-visible) are exponential fits to average values.

In reading the relevant information from Figure 2, it should be noted that while the high-similarity groups are clearly seen, so are the groups where agents are dissimilar (one agent can well belong to both groups, since similarity is a pairwise property). Therefore, we see not only groups where similarity has increased but also groups where it has decreased. Such behavior is a natural hallmark of the segregation of consensus groups.

Consensus cluster formation proceeds gradually from the diffuse initial state to a stabilized final state, depending on the number of interaction events. Figure 3 shows the dy-

namics of cluster formation as it is monitored through number of clusters N (Figure 3a–d), relative occupancy (average fraction of agents in a cluster) R (Figure 3e–h), and average similarity S (Figure 3i–l) of clusters, as well as through entropy H of cluster distribution (Figure 3m–p). Results are shown for different strengths of sensitivity to similarity β (scaled by a factor of 10) from high ($\beta = 30$) and to low sensitivity ($\beta = 3$) as a function of update events τ (scaled by a factor of 1000). For $\beta \leq 3$ and $\beta \geq 30$, results remain essentially the same as for the corresponding limiting values shown in Figure 3. In all cases, dynamic behavior can be fitted to an exponentially decaying function (exponential fits are shown in Figure 3 but are barely visible).

The average number N of consensus clusters in the final, stabilized state (Figure 3a–d) depends on the sensitivity β of agents to similarity in making choices to interact with other agents. For high-sensitivity $\beta = 30$ (scaled), the number of clusters in stabilized state is on average about six (see Figure 3a), but fluctuations as measured by the standard deviation of the ensemble averages are large. The relative occupation R of clusters shown in Figure 3e is on average about 0.30, corresponding to six or seven agents in a cluster. Increasing the level of sensitivity by increasing values of β does not change the situation, and thus results for $\beta = 30$ appear to represent the most extreme segregation found in the group of 25 agents. As seen later, changing the number of edges and vertexes in initial WoBs about by $\pm 20\%$ also leaves the number of clusters in this case nearly intact. When the sensitivity to similarity in making choices becomes smaller, and values of β are reduced (Figure 3b–d), the number of clusters decreases steadily, reaching the lowest attainable values of about two clusters on average for $\beta = 3$. Additional reduction in β appears not to lead to a definite monocluster situation, although single clusters become more abundant. As is seen from Figure 3f–h, the relative occupancy R of clusters follows roughly inversely the behavior of the average number of clusters, so that the product $N \times R$ remains roughly a constant, indicating relatively uniform distribution of agents in different clusters. In all cases, the similarity S (Figure 3i–l) has nearly the same average value, owing to the choice of parameter $\beta^* = 10.0$ controlling the ratio of utility of addition to deletion. The larger the value of β^* , the larger the bias towards addition, and thus growth of clusters. The choice to keep the bias from growth moderate, and average similarity close to a constant value, makes the interpretation of consensus cluster formation easier and rules out the possibility that the formation of high-similarity clusters is mainly due to bias towards systematically higher average similarities.

The entropy H of the cluster distribution (Figure 3m–p) relaxes considerably more slowly to a stable value in comparison with cluster number N and relative occupancy R . This indicates that consolidation of clusters and segregation of clusters occurs only partly simultaneously, and within the clusters, WoBs continue to evolve more similarly, as also indicated by the slow convergence of similarity S ; clusters consolidate without further segregation.

The stabilized, final values of cluster number N , entropy H , and the average occupancy R as a function of the sensitivity β is shown in Figure 4 for a range of values from $\beta = 1$ to $\beta = 30$, on a (natural) log-scale. It is now seen (Figure 4a) that if the sensitivity to similarity is high enough ($\beta > 10$), cluster formation takes place, and in the group of 25 agents about 6–7 are formed, with roughly equal numbers of 5–6 agents ($R \approx 0.3$) in a cluster (Figure 3c), with entropy $H \approx 3.5$, which roughly corresponds to cases such as the one shown in Figure 2b. However, in all cases, variations around mean values are large, as indicated by the error bars showing the standard deviation of values. In all cases, a transition to single cluster takes place around the value $\beta \approx 5$. After the transition, only two clusters are obtained on average, with high entropy ($H \approx 5.5$), roughly corresponding to two large and diffuse clusters of similar type as an initial cluster, shown in Figure 2a.

The results in Figure 4 show that the high sensitivity of agents to similarity in selecting similar partners (i.e., strong homophily) for interaction and for updating their WoBs invariably leads to the formation of segregated consensus groups. In addition to parameter β regulating the sensitivity to similarity, the parameter β^* for regulating agents' confidence

in decisions to add or delete edges affects the dynamics. Here, the value $\beta^* = 10.0$ is chosen to maintain the average similarity in stabilized stage nearly a constant. Parameter β^* regulates mainly the number of events needed for the relaxation and progress in harvesting new vertexes and edges. At low values of $\beta^*/E_0 < 1$, there is no average growth in similarity, at high values of beta $\beta^*/E_0 \gg 1$ in the region of a confident decision to add or delete, the growth is maximal, constrained only by the new options allowed for the addition of edges within the consensus clusters. In particular, β^* has no effect on the transition from single to multiple clusters.

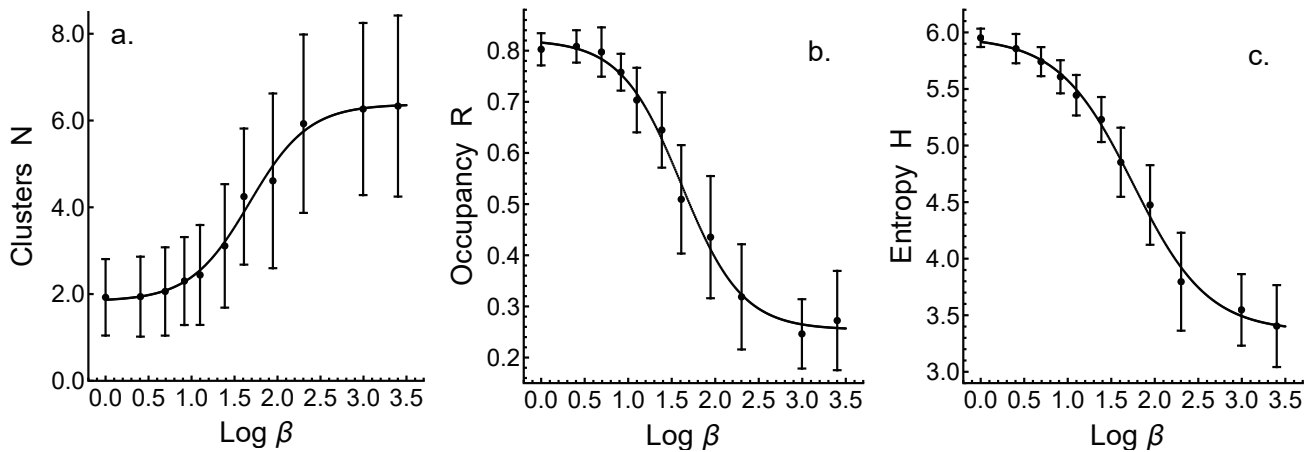


Figure 4. Average number of clusters N (a), relative occupancy R (b), and entropy H (c) for different values of parameter β (scaled) on (natural) logarithmic scale. Results are for clusters in the final stabilized region of the formation of consensus clusters. Error bars correspond to standard deviations.

Finally, the effect of the initial number of edges E_0 in WoBs and their variation ΔE_0 is checked by changing their absolute values by $\pm 20\%$, scaling both values by a factor of $\eta = 0.8$ and 1.2 but keeping the ratio $E_0/\Delta E_0$ fixed. Figure 5 shows the evolution of cluster number N (Figure 5a–c), occupancy R (Figure 5d–f), and entropy H (Figure 5g–i) for the altered configurations with high sensitivity β (the most interesting case with a high number of clusters) for values $\eta = 1.2$ (the upper row) and $\eta = 0.8$ to be compared with the results shown in Figure 3, corresponding to $\eta = 1.0$ (reproduced in the middle row). The evolution of similarity is essentially similar to the results shown in Figure 4 and is thus omitted in Figure 5. As can be seen from the results, the increase in size of the WoB corresponding to factor $\eta = 1.2$ slightly increases the number of clusters, but relaxation to steady state then requires more interaction events. This is as expected, because agents now have more choices available (i.e., more edges and vertexes) to change their WoBs, and thus more comparison events are needed to reach consensus. When the initial WoBs are smaller, corresponding to the choice $\eta = 0.8$, the number of consensus clusters is slightly smaller than for $\eta = 1.0$, and the events needed for relaxation are fewer. Qualitatively, for other values of β , the results remain essentially similar to the cases shown in Figure 3.

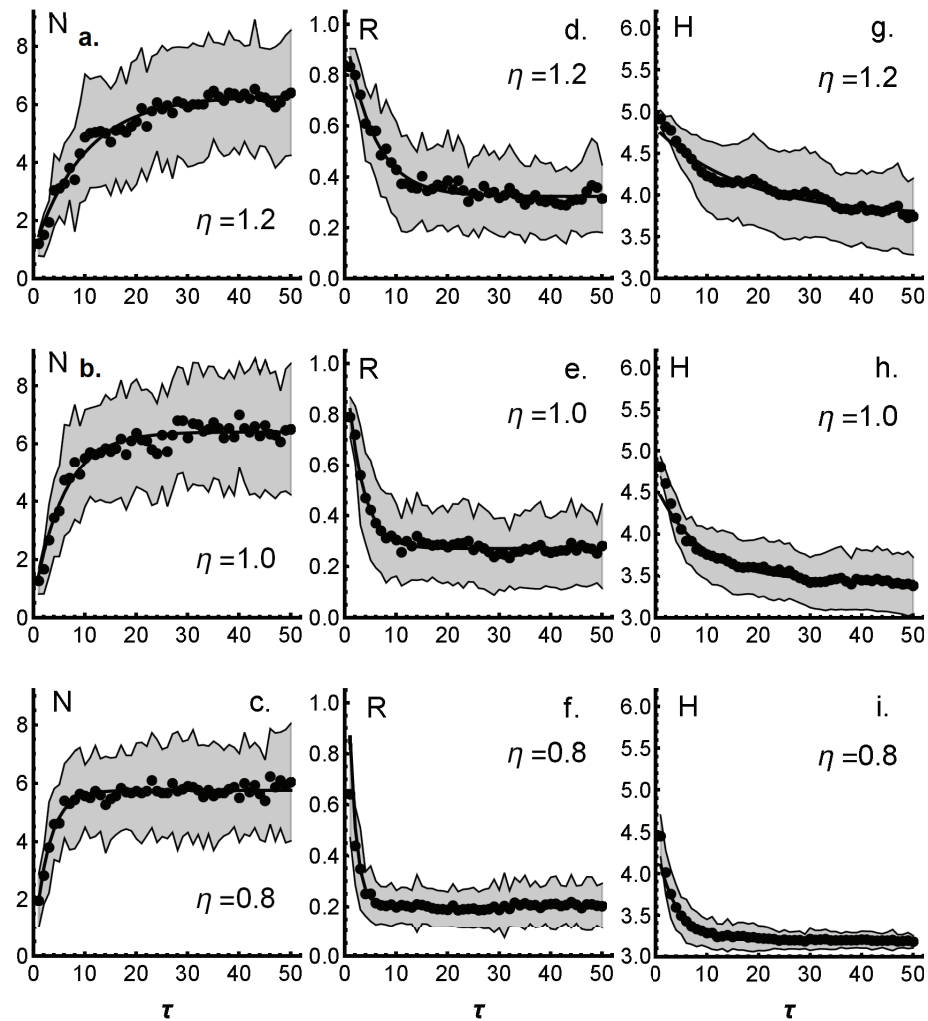


Figure 5. Average number of clusters N (a–c), relative occupancy R (d–f), and entropy H (g–i), for different values of parameter β (scaled with $N_0 = 100$). Results are for clusters in the final stabilized region of the formation of consensus clusters. A transition from multiple clusters to a single large cluster takes place around $\beta = 0.20$.

4. Discussion

In many respects, the ABM presented here describing the formation of consensus clusters resembles the so-called bounded confidence models (see e.g., [1–3,5]) but differs from those models in two important ways. First, the ABM model describes the space of states of the agents (i.e., opinion-like states) as complex networks (WoBs) of elements related to each other, mimicking conceptual or semantic networks, not as discrete sets of choices or fixed to a few choices as in most bounded confidence models. To extend the applicability of consensus formation models provides understanding of not only choices of existing opinions (e.g., as in political or consumer choices) but allow the evolution of opinions. It is important to find ways to use flexible, dynamic, and changing states of agents, and furthermore, allow these states to be affected by choices that the agents make during the course of the unfolding interaction events. Such features provide a more realistic basis to model the formation of consensus groups in comparison with traditional models of predetermined fixed choices between different opinion states. Second, the ABM model introduced here does not assume a fixed bounded confidence criterion for the realization of the interactions. Instead, decisions to interact are made stochastically on the basis of utility-type evaluations of the prospect of increasing similarity. Therefore, there is no sharp exclusion of interactions, but instead, bias for similarity (homophily). In this respect, the

WoBs introduced here parallel some other recent attempts to extend the consensus and opinion formation models [7,13].

Another set of models that resemble the ABM presented here are the so-called epistemic landscape models, which have found several potential applications in describing the formation and segregation of collaborative or consensus groups. The epistemic landscape models assume a fixed landscape of “knowledge” that agents explore [42,43] or closely related structures of fixed ground truths [10] or epistemic landscapes with agents sensing the distance from the ground truths or the gradients toward them. Such models assume a fixed, pre-existing landscapes of “knowledge” to be explored by agents, and thus the outcomes of the exploration are more or less predetermined by the structure of the landscape and its gradients [10,42,43]. The epistemic landscape models have turned out to be relevant to discussions of division of labor and how the ability of agents affects their collaboration, but such models do not easily yield to situations where dynamic changes in the landscape or the problem space are of interest.

The present ABM has two major limitations. First, the WoBs utilized in it are always substructures of more extensive templates. Although individual agents’ WoBs evolve, no new elements or connections are created, and WoBs will always remain as partial structures of the initial template. This, however, may not be a severe restriction in cases of intended applications where the targeted area of knowledge is opinions, views or belief of existing knowledge (i.e., discovery and creation of new knowledge is not in focus). Second, the WoBs and their update rules do not take into account the coherence of elements in the WoBs, nor requirements of coherence when elements are added or deleted. In some recent ABM approaches, the structure and coherence of agents’ opinions (or beliefs) are taken into account, allowing more realistic description of complex opinion and belief systems [7,44,45]. The inclusion of coherence of belief elements as part of ABM is, however, not unproblematic; diverse opinions exist regarding how to implement that notion as part of the idea of network-like knowledge. In some views, cognitive dissonance is important [46], and such aspects have been successfully implemented in ABM, where a model of cognitive (or conceptual) structures co-evolves with structural agents’ interactions [44]. In the present model, the coherence of belief systems arises from its dovetailed network structure, which is also a form of coherence of knowledge (see e.g., [47]). In that picture, adoption or deletion of nodes depends on their neighborhoods, where changes take place, always requiring the network to remain connected, but dissonant aspects of coherence are not taken into account. A lack of attention on dissonant connection is a clear restriction of the ABM introduced here.

Recent extensions of ABM have also included many other complex relevant features, for example: opinion dynamics making a separation between private and public opinions and the role of social hierarchies in interactions [48]; prior beliefs and knowledge-making decisions to adopt new beliefs [14,49]; and collective cognitive alignment, when group members perceive and recall the information they receive in aligned ways [50]. In addition to such features, it is possible to imagine many other socially important aspects the future ABM should take into account, for example, trust and its effects on social interactions [51]. While it is unreasonable to assume that at present any ABM can meet such diverse requirements, it is useful to keep in mind that omitting such features unavoidably limits the applicability of any suggested model, but in different ways.

In the current simple and idealized model presented here, many advanced and complex features are omitted. The goal of the model is to take a moderate further step to use more complex belief systems in similarity-biased models to show that even in the case of WoB-like structures, robust consensus clusters are formed; segregation and formation of consensus groups may well arise from the distribution of continuously evolving systems describing complex WoBs. Moreover, the dynamics of the formation of disciplinary consensus groups based on WoBs need not be overly complex; simple reinforcing similarity-seeking dynamics may clarify many empirically found features of how disciplinary groups and their boundaries are formed.

The most obvious areas of applications of the present model are found at least in three special cases. First, in disputes regarding how to frame and understand the meaning of abstract words or terms central to given scientific paradigms, where paradigms have their own specialized lexicons. Different ways of using scientific terms and framing problems are an interesting area of applications related to the social nature of science, the role of scientific discourse, and the argumentation and agreement of truth of scientific claims, as discussed extensively in the philosophy of science [19–22], as well as finding a tenable grounding on empirical research about scientific activity and formation of disciplinary groups [23,24,52,53]. Second, the ABM presented here may find applications in making sense of the formation of disciplinary groups in scholarly disputes about a given pre-existing corpus of study, for example, in disputes in the humanistic sciences about how to interpret the work of some renowned but difficult to follow scholar (e.g., Hegel, Kant or Wittgenstein), where the amount of interpretative literature with differing and even opposing views may be extensive and exceed the amount of original work. Third, and perhaps with the most foreseeable practical utility, to describe group formation in learning situations, where a group of students tries to make sense of a limited amount of sources about a given topic; in the simplest case, using a single textbook (see e.g., [28,29,35]). In such cases, due to sharing of views and opinions about same sources, one can nevertheless assume formation of different groups of consolidated views.

The possibility to connect the results of the current ABM more securely to existing empirical findings is not, however, straightforward. The ABM provides some insight structure and formation of discourse groups in learning and teaching, where groups of the size from three to about five to six students appear to perform the best [54–57], but in larger groups the phenomenon of isolation emerges [58,59]. While many unrelated factors (e.g., teaching arrangements and designs) affect group sizes, it is a plausible assumption that at least in the case of discourse groups, shared meanings of key terms and concepts is also a factor affecting group stability and how the number of students in a group evolves. For example, the isolation and break-up of small isolated groups from larger ones may well be a phenomenon related to similarity bias. In the case of scholarly disciplinary groups, empirical evidence of group sizes is more elusive and unclear. Scientometric analyses of the number of co-authors in publications provide some information, showing that small co-author groups are the most common, signaled as a fat-tailed distribution of number of authors [60,61]. This notion is supported also by results based on the size of scholarly groups that introduce (and thus use) new concepts, in which case small groups are also the most active [62,63]. Interestingly, reminiscent formation of disciplinary groups are found within focused research areas [64], as well as in case of non-scientific but organized beliefs groups (i.e., religion), [65]. In all these cases, however, the assumption that similarity bias guiding the formation of groups is a factor affecting the formation, segregation, and consolidation of groups is only tentative, although plausible. Without better knowledge of how, for example, that the changes in shared vocabulary or lexicons used by groups are correlated with group formation it is difficult to make more conclusive inferences on, or to propose more specific empirically testable hypotheses.

Nevertheless, due to the obvious tentativeness and generic nature of the results presented in this study, the present ABM model, where similarity is monitored through WoBs and where changes in WoBs are the basis of group formation dynamics, suggests that in future, more detailed studies, paying attention to correlations between consensus group formation and changes in WoBs or related structures may provide essentially new important insights on group dynamics. In that, ABM may also significantly guide designs of empirical research.

5. Conclusions

We presented an agent-based model (ABM) of the formation of consensus clusters when the agents possess a complex network of knowledge or belief elements, which form a web of beliefs (WoBs). The WoBs affect the ways they can communicate with each other,

and they are dynamically changed as an outcome of the interactions. The ABM takes into account dynamic changes in WoBs due to the sharing of their elements in events of interactions (i.e., communication). The dynamics are driven and constrained by similarity evaluations between agents (i.e., homophilic dynamics), evaluated on the basis of the similarity of WoBs.

The results of the model show that a group of agents, if their sensitivity to similarity is sufficiently high, will eventually form consensus groups, where agents have more or less similar WoBs. If the sensitivity to similarity is low, no segregation takes place, and agents remain in one or two large and diffuse groups. The main message of the ABM and its results is that even in the case of complex webs of beliefs, which are allowed to change dynamically in interactions between agents, a very simple dynamic is enough to produce segregation and consolidation of consensus groups in the presence of sensitivity to similarity (as described by webs of beliefs).

The simple and idealized agent-based model presented here is one step towards a better understanding of such complex situations and may also help to construct empirical settings of investigations to resolve the auxiliary features from the core features in driving consensus formation.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Open access funding provided by University of Helsinki.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABM	Agent-Based Modeling
WoB	Web of Belief
DBSCAN	Density-Based Scanning of Clusters

Appendix A. Generation of Template WoBs

Many real networks have nodes that have a relatively broad distribution of degrees (number of links or edges attached to them), so that in this sense, they can be characterized as complex. In extreme cases, such degree distributions are heavy-tailed distributions, heavy tails referring to a form of the trailing edge of a distribution that decays considerably more slowly than a normal distribution and where trailing edges resemble to some degree the inverse power-law-type distribution given as $P(d) \propto d^{-\lambda}$, with a power $\lambda' \in]1, 3]$. However, heavy-tailed distributions are seldom a genuinely inverse power-law [66]. For practical reasons, however, it is useful to consider the networks with heavy-tailed distributions using the model of inverse power-laws, because many essential characteristics of the networks are captured by that class of distributions [67]. At the limit $\lambda \rightarrow 3$, distributions will have well-defined first and second moments and are thus only moderately heavy-tailed. It is this type of distribution obtained at the limit $\lambda \rightarrow 3$ that we are interested here.

The networks used as templates for WoBs are generated using for affinities π_k a distribution

$$P(\pi_k) = P_0 [1 - (\lambda - 1) \Lambda \pi_k]^{-1/(\lambda-1)}, \quad (\text{A1})$$

where $\lambda \in]1, 3[$ and $\Lambda > 0$. The parameter λ determines the inverse power of the resulting degree distribution of the network, while Λ controls the cut-off of affinities, and $\pi_k < \Lambda(\lambda - 1)$. The affinity distribution can be derived in several ways and by using several parameterizations [35,68,69].

In simulations, we always use the same distributions of the affinities, but the template networks obtained from the affinity distribution are stochastically generated by using the IGraph package [70], which provides functionality for generating efficiently affinity-based networks simply by providing the probabilities π_k for the routine IGStaticFitnessGame. The output of the routine is a network with a predetermined number of edges, linked according to the probabilities π_k drawn from distribution $P(\pi_k)$ in Equation (A1).

The resulting WoBs of interest, which contain 100–200 vertexes, are networks having vertexes with an average degree of about three and containing only a few nodes with degrees up to 10 or slightly more. Most of the applications we are interested in typically have such distribution connections between their elements, which might be concepts, words, or terms (see the main text).

References

1. Galam, S. *Sociophysics*; Springer: New York, NY, USA, 2012.
2. Zafeiris, A.; Vicsek, T. *Why We Live in Hierarchies: A Quantitative Treatise*; Springer: Cham, Switzerland, 2018.
3. Flache, A.; Mäs, M.; Feliciani, T.; Chattoe-Brown, E.; Deffuant, G.; Huet, S.; Lorenz, J. Models of Social Influence: Towards the Next Frontiers. *JASSS-J. Artif. Soc. Soc. Simul.* **2017**, *20*, 2. [[CrossRef](#)]
4. Lorenz, J. Continuous opinion dynamics under bounded confidence: A survey. *Int. J. Mod. Phys. C* **2007**, *18*, 1819–1838. [[CrossRef](#)]
5. Castellano, C.; Fortunato, S.; Loreto, V. Statistical physics of social dynamics. *Rev. Mod. Phys.* **2009**, *81*, 591–646. [[CrossRef](#)]
6. Sîrbu, A.; Loreto, V.; Servedio, V.D.P.; Tria, F. Cohesion, consensus and extreme information in opinion dynamics. *Adv. Complex Syst.* **2013**, *16*, 1350035. [[CrossRef](#)]
7. Zafeiris, A. Opinion Polarization in Human Communities Can Emerge as a Natural Consequence of Beliefs Being Interrelated. *Entropy* **2022**, *24*, 1320. [[CrossRef](#)]
8. Carletti, T.; Fanelli, D.; Righi, S. Emerging structures in social networks guided by opinions' exchanges. *Adv. Complex Syst.* **2011**, *14*, 13–30. [[CrossRef](#)]
9. Zafeiris, A.; Koman, Z.; Mones, E.; Vicsek, T. Phenomenological theory of collective-decision making. *Physica A* **2017**, *470*, 287–298. [[CrossRef](#)]
10. Baliotti, S.; Mäs, M.; Helbing, D. On disciplinary fragmentation and scientific progress. *PLoS ONE* **2015**, *10*, e0118747. [[CrossRef](#)]
11. Hegselmann, R.; Krause, U. Opinion dynamics and bounded confidence models, analysis, and simulation. *JASSS-J. Artif. Soc. Soc. Simul.* **2002**, *5*, 2.
12. Deffuant, G.; Neau, D.; Amblard, F.; Weisbuch, G. Mixing beliefs among interacting agents. *Adv. Complex Syst.* **2000**, *3*, 87–98. [[CrossRef](#)]
13. Liu, Q.; Wang, X. Opinion dynamics with similarity-based random neighbors. *Sci. Rep.* **2013**, *3*, 2968. [[CrossRef](#)] [[PubMed](#)]
14. Liu, Q.; Wang, X. Social learning with bounded confidence and heterogeneous agents. *Phys. A* **2013**, *392*, 2368–2374. [[CrossRef](#)]
15. Wang, C. Opinion Dynamics with Higher-Order Bounded Confidence. *Entropy* **2022**, *24*, 1300. [[CrossRef](#)] [[PubMed](#)]
16. Li, J.; Xiao, R. Agent-based modelling approach for multidimensional opinion polarization in collective behaviour. *JASSS-J. Artif. Soc. Soc. Simul.* **2017**, *20*, 4. [[CrossRef](#)]
17. Lorenz, J. Fostering consensus in multidimensional continuous opinion dynamics under bounded confidence. In *Managing Complexity: Insights, Concepts, Applications*; Helbing, D., Ed.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 321–334.
18. Schawe, H.; Fontaine, S.; Hernández, L. When network bridges foster consensus. Bounded confidence models in networked societies. *Phys. Rev. Res.* **2021**, *3*, 023208. [[CrossRef](#)]
19. DeSolla Price, D.J. The Science of Science. *Bull. Atom. Sci.* **1965**, *21*, 2–8. [[CrossRef](#)]
20. Garfield, E. Citation indexes for science: A new dimension in documentation through association of ideas. *Science* **1955**, *122*, 108–111. [[CrossRef](#)]
21. Kuhn, T.S. *The Structure of Scientific Revolutions*; University of Chicago Press: Chicago, IL, USA, 1962.
22. Kuhn, T.S. *The Road since Structure*; University of Chicago Press: Chicago, IL, USA, 2000.
23. Fanelli, D. "Positive" results increase down the hierarchy of the sciences. *PLoS ONE* **2010**, *5*, e10068. [[CrossRef](#)]
24. Fanelli, D.; Glänzel, W. Bibliometric evidence for a hierarchy of the sciences. *PLoS ONE* **2013**, *8*, e66938. [[CrossRef](#)]
25. Wray, K.B. History of Epistemic Communities and Collaborative Research. In *International Encyclopedia of the Social and Behavioral Sciences*; Wright, J.D., Ed.; Elsevier: Amsterdam, The Netherlands, 2015; pp. 867–872.
26. Koponen, I.T. Nature of Science (NOS) being acquainted with Science of Science (SoS): Providing a panoramic picture of sciences to embody NOS for pre-service teachers. *Educ. Sci.* **2021**, *11*, 107. [[CrossRef](#)]
27. Koponen, I.; Södervik, I. Lexicons of Key Terms in Scholarly Texts and Their Disciplinary Differences: From Quantum Semantics Construction to Relative-Entropy-Based Comparisons. *Entropy* **2022**, *24*, 1058. [[CrossRef](#)] [[PubMed](#)]
28. Koponen, I.T.; Nousiainen, M. University Students' Associative Knowledge of History of Science: Matthew Effect in Action? *Eur. J. Sci. Math. Educ.* **2018**, *6*, 69–81. [[CrossRef](#)]
29. Lommi, H.; Koponen, I.T. Network cartography of university students' knowledge landscapes about the history of science: Landmarks and thematic communities. *Appl. Netw. Sci.* **2019**, *4*, 6. [[CrossRef](#)]

30. Quine, W.V.; Ullian, J.S. *The Web of Belief*; McGraw-Hill: New York, NY, USA, 2012.
31. Lipowski, A.; Lipowska, D.; Ferreira, A.L. Emergence of social structures via preferential selection. *Phys. Rev. E* **2014**, *90*, 032817. [[CrossRef](#)] [[PubMed](#)]
32. Lipowski, A.; Lipowska, D. Roulette-wheel selection via stochastic acceptance. *Phys. A* **2012**, *391*, 2193–2196. [[CrossRef](#)]
33. Thompson, G.W.; Kello, C.T. Walking across Wikipedia: A scale-free network model of semantic memory retrieval. *Front. Psych.* **2014**, *5*, 86. [[CrossRef](#)]
34. Masucci, A.P.; Kalampokis, A.; Equiluz, V.M.; Hernández-García, E. Wikipedia Information Flow Analysis Reveals the Scale-Free Architecture of the Semantic Space. *PLoS ONE* **2011**, *6*, e17333. [[CrossRef](#)] [[PubMed](#)]
35. Koponen, I.T. Modelling Students' Thematically Associated Knowledge: Networked Knowledge from Affinity Statistics. In *Complex Networks X. CompleNet 2019. Springer Proceedings in Complexity*; Cornelius, S.P., Martorell, C.G., Gomez-Gardanes, J., Goncalves, B., Eds.; Springer: Cham, Switzerland, 2019; pp.123–134.
36. Laciana, C.A.; Oteiza-Aguirre, N. An agent based multi-optional model for the diffusion of innovations. *Phys. A* **2014**, *394*, 254–265. [[CrossRef](#)]
37. Yukalov, V.I.; Sornette, D. Self-organization in complex systems as decision making. *Adv. Complex Syst.* **2014**, *17*, 1450016. [[CrossRef](#)]
38. Yukalov, V.I.; Sornette, D. Quantum Probabilities as Behavioral Probabilities. *Entropy* **2017**, *19*, 112. [[CrossRef](#)]
39. Weisberg, M. *Simulation and Similarity: Using Models to Understand the World*; Oxford University Press: Oxford, UK, 2013.
40. *Mathematica*, Version 13.1; Wolfram Research, Inc.: Champaign, IL, USA, 2013.
41. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*; AAAI Press: Palo Alto, CA, USA, 1996; pp. 226–231.
42. De Langhe, R. A comparison of two models of scientific progress. *Stud. Hist. Phil. Sci.* **2014**, *46*, 94–99. [[CrossRef](#)]
43. Weisberg, M.; Muldoon, R. Epistemic Landscapes and the Division of Cognitive Labor. *Phil. Sci.* **2009**, *76*, 225–252. [[CrossRef](#)]
44. Edmonds, B. Co-developing beliefs and social influence networks—Towards understanding socio-cognitive processes like Brexit. *Qual. Quant.* **2020**, *54*, 491–515. [[CrossRef](#)]
45. Parravano, A.; Andina-Díaz, A.; Meléndez-Jiménez, M.A. Bounded Confidence under Preferential Flip: A Coupled Dynamics of Structural Balance and Opinions. *PLoS ONE* **2016**, *11*, e0164323. [[CrossRef](#)]
46. Thagard, P. *Coherence in Thought and Action*; MIT Press: Cambridge, MA, USA, 2002.
47. Rescher, N. *The Coherence Theory of Truth*; Clarendon Press: Oxford, UK, 1973.
48. León-Medina, F.J.; Tena-Sánchez, J.; Miguel, F.J. Fakers becoming believers: How opinion dynamics are shaped by preference falsification, impression management and coherence heuristics. *Qual. Quant.* **2020**, *54*, 385–412. [[CrossRef](#)]
49. Shaw, L.A. Something out of nothing: A Bayesian learning computational model for the social construction of value. *J. Math. Soc.* **2020**, *44*, 65–89. [[CrossRef](#)]
50. Falandays, J.B.; Smaldino, P.E. The emergence of cultural attractors: How dynamic populations of learners achieve collective cognitive alignment. *Cog. Sci.* **2022**, *46*, e13183. [[CrossRef](#)] [[PubMed](#)]
51. Barrio, R.A.; Govezensky, T.; Ruiz-Gutiérrez, E.; Kaski, K.K. Modelling trading networks and the role of trust. *Phys. A* **2017**, *471*, 68–79. [[CrossRef](#)]
52. Klavans, R.; Boyack, K.W. Toward a consensus map of science. *J. Am. Soc. Inform. Sci. Tech.* **2009**, *60*, 455–476. [[CrossRef](#)]
53. Leydesdorff, L.; Carley, S.; Rafols, I. Global maps of science based on the new Web-of-Science categories. *Scientometrics* **2013**, *94*, 589–593. [[CrossRef](#)]
54. Barron, B. When smart groups fail. *J. Learn. Sci.* **2003**, *12*, 307–359. [[CrossRef](#)]
55. Heo, H.; Lim, K.Y.; Kim, Y. Exploratory study on the patterns of online interaction and knowledge co-construction in project-based learning. *Comp. Educ* **2010**, *55*, 1383–1392. [[CrossRef](#)]
56. Hogan, K.; Nastasi, B.K.; Pressley, M. Discourse patterns and collaborative scientific reasoning in peer and teacher-guided discussions. *Cog. Instr.* **1999**, *17*, 379–432. [[CrossRef](#)]
57. Tschan, F. Ideal cycles of communication (or cognitions) in triads, dyads, and individuals. *Small Group Res.* **2002**, *33*, 615–643. [[CrossRef](#)]
58. Bonito, J.A. The effect of contributing substantively on perceptions of participation. *Small Group Res.* **2000**, *31*, 528–553. [[CrossRef](#)]
59. Bonito, J.A. The analysis of participation in small groups: Methodological and conceptual issues related to interdependence. *Small Group Res.* **2002**, *33*, 412–438. [[CrossRef](#)]
60. Newman, M.E. Coauthorship networks and patterns of scientific collaboration. *Proc. Natl. Acad. Sci. USA* **2004**, *101*, 5200–5205. [[CrossRef](#)]
61. Palla, G.; Barabási, A.-L.; Vicsek, T. Quantifying social group evolution. *Nature* **2007**, *446*, 664–667. [[CrossRef](#)] [[PubMed](#)]
62. Milojevic, S. Quantifying the cognitive extent of science. *J. Informetr.* **2015**, *9*, 962–973. [[CrossRef](#)]
63. Milojevic, S. Principles of scientific research team formation and evolution. *Proc. Natl. Acad. Sci. USA* **2014**, *111*, 3984–3989. [[CrossRef](#)] [[PubMed](#)]
64. Diallo, S.Y.; Gore, R.J.; Padilla, J.J.; Lynch, C.J. An overview of modeling and simulation using content analysis. *Scientometrics* **2015**, *103*, 977–1002. [[CrossRef](#)]
65. Gore, R.; Lemos, C.; Shults, F.L.; Wildman, W.J. Forecasting changes in religiosity and existential security with an agent-based model. *JASSS-J. Artif. Soc. Soc. Simul.* **2018**, *21*, 1 [[CrossRef](#)]

66. Broido, A.D.; Clauset, A. Scale-free networks are rare. *Nat. Commun.* **2019**, *10*, 1017. [[CrossRef](#)]
67. Holme, P. Rare and everywhere: Perspectives on scale-free networks. *Nat. Commun.* **2019**, *10*, 1016. [[CrossRef](#)]
68. Servedio, V.P.D.; Caldarelli, G.; Buttà, P. Vertex intrinsic fitness: How to produce arbitrary scale-free networks. *Phys. Rev. E* **2004**, *70*, 056126. [[CrossRef](#)] [[PubMed](#)]
69. Caldarelli, G.; Capocci, A.; De Los Rios, P.; Muñoz, M.A. Scale-Free Networks from Varying Vertex Intrinsic Fitness. *Phys. Rev. Lett.* **2002**, *89*, 258702. [[CrossRef](#)] [[PubMed](#)]
70. Csárdi, G.; Nepusz, T. The Igraph software package for complex network research. *InterJ. Complex Syst.* **2006**, *1695*, 1–9.

Article

Research on the Evolutionary Path of Eco-Conservation and High-Quality Development in the Yellow River Basin Based on an Agent-Based Model

Aiwu Zhao ¹, Jingyi Wang ¹, Zhenzhen Sun ¹ and Hongjun Guan ^{1,2,*}

¹ School of Management Science and Engineering, Shandong University of Finance and Economics, Jinan 250014, China; aiwuzh@sdufe.edu.cn (A.Z.); jingyi_wang@mail.sdufe.edu.cn (J.W.); 191106001@mail.sdufe.edu.cn (Z.S.)

² Institute of Marine Economy and Management, Shandong University of Finance and Economics, Jinan 250014, China

* Correspondence: jjxyghj@sdufe.edu.cn

Abstract: The high-quality economic and social development of the Yellow River Basin is a combined system comprising the coordinated development of “economy–resources–environment–society”, with resources and the ecological environment bearing capacity as the constraints, and green innovative development as the driving force. Based on the systematic analysis of the structural dimensions of the composite system, this paper uses the balanced indicators and their coordinated development effectiveness to describe the development quality of the macro-composite system. In order to reveal the mechanism of the evolutionary path of the macro system, the resource- and environment-bearing capacity, regional high-quality development potential, regional innovation capacity, and high-quality development guarantee capacity are adopted as the main attributes and decision-making basis of the autonomous agents. The simulation results show that, under the existing development model, the economic development of all of the provinces in the Yellow River Basin will be constrained by resources and the environment. However, different policy scenarios significantly affect the evolutionary trends of economic development, resource consumption, and the environmental pollution situation. The mechanisms to overcome the bottleneck of the resource and ecological constraints are different for these policies, and the effects of the same policy in different provinces are also not the same.

Keywords: eco-conservation; high-quality development; agent-based model (ABM); composite system; balanced indicators

Citation: Zhao, A.; Wang, J.; Sun, Z.; Guan, H. Research on the Evolutionary Path of Eco-Conservation and High-Quality Development in the Yellow River Basin Based on an Agent-Based Model. *Systems* **2022**, *10*, 105. <https://doi.org/10.3390/systems10040105>

Academic Editors: Philippe Mathieu, Fernando De la Prieta, Alfonso González-Briones and Juan M. Corchado

Received: 16 June 2022

Accepted: 23 July 2022

Published: 27 July 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the increasing concern of humans regarding the issue of sustainable development, an increasing number of studies are exploring the path of sustainable economic development from the perspective of a complex eco-economic system (Sun et al., 2018) [1]. As economic development leads to a large concentration of the regional population, materials, and energy, and a high level of the consumption of resources, the ecological relationship becomes imbalanced; this reduces the ecological function of the natural system. Meanwhile, strict ecological constraints and the maintenance of ecological functions necessarily require constraints on economic growth, thus weakening the economic function of the system, which indicates a conflict between the two. At the same time, the improvement of the ecological function can improve the livelihoods and physical and mental health of watershed residents, which is conducive to the attraction of capital, talent, and other economic development factors, and has an important role in promoting the full exploitation of the economic function. Therefore, the two are unified, and there is a complex non-linear

relationship between economic development and ecological and environmental protection that is antagonistic.

The Yellow River Basin is an important core area in China for food production, energy-rich areas, and raw chemical materials; it is an important industrial base, and serves multiple ecological functions as an important ecological resource protection area. In terms of economic and social development and ecological security, it occupies an important position. However, due to various factors, such as its history and natural conditions, the economic and social development of the Yellow River Basin is relatively lagging, and the ecological environment of the basin shows strong vulnerability. With increasing human economic activities, the shortage of water resources, water environment pollution, and the over-utilization of water resources in the Yellow River Basin are becoming more serious. The unbalanced development of the provinces and regions in the Yellow River Basin and the inadequate development of the nine provinces in the upper and middle reaches are becoming increasingly prominent. Based on multi-agent modeling technology, this paper constructs a computational, experimental model for high-quality economic and social development in each province and region of the Yellow River Basin, combines multi-dimensional equilibrium indicators of the composite system with the attributes and behaviors of the agents, contrasts and analyzes the evolutionary paths of eco-conservation and high-quality development in each province and city of the Yellow River Basin through evolutionary simulation analysis under multiple scenarios, and explores the systematic optimization schemes of policy strategies such as green innovation, regulatory constraints, ecological compensation, and upstream and downstream linkages.

The remainder of this paper is organized as follows: Section 2 reviews the relevant literature, Section 3 presents an analysis of the subsystem components from a complex system perspective and the basis for measuring the effectiveness of the coordinated development of complex systems, Section 4 describes the construction and rules of the agent-based model (ABM), Section 5 analyzes the simulation results in terms of different scenarios, and Section 6 presents the discussion and conclusions of this paper.

2. Literature Review

Since General Secretary Xi Jinping's speech at the symposium on the eco-conservation and high-quality development of the Yellow River Basin in 2019, there has been an increasing amount of academic research on this topic (Ma et al.; Shi; Ren and Zhang) [2–4]. It is believed that synergistic development is the optimal solution to achieve ecological and social sustainability in the Yellow River Basin (Wang and Li) [5], and the “ecological priority” policy should be used as a guide to promote eco-conservation and high-quality development in the Yellow River Basin (Geng et al.) [6].

Research on the coordination relationship between two or more subsystems from the perspective of a composite system of the economy, society, resources, ecology, and the environment has become the basic framework for sustainable development issues (Fang et al.) [7]. Because of the complex non-linear coupling relationships among the subsystems, the process of coordinated development in composite systems is also the process of system coupling evolution (Sun et al.) [8]. The theory of the coordinated development of complex systems has been widely applied to the human environment (Srinivasan et al.) [9], the economic resource environment (Ma et al.) [10], the economic and social environment (Bastianoni et al.) [11], social ecology (Estoque and Murayama) [12], the urban environment (Li et al.) [13], and the climate economic environment (Aldieri and Vinci) [14]. Conceptual analysis and relationship analysis in the framework of the coordinated development of complex systems enrich the theoretical connotations of sustainable economic and social development. On the methodological side, environmental Kuznets curves (Zhao et al.) [15], coupled coordination models (Xing et al.) [16], gray models (GM) (Shi et al.) [17], autoregressive moving averages (ARMA) (Han et al.) [18], and machine learning algorithms (Li et al.) [19], etc., have been used for the analysis, evaluation, and prediction of composite systems.

In multi-objective complex systems, subsystems cooperate synergistically to transition from a disordered non-equilibrium state to a dynamic equilibrium state with certain functional and self-organizing structural mechanisms, which is a basic requirement for the coordinated development of complex systems (Turner) [20]. Based on complex system theory, synergy theory, and the idea of ecological civilization and green development in the “new normal” period, Zhao and Zhang divided the three regional unit subsystems in the spatial dimension into five subsystems: economic growth, social development, environmental quality, ecological health, and governance regulation [21]. They constructed a coordinated ecological development system composed of multiple interacting subsystems based on the state indicators and sequential parameters of each subsystem. Deng et al. used the gray water footprint and bearing capacity coefficients to predict the coupled evolution of the water environment and socioeconomic system under different scenarios in the Yangtze River Economic Zone based on physical and statistical models, and they accordingly proposed policy recommendations for the coordinated and sustainable development of the regional ecological environment and socio-economy [22]. In order to further clarify the mechanisms by which to achieve the performance goals, Kaplan and Norton proposed and enriched the balanced scorecard theory, and viewed it as a comprehensive strategic management and implementation tool for the translation of strategic goals into action [23–25]. However, the balanced scorecard neither establishes a causal relationship between indicators nor takes into account the time delay in the causal relationship; it is a diagnostic control system rather than an interactive control system (Ahn) [26].

As a bottom-up modeling approach, the multi-agent modeling technique offers the possibility to reveal the non-linear relationship between the global state of a complex system and the interaction of local constituent elements. Compared with modeling approaches based on system effects (such as System Dynamics (SD)) or process-oriented modeling approaches (such as Discrete-Event Simulation (DES)), ABM has unique research paradigm advantages, from individual behavior to macro “emergence”. It can better reflect the evolution mechanism and process of the complex system of the Yellow River Basin under resource and environmental constraints. Specifically, SD simulates the evolution of system development through the causal relationship between system elements. It is difficult to reflect the impact mechanism of environmental changes on the micro-individuals constituting the system, nor can it reflect the individual heterogeneity and the “emergence” of individual behavior response and individual interaction changes at the system level. Although DES has a high efficiency, reflecting the response of specific environment change, it is not suitable for a composite system because of its poor scalability and low coupling between modules. On the other hand, ABM reflects the differences of resource endowments in different regions through individual attributes such as environmental carrying capacity and innovation ability. It also reflects the heterogeneity of policy responses through individual behavior rules, and reflects the relationship between the upstream and downstream of the Yellow River Basin through individual interaction rules. When micro-adjustments of local agents accumulate to a certain extent, it will in turn restrict and affect the macro system environment, causing the agent to be in a dynamically changing environment and generate new evolution and learning momentum [27–29]. This will eventually lead to the appearance of deeper complex structural characteristics in the system. Therefore, ABM is suitable to simulate the evolutionary process deeply, and to reveal the micro-mechanism of the development of the Yellow River Basin. Multi-agent modeling technology is widely used in the study of the complex system evolution path. For example, Zhang et al. integrates the macro-factors (investment volatility) and micro-factors (individual behaviours) into a single analytical model, and simulates the evolutionary path of residential photovoltaic industry from the perspective of consumer behaviors [30]. Macal and North describe the following three elements as the basis of an agent-based model [31]:

- (1) a set of agents, including their attributes and behaviors;
- (2) a set of agent relationships, i.e., an underlying topology of connections that determines which agents interact with each other;

- (3) the agents' environment, with which they can also interact.

In this paper, the Yellow River Basin's eco-conservation and high-quality development system is regarded as a composite system of economy–resources–environment–society. Potential, regional innovation ability, and high-quality development guarantee ability, etc., are the attributes and decision-making basis of the basin's constituent units. Based on multi-agent modeling technology, the economic development, resource consumption, and ecological environment development of the Yellow River Basin provinces under different policy scenarios are simulated, including their trends and evolution. Compared with existing studies, this study combines empirical data with simulation methods to reproduce the microscopic dynamics of the macro-level state changes of composite systems through virtual–real linkage, and visualizes the policy design by comparing and analyzing the intrinsic mechanisms and laws of the system's evolution under different scenarios.

3. Analysis of the Composite System

This study focuses on the measurement of the state of high-quality development in the Yellow River Basin by constructing a balanced framework and tracking the evolutionary characteristics of key indicators across regions within the basin under different policy scenarios. To this end, based on complex systems theory and from the perspective of the complex multi-factors affecting the level of industrial water resource utilization, this paper builds a conceptual model of the economy–resource–environment–society complex system, incorporating economic development and natural resource development and utilization, including water resources, into the research scope. Due to the materiality of human life and the diversity of activities, it is difficult to objectively distinguish between resources, the environment, the economy, and society, which have formed a coupled and complex relationship of interaction, interconnection, and mutual influence among themselves and their subsystems and elements. In this context, the indicators are screened and theoretically analyzed using the theory of the human–earth relationship, based on relevant domestic and international literature on the design of economic, social, natural resource, and environmental indicators.

The relationship between subsystems in the composite system of eco-conservation and high-quality development in the Yellow River Basin is formed by the development of three levels of systematic coupling: (1) the coupling within a single subsystem develops in a coordinated manner, (2) the coupling between the two subsystems develops in a coordinated manner, and (3) the coupling between the systems develops in a coordinated manner. The three levels of the system constitute a complex system with their characteristics, structure, and function through various types of influence mechanisms, such as mutual influence, interdependence, and interaction, and this complex system—along with its characteristics, structure, and function—can be expressed using the following equation:

$$MCS \in \{S_1, S_2, S_3, S_4, Rel, Rst, Ob\}, S_i \in \{E_i, C_i, F_i\}, i = 1, 2, 3, 4 \quad (1)$$

Here, S_I represents the i th subsystem, and $E_i, C_i,$ and F_i refer to the characteristics, structure, and function of the subsystems. Rel denotes the coupling relation of mutual influence, interdependence, and interaction in the system coupling, which is called the system coupling set. It includes not only the internal coupling relation of the four subsystems but also the coupling relations between subsystems. Rst is a set of constraints faced by the subsystems, and Ob refers to the goals to be achieved by each subsystem. The coupling structure of the resources–environment–economy–society system in the Yellow River Basin is shown in Figure 1.

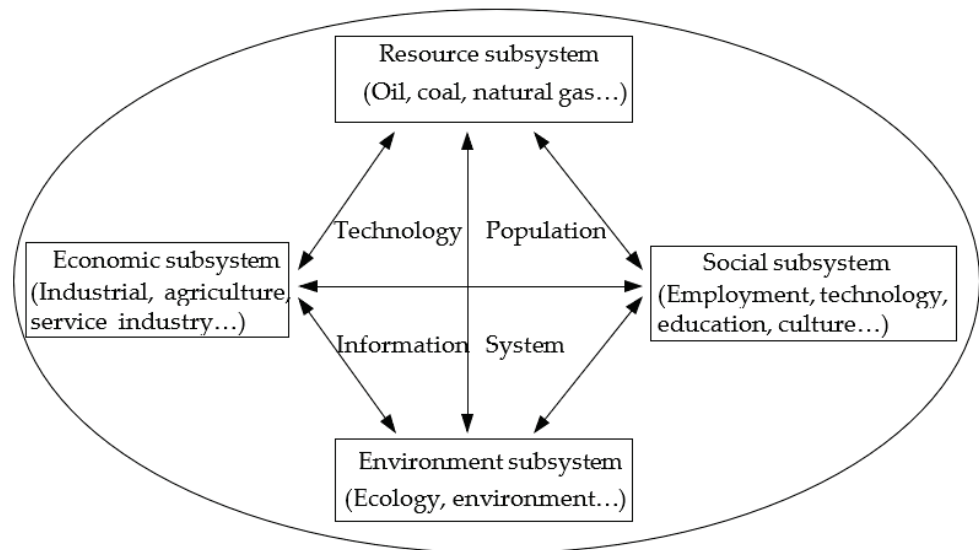


Figure 1. Coupled structure of the resource–environment–economy–society system in the Yellow River Basin.

The concepts of the four subsystems of the economy, society, resources, and the environment are separately extrapolated and defined, and the concepts of the economy (gross regional product, economic structure, etc.), society (employed population, education expenditure, science, and technology investment; the share of cultural and recreational expenditure in consumption expenditure; per capita disposable income, etc.), resources (water resources, land resources, forest coverage, ecological adaptability, etc.), and the environment (industrial waste gas in relative emissions of pollutants, wastewater emissions, energy consumption per unit of gross regional product, etc.) are clarified. The specific index system of each subsystem is selected to reflect both the basic characteristics and comprehensive effects of each subsystem, and to include the key variables that affect the change in the state of each decision unit. This paper evaluates the relative effectiveness of decision-making units (DMUs) with multiple inputs and outputs using Data Envelopment Analysis (DEA) measures. Because the C²R model is an ideal and effective method to study “production sectors” with multiple inputs, especially “production sectors” with multiple outputs that are “scale-efficient” and “technically efficient” at the same time, the C²R model cannot simply evaluate the technical validity between sectors; the C²GS² model compensates for the shortcomings of the C²R model, and is an ideal method to study the relative technical validity between production sectors. Therefore, in this paper, the C²R model is used to analyze the comprehensive effect of the decision unit, and the C²GS² model is used to analyze its specific technical effect. Taking the coupling relationship between subsystem A and subsystem B as an example, the coordinated development effectiveness function of both, based on the C²R model, is

$$\begin{aligned}
 & Z_e(A/B) = \min(\theta_e(A/B)) \\
 \text{s.t.} \quad & \begin{cases} \sum_{j=1}^n x_{Aj} \gamma_{A/Bj} + s^- = x_{A0} \theta_e(A/B) \\ \sum_{j=1}^n y_{Bj} \gamma_{A/Bj} - s^+ = y_{B0} \\ \forall \gamma_{A/Bj} \geq 0, j = 1, 2, \dots, n; s^+ \geq 0; s^- \geq 0 \end{cases} \quad (2)
 \end{aligned}$$

where $Z_e(A/B)$ denotes the coordination development effectiveness of subsystem A on subsystem B; the denominator is the input of subsystem A; the numerator is the output of subsystem B; n is the number of decision units; x and y are the input and output quantities of the subsystem, respectively; and s^- and s^+ are slack variables.

Similarly, the coordination validity function of subsystem A and subsystem B based on the C²GS² model is

$$\begin{aligned}
 & X_e(A/B) = \min(\sigma_e(A/B)) \\
 \text{s.t.} & \begin{cases} \sum_{j=1}^n x_{Aj} \gamma_{A/Bj} + s^- = x_{A0} \sigma_e(A/B) \\ \sum_{j=1}^n y_{Bj} \gamma_{A/Bj} - s^+ = y_{B0} \\ \sum_{j=1}^n \gamma_{A/Bj} = 1 \\ \forall \gamma_{A/Bj} \geq 0, j = 1, 2, \dots, n; s^+ \geq 0; s^- \geq 0 \end{cases} \quad (3)
 \end{aligned}$$

The developmental validity of subsystem A for subsystem B is calculated by the following equation:

$$F_e(A/B) = Z_e(A/B) / X_e(A/B) \quad (4)$$

As a result, the coordination validity, development validity, and coordinated development validity of the four subsystems of resources, ecology, the environment, and the economy, as well as society, are expressed as follows:

$$X_e(1, 2, \dots, k) = \frac{\sum_{i=1}^4 X_e(i/\bar{i}_{k-1}) \times X_{ek-1}(i/\bar{i}_{k-1})}{\sum_{i=1}^4 X_{ek-1}(\bar{i}_{k-1})} \quad (5)$$

$$Z_e(1, 2, \dots, k) = \frac{\sum_{i=1}^4 Z_e(i/\bar{i}_{k-1}) \times Z_{ek-1}(i/\bar{i}_{k-1})}{\sum_{i=1}^4 Z_{ek-1}(\bar{i}_{k-1})} \quad (6)$$

$$F_e(1, 2, \dots, k) = Z_e(1, 2, \dots, k) / X_e(1, 2, \dots, k) \quad (7)$$

Here, X_e , Z_e , and F_e refer, respectively, to the coordination validity, coordinated development validity, and development validity of the four subsystems; $k = 4$, \bar{i}_{k-1} refers to the set of different forms of any other $k - 1$ subsystems except a single subsystem i . The formula $Z_{ek-1}(\bar{i}_{k-1})$ refers to the coordinated development among $k - 1$ subsystems, and the formula $Z_{ek-1}(i/\bar{i}_{k-1})$ refers to the coordinated development validity of any other $k - 1$ subsystems.

4. Construction of the Agent-Based Model

In order to further study the dynamic process and evolutionary law of the synergistic evolution of the subsystems in the Yellow River Basin at different scales, and to reveal the influence mechanisms of policy scenarios such as innovation policy, environmental regulation, and ecological compensation on the high-quality economic and social development of the provinces and regions in the Yellow River Basin and the evolutionary law of the synergistic development of the composite system, this section summarizes the factors affecting the effectiveness of the coordinated development of the Yellow River Basin into four dimensions, namely the resource and environmental bearing capacity of the basin, the guaranteed capacity of high-quality development, the potential for the high-quality development of the region, and the innovation development capacity of the region. The interaction mechanism between the behavioral results of the agents and the coordinated development states of the system is shown in Figure 2.

In this work, we used the sample data of each province, region, and prefecture-level city in the Yellow River Basin from 2010 to 2018 as training data to construct an experimental model and compute a multi-agent framework for high-quality development in the Yellow River Basin. The data were mainly derived from the China Statistical Yearbook, China Environmental Statistical Yearbook, China Industrial Enterprise Database, China Industrial Enterprise Pollution Emission Database, China Ecological and Environmental Status Bulletin, China Water Resources Statistical Yearbook, China Water Resources Bulletin, and China City Statistical Yearbook, etc. The model is mainly composed of two parts: the agent and the spatial grid. The agent is mainly a virtual individual reflecting the economic and social characteristics and behaviors of each region. In addition to the spatial grid's need for the representation of its own assigned spatial environmental characteristics, it also stores a wide range of policy and statistical information which is needed for computation. In the process of a specific operation, the agent will make subjective decisions based on spatial attribute information provided by the grid, and the results of the agent's behavior

will be reflected in the changes in various indicators and affect the overall coordinated development status and environmental layout of the watershed.

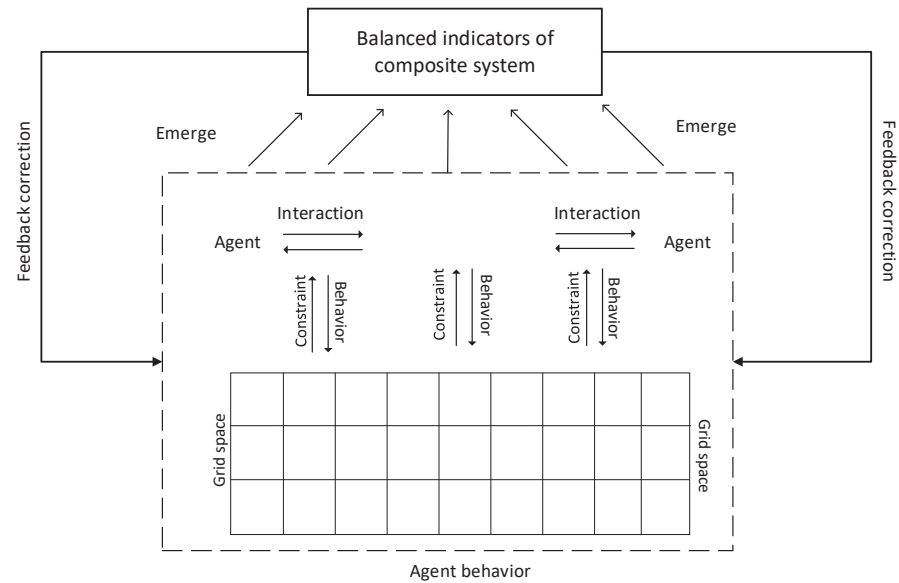


Figure 2. Interaction mechanism between balanced indicators and agents.

4.1. Agent Properties and Evolutionary Rules

Agent resource and environment bearing capacity includes natural resource variables (water resources, land resources, forest cover, ecological adaptability, etc.), the population bearing capacity, and environmental resource variables (including the relative emissions of pollutants in industrial waste gas, wastewater emissions, etc.). The resource and environmental bearing capacity is a key constraint for the high-quality development of the Yellow River Basin. Because the system model estimates the future environmental bearing capacity, the traditional method of calculating the regional resource and environmental bearing capacity is not applicable. Here, we use the resource and environmental capacity to measure the size of the regional resource environmental bearing capacity. The functional relationship is

$$B_t = f_b(Sou_t^i, Pep_t, Env_t^j) \tag{8}$$

where B_t is the resource bearing capacity of the region in year t , Sou_t^i is the stock of natural resources of category i in year t , Pep_t is the total population in year t , and Env_t^j is the stock of environmental resources of category j in year t . The values of the above variables are all relative values, with 2018 as the base period.

Agent development potential is the way in which the factor capacity of a region’s high-quality development is quantified, including the regional GDP, energy consumption per unit of output value, and pollutant emissions per unit of output value, etc. The function relationship is

$$G_t = f_g(Gdp_t^i, Eng_t^i, Pol_t^i) \tag{9}$$

Here, G_t is the comprehensive evaluation result of the region’s high-quality development potential in year t . Gdp_t^i , Eng_t^i , and Pol_t^i are the gross regional product, energy consumption per unit of output value, and pollutant emission per unit of output value of the industry category i in year t , respectively, and the values are taken as relative values with 2018 as the base period.

Agent innovation capability is a characterization of the level of science and technology development and innovation capacity of a region, including the level of science and

technology (scientific and technological talent, R&D institutions, number of patents, etc.), labor force, and investable R&D funds, etc. The functional relationship is

$$N_t = f_n(Tec_t, Lab_t, Fin_t) \tag{10}$$

where N_t is the innovation development capacity of the region in year t ; Tec_t , Lab_t , and Fin_t are the science and technology level, labor force, and investable R&D funds of the region in year t , respectively, and the values are taken as relative values, with 2018 as the base period. The investable R&D capital is related to the total regional GDP and R&D investment strength.

$$Fin_t = \sum Gdp_t^i \times \sigma_t \tag{11}$$

where σ_t is the share of R&D investment in GDP in year t ; due to the uncertainty of research development and innovation activities, the following conditions need to be satisfied in order for R&D investment to drive the progress of science and technology:

$$1 - e^{-\theta_w \times Fin_t} \geq u(0, 1) \tag{12}$$

θ_w is the speed control parameter of scientific and technological progress, and u is randomly distributed within $(0, 1)$, reflecting the uncertainty of innovation activities. If the above conditions are satisfied, this indicates that the innovation activity of R&D investment has achieved specific results and the level of science and technology has been improved:

$$Tec_t = Tec_{t-1} + \theta_{e1} \times u(0, 1) \times (Tec_{max} - Tec_{t-1}) \tag{13}$$

$$Gdp_t^i = Gdp_{t-1}^i + \theta_{e2} \times u(0, 1) \times (1 - \rho_t) \times (Gdp_{max} - Gdp_{t-1}) \tag{14}$$

$$Eng_t^i = Eng_{t-1}^i - \theta_{e3} \times u(0, 1) \times \rho_t \times (Eng_{t-1} - Eng_{min}) \tag{15}$$

$$Pol_t^i = Pol_{t-1}^i - \theta_{e4} \times u(0, 1) \times \rho_t \times (Pol_{t-1} - Pol_{min}) \tag{16}$$

θ_{e1} , θ_{e2} , θ_{e3} , and θ_{e4} are the control parameters of the change rate of the technology level, regional GDP, energy consumption per unit of output value, and pollutant emission per unit of output value, respectively. Tec_{max} and Gdp_{max} are the limit values of the maximum growth rate of the technology level and regional GDP, respectively (only the contribution of technological progress is considered for regional GDP in the forecast year). Eng_{min} and Pol_{min} are the limit values of the reduction rate of energy consumption per unit of output value and the pollutant emission per unit of output value. ρ_t denotes the importance of R&D activities for environmental performance, respectively, and is related to the industrial policy of the region.

Energy consumption per unit of output value and pollutant emissions per unit of output value change the environmental resource variables, and the functional relationship is expressed as

$$Env_t^j = f_e\left(\sum(Gdp_t^i \times Eng_t^i), \sum(Gdp_t^i \times Pol_t^i)\right) \tag{17}$$

The labor force variable is related to regional economic development and livability (a function of resource and environmental bearing capacity). It causes regional population changes, and the changes in the population variables and environmental resource variables change the regional resource and environmental bearing capacity. It is assumed that when the regional resource and environmental bearing capacity reach a threshold value that can be sustained, the regional environment deteriorates, the labor force is lost, and the rate of scientific and technological progress, θ_w , decreases.

Agent development security capacity is used to consider the degree of government, society, and public support for the region's high-quality development, including infrastructure construction, the service guarantee, information sharing, and environmental protection and governance, etc.; it has a functional relationship with parameter ρ_t , which improves environmental performance:

$$\rho_t = f_v(V_g, V_s, V_p, V_c) \tag{18}$$

$V_g, V_s, V_p,$ and V_c represent the degree of government, society, public participation, and support for the development and the degree of improvement in related policies.

Regarding the rules of agent evolution, based on the empirical data analysis, a probabilistic language set is used to express the empirical rules of the historical dataset. Specifically, the language set $S = \{s_0 : \text{low}, s_1 : \text{lower}, s_2 : \text{average}, s_3 : \text{higher}, s_4 : \text{high}\}$ is used to describe the data of various indicators affecting total factor productivity and regional policy information data, etc., for each year from 2010 to 2018, and is categorized into the regional resource and environmental bearing capacity, high-quality development guarantee capacity, regional high-quality development potential, and regional innovation development capacity in four dimensions. The comprehensive evaluation results of each dimension are expressed in a probabilistic language set, as follows:

$$X_{t,i} = \left\{ s_\alpha \left(p^{(\alpha)} \right) \mid s_\alpha \in S, 0 \leq p^{(\alpha)} \leq 1, \alpha = 0, 1, 2, \dots, \tau, \sum_{\alpha=0}^{\tau} p^{(\alpha)} = 1 \right\} \quad (19)$$

where $X_{t,i}$ represents the comprehensive evaluation results of each dimension index in year t , respectively, and $s_\alpha \left(p^{(\alpha)} \right)$ is the probability language variable, which is the probability $p^{(\alpha)}$ related to the language term s_α .

The evolution rules of each year in the historical data are expressed as $X_{t-1,i} \rightarrow X_{t,i}$; in other words, based on the comprehensive evaluation results of the historical data, the equilibrium relationship between the development of the four dimensions of the year (such as the expected development level based on the current situation of high-quality development supportability) is predicted. Based on the evaluation results of each factor in the above period, the possibility of approaching the previous evaluation results is found from the historical dataset. The distance measurement method of the rule reference was adopted from Yu et al. (2018):

Suppose that $h_s(p) = \left\{ s_\alpha \left(p^{(\alpha)} \right) \mid \alpha = 0, 1, \dots, \tau \right\}$ and $h'_s(p) = \left\{ s'_\beta \left(p^{(\beta)} \right) \mid \beta = 0, 1, \dots, \tau' \right\}$ are two probabilistic linguistic sets; then, the distance between them is defined as

$$d(h_s(p), h'_s(p)) = \left\{ \frac{1}{2} \left(\frac{1}{\tau} \sum_{(s_\alpha(p^{(\alpha)}))} \min_{(s_\alpha(p^{(\alpha)})) \in h_s(p)} \left(\left| f^*(s_\alpha) p^{(\alpha)} - f^*(s'_\beta) p^{(\beta)} \right| \right)^r + \frac{1}{\tau'} \sum_{(s'_\beta(p^{(\beta)}))} \min_{(s'_\beta(p^{(\beta)})) \in h'_s(p)} \left(\left| f^*(s'_\beta) p^{(\beta)} - f^*(s_\alpha) p^{(\alpha)} \right| \right)^r \right) \right\}^{\frac{1}{r}} \quad (20)$$

where f^* is a semantic scale function that can be defined as

$$f(s_\alpha) = \frac{\alpha}{\tau} (\alpha = 0, 1, \dots, \tau) \quad (21)$$

When $r = 1$, Formula (18) can be simplified as the Hamming–Hausdorff distance.

Regarding the quantitative methods for interaction of spatial lattices, according to Tobler’s (1970) first law of geography, similar areas in space have a higher interaction intensity. Distance is an important factor in the interaction of the ecological environment in the upper and lower reaches of the Yellow River Basin. The ecological environment in the upper reaches of the Yellow River Basin has distance decay characteristics. Referring to the distance decay estimation method in geography, the influence of distance on spatial interaction is represented by the Wilson maximum entropy model, as follows:

$$G_{ij} = A_i P_i B_j P_j f(d_{ij}) \quad (22)$$

Here, G_{ij} is the degree of ecological impact between region i and region j , P_i and P_j reflect the sizes of the two regions, A_i and B_j are the normalized factors of regional scale, and the distance decay function $f(d_{ij})$ represents the function with distance d as the independent variable to describe the influence of distance factors. This model adopts the following exponential distance decay function:

$$f(d) = e^{-\gamma d} (\gamma > 0) \quad (23)$$

where γ is the distance decay function factor.

4.2. Parameter Setting

In this study, the public parameters required for the system simulation and the attribute parameters of each region are set based on the training of sample data, and the values of each parameter are first standardized in the specific application. For the change speed parameters, such as the speed of technological progress, the level of science and technology, GDP, the energy consumption per unit of output value, and the pollutant emission per unit of output value, the system adopts the method of multiple simulation training and comparison with adjustment.

The system's main variables and their initial assignment rules are shown in Table 1.

Table 1. Main variables and initial assignment rules for computational experiments.

Variable/Parameter	Assignment Interval	Meaning	Assignment Rule
T	50	Simulation cycle	Fixed value
θ_w	0.01	Speed control parameter of scientific and technological progress	Training value
θ_{e1}	0.01	Speed control parameter of technical improvement	Training value
θ_{e2}	0.01	Control parameter of GDP growth rate	Training value
θ_{e3}	0.01	Speed control parameter for energy consumption reduction per unit output value	Training value
θ_{e4}	0.01	Control parameter of pollutant reduction rate per unit output value	Training value
γ	5	Distance decay function factor	Training value
σ_t	0.05	The proportion of R&D investment in GDP	Empirical value
Gdp_{max}	0.2	The maximum growth rate of GDP	Empirical value
Eng_{min}	0.2	Maximum reduction rate of energy consumption per unit output value	Empirical value
Pol_{min}	0.2	Maximum reduction rate of pollutant discharge per unit output value	Empirical value
Tec_{max}	0.2	Maximum speed of technical improvement	Empirical value

The specific attribute parameter settings affect the simulation results of the system, such that the correspondence with the empirical results is considered in the parameter settings as much as possible, and the universality and representativeness are considered. Because the detailed design and parameter setting affect the research results, research based on a multi-agent model should pay attention to the "virtual-reality linkage". That is, through the comparison of simulation results and real data, the rationality of the model should be tested. This method is good at comparing and analyzing the results of system evolution under different scenarios. As the change of parameters means the change of the environment, it is convenient to visually analyze the impact differences of different policies for the same object under the same rules. Because of the complexity and uncertainty of the evolutionary path of the actual system, the multi-agent model in this paper may not accurately predict the future. It is simplified to the above attributes and behavior rules. The results of the simulation experiments are only used for the comparison of different scenarios.

5. Simulation of High-Quality Development Evolution under Different Scenarios

In order to compare the evolution paths of eco-conservation and high-quality development in the Yellow River Basin under different scenarios, the following scenarios were designed for comparative experiments:

- O: The economic development model without policy instrument intervention. According to the profit-maximization principle, industries conduct their economic activities, and R&D investments are focused on improving production efficiency and reducing production costs.
- I: Economic policy development model related to green innovation. This promotes the high-quality development of green innovation through economic incentives.
- I_EN: The combined development model of green innovation with no different ecological environment constraints throughout the whole basin. On the one hand, it promotes green innovation through economic incentives and other means, and on the other hand, it adopts indiscriminate ecological and environmental protection constraints in all provinces in the Yellow River Basin.
- I_ED: A combined development model of green innovation and differentiated ecological and environmental constraints in the upper, middle, and lower reaches. On the one hand, green innovation is promoted through economic incentives and other means. On the other hand, differentiated ecological and environmental protection constraints are applied to the provinces in the upper, middle, and lower reaches of the Yellow River Basin, with the lower reaches being compensated according to the ecological and environmental level of the upper reaches.

The computational experiment platform of the proposed model was developed with Delphi Xe 11.1, and Oracle 11g was adopted as the database tool. Based on the powerful PASCAL language, Delphi has a good database interface and a friendly visual programming environment. Its convenient modular design is flexible for function expansion and policy scenario setting. The initial values of the system evolution simulation are based on 2018 data, and the evolution statistics of each region's economy, society, resources, and environment under different scenarios are obtained after calculation experiments. The year is taken as the simulation evolution cycle, and the evolution cycle is set to 50 years. In order to eliminate the influence of random factors on the evolution results, each scenario is simulated 100 times, and the average value of multiple simulations is taken as the final evolution result.

5.1. Analysis of the Evolution Path of the Economic Development Trend in the Yellow River Basin under Different Scenarios

Taking 2018 as the base period, the evolution paths of each province in the Yellow River Basin under different development modes are simulated individually. The evolution trends of economic development in each province under different scenarios in 50 cycles are shown in Figure 3.

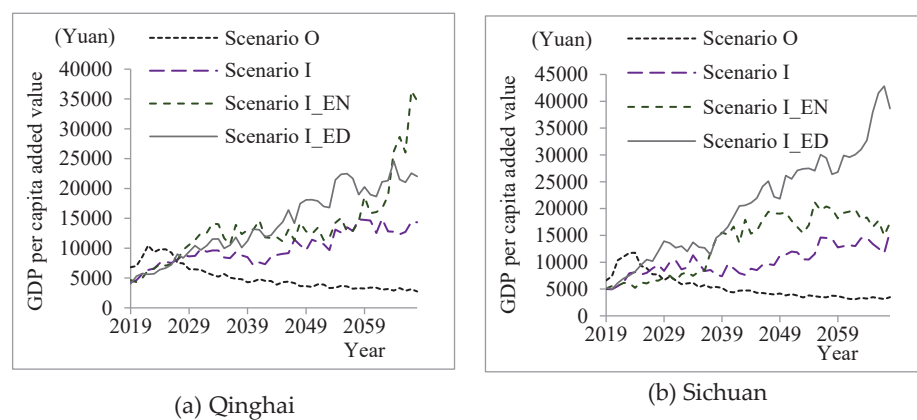


Figure 3. Cont.

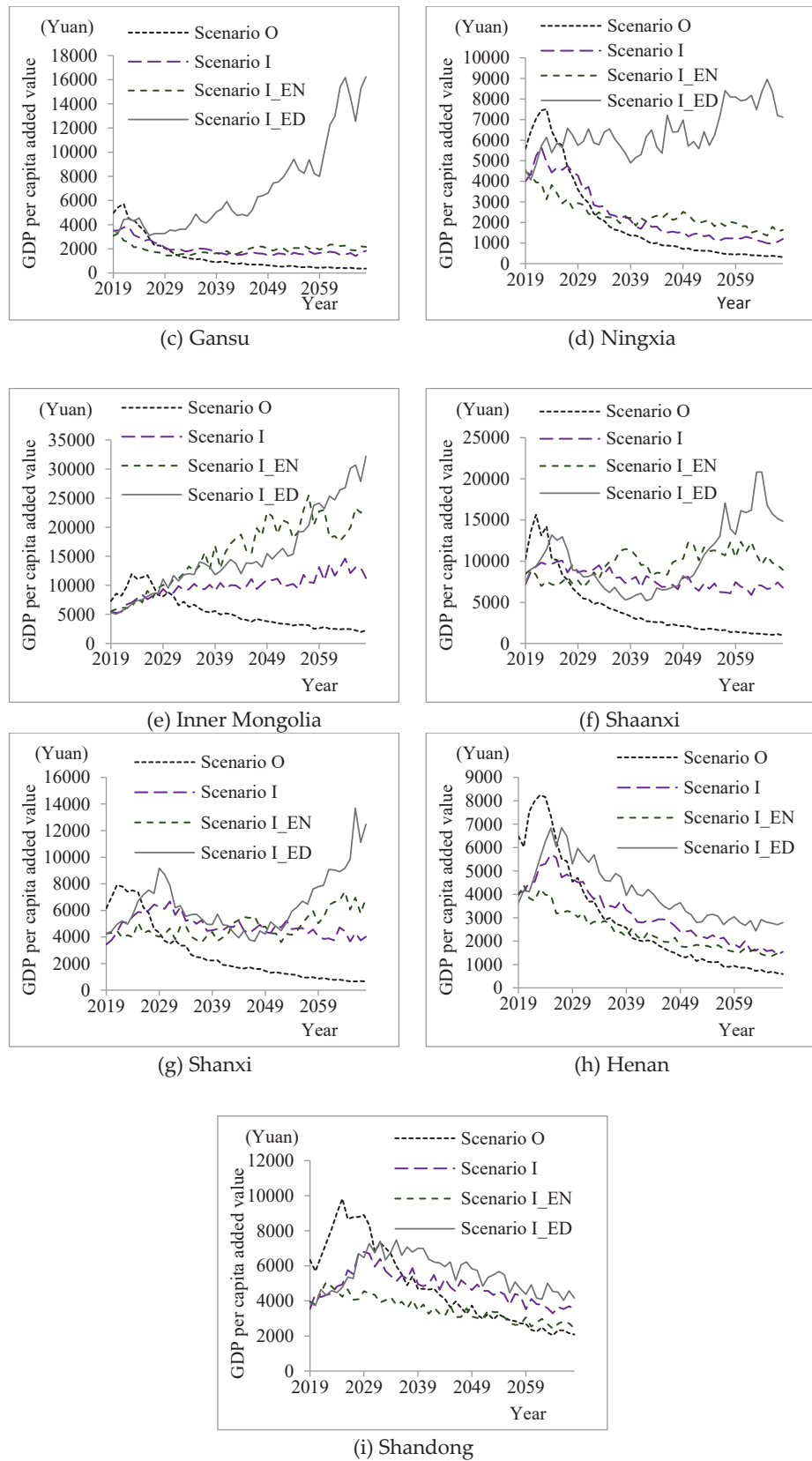


Figure 3. Evolutionary trends of the economic development in the provinces of the Yellow River Basin under different scenarios.

As seen in Figure 3, the economic development trend of each province in the Yellow River basin varies over 50 years under different scenarios. However, in general, scenario I_ED (the green innovation and differentiated ecological and environmental constraints model for the upper, middle, and lower reaches) has a more significant advantage for GDP per capita growth in the middle and late stages of the simulated evolution, except for Qinghai Province in the upper reaches.

In particular, scenario O (the crude development model without ecological constraints) prevails in the early stage of simulation evolution, but the overall economic growth under this development model shows an apparently inverted “U” shape, which is unsustainable in the long term.

Scenario I (the green innovation development model with economic incentives) has different evolutionary paths in different provinces, among which Qinghai, Sichuan, and Inner Mongolia show a slow upward trend; Shanxi and Shaanxi have no apparent fluctuation, and Ningxia, Gansu, Henan, and Shandong show an inverted “U”-type trend. Although the long-term trend is better than the extensive development model, it still shows a downward trend in the middle and late stages; it also highlights the importance of increasing the support for science and technology innovation in Qinghai, Sichuan, and Inner Mongolia to promote local economic development.

Scenario I_EN (the green innovation with basin-wide undifferentiated ecological and environmental constraint model) evolves similarly to green innovation scenario I in most provinces. However, in the Qinghai, Sichuan, Shaanxi, and Shanxi provinces, their GDP per capita growth is significantly better than in scenario I in the middle and late stages of the simulated evolution, reflecting the effect of ecological and environmental protection in the promotion of economic growth in the region.

Under scenario I_ED, although all of the provinces achieve higher GDP per capita growth than other scenarios in the late stage of simulation evolution, the evolutionary paths of all of the provinces are not consistent, among which Qinghai, Sichuan, and Inner Mongolia show an overall upward trend. However, Qinghai is the only province with better GDP per capita growth than I_ED under scenario I_EN. Gansu, Shaanxi, and Shanxi show a “U” shape, while Henan and Shandong show a moderately inverted “U” shape. It can be seen that in order to achieve high-quality development in the Yellow River Basin, economic policies should be formulated not only by distinguishing among the geographical characteristics of the upper, middle, and lower reaches but also by taking into account the resource endowment and ecological environment characteristics of different regions, and by formulating differentiated policy strategies.

5.2. Scenario-Based Comparative Analysis of the Development of the Yellow River Basin by Province

The simulation of the 50-year evolution of each province in the Yellow River Basin under different scenarios shows that the combination of the scenarios of innovation policies and eco-conservation policies has long-term effects on economic development, resource consumption, and the environment in each province, and the results of the scenarios vary greatly among provinces. The economic growth, resource consumption, ecological environment, and impact on the lower reaches’ ecological environment in each province under scenario O (the crude development model without ecological constraints) are shown in Table 2, and the “mean ranking” refers to the comparison of the annual mean values of the corresponding dimensions under scenario O, scenario I, scenario I_EN, and scenario I_ED. The results are shown in Table 2, where economic growth refers to the average annual increase in GDP per capita, which is a positive indicator, and “1” indicates the best; resource consumption, ecological environment, and the lower reaches’ impact are negative indicators, and “1” again indicates the best.

Table 2. Comparative ranking of the development of each dimension under scenario O in the Yellow River Basin provinces.

Province	Economic Growth		Resources Consumption		Ecology Environment		Impact Lower Reaches	
	Periodic Mean	Mean Sort	Periodic Mean	Mean Sort	Periodic Mean	Mean Sort	Periodic Mean	Mean Sort
Qinghai	11,650.25	4	16,460.93	4	140,490.11	4	7356.87	4
Sichuan	170,335.71	4	85,103.21	4	10,430,415	4	66,880.01	4
Gansu	21,328.36	3	19,184.24	4	2,128,716.6	4	49,712.45	4
Ningxia	8929.3	2	15,311.95	4	1,009,006.5	4	52,867.58	4
Inner-Mongolia	59,270.51	4	66,669.79	4	2,942,370.4	4	153,081.69	4
Shaanxi	80,319.17	4	41,108.57	4	4,578,867.4	4	157,932.48	4
Shanxi	51,381.14	4	60,608.01	4	4,287,611.2	4	224,095.52	4
Henan	141,309.92	2	66,852.93	4	5,981,967.5	4	297,824.63	4
Shandong	223,396.87	1	111,192.29	4	17,217,165	4	-	-

Note: The color block, from light to dark, indicates the sorting results from the best to the worst.

As can be seen from Table 2, except for the four data on economic growth, the provinces in the Yellow River Basin ranked first in the bottom in terms of resource consumption, ecological environment, and impact on the lower reaches under scenario O. This indicates that although the crude development model is beneficial to the economic growth of the region in individual provinces, at the expense of the ecological environment, this development model will also have a significant impact on the ecological environment of the lower reaches.

As shown in Table 3, green innovation has significant effects on the reduction of resource consumption, optimizing the ecological environment and reducing the impact of environmental pollution in the region on the lower reaches, especially in terms of the reduction of resource consumption. The Gansu, Inner Mongolia, Shaanxi, and Shanxi provinces reach the optimal resource consumption under this scenario; in terms of the ecological environment and impact on the lower reaches, this scenario is significantly better than scenario O of the crude development model. In terms of economic growth, the Shandong, Henan, and Ningxia provinces lag behind the crude development scenario O.

Table 3. Comparative ranking of the development of each dimension under Scenario I in the Yellow River Basin provinces.

Province	Economic Growth		Resources Consumption		Ecology Environment		Impact Lower Reaches	
	Periodic Mean	Mean Sort	Periodic Mean	Mean Sort	Periodic Mean	Mean Sort	Periodic Mean	Mean Sort
Qinghai	15,450.36	3	11,799.64	2	75,008.79	3	3926.87	3
Sichuan	219,439.4	3	54,970.65	2	6,135,319.7	3	36,179.37	3
Gansu	22,694.15	2	8846.17	1	1,013,780.4	3	22,271.67	3
Ningxia	8869.36	3	10,245.05	2	469,406.02	3	24,594.85	3
Inner-Mongolia	71,990.12	3	35,578.52	1	1,709,489.1	3	89,057.09	3
Shaanxi	102,047.13	3	17,895.28	1	2,537,886	3	84,071.45	3
Shanxi	62,569.09	2	28,929.15	1	2,030,434.5	3	106,122.33	3
Henan	136,472.67	3	36,077.92	2	4,254,877.6	3	211,837.88	3
Shandong	201,533.46	3	61,046.84	3	14,936,784	2	-	-

Note: The color block, from light to dark, indicates the sorting results from the best to the worst.

As seen in Table 4, the inclusion of strict ecological and environmental constraints does not always have a negative impact on the economy. In terms of the average annual growth value of the economy over 50 years, Inner Mongolia and Shaanxi achieve optimal economic growth under the scenario with the inclusion of strict environmental constraints; the resource consumption under this scenario is much better than that of the crude development scenario O. Compared with green innovation scenario I, scenario I_EN, with the dual combination of green innovation and ecological and environmental protection, is slightly better; moreover, this scenario is significantly better than both scenario O and scenario I in terms of the ecological environment and the impact on the lower reaches.

Table 4. Comparative ranking of the development of each dimension under Scenario I_EN in the Yellow River Basin provinces.

Province	Economic Growth		Resources Consumption		Ecology Environment		Impact Lower Reaches	
	Periodic Mean	Mean Sort	Periodic Mean	Mean Sort	Periodic Mean	Mean Sort	Periodic Mean	Mean Sort
Qinghai	18,451.96	2	9064.92	1	56,192.16	2	2937.88	2
Sichuan	258,358.59	2	53,122.8	1	4,010,785.9	2	21,964.45	2
Gansu	20,989.1	4	9271	2	593,202.26	2	13,340.78	2
Ningxia	8324.35	4	7876.64	1	243,600.44	2	12,763.61	2
Inner-Mongolia	94,928.05	1	44,085.16	2	787,153.95	1	40,579.8	1
Shaanxi	108,755.5	1	20,922.59	2	1,349,889.5	1	39,129.28	1
Shanxi	57,418.3	3	38,773.23	2	1,095,447.9	1	57,254.48	1
Henan	115,455.25	4	35,022.77	1	3,056,135.4	1	152,156.02	1
Shandong	172,713.63	4	46,903.43	1	12,232,450	1	-	-

Note: The color block, from light to dark, indicates the sorting results from the best to the worst.

As shown in Table 5, the implementation of the segmented control ecological and environmental protection strategy in the Yellow River Basin is much better than other scenarios in terms of economic growth, but it is significantly inferior to scenario I and scenario I_EN in terms of reducing resource consumption; in terms of the ecological environment and impact on the lower reaches, this scenario is significantly better than scenario O and scenario I, but not significantly different from scenario I_EN. It can be seen that the implementation of the segmented control of the Yellow River basin-wide ecological and environmental protection strategy can better guarantee long-term economic growth, but under the existing technical level and green innovation conditions, most of the provinces will be limited by the resource bearing capacity. Therefore, it is necessary to vigorously develop green industries and new industries while protecting the ecological environment throughout the region in order to achieve the comprehensive, high-quality development of industry, technology, ecology, the environment, and society by changing the existing industrial structure.

Table 5. Comparative ranking of the development of each dimension under scenario I_ED in the Yellow River Basin provinces.

Province	Economic Growth		Resources Consumption		Ecology Environment		Impact Lower Reaches	
	Periodic Mean	Mean Sort	Periodic Mean	Mean Sort	Periodic Mean	Mean Sort	Periodic Mean	Mean Sort
Qinghai	19,018.51	1	12,941.77	3	32,695.17	1	1712.49	1
Sichuan	342,864	1	61,524.9	3	3,088,360.3	1	21,516.52	1
Gansu	40,393.65	1	12,357.85	3	436,312.02	1	9024.54	1
Ningxia	13,645.62	1	11,461.19	3	176,289.76	1	9236.82	1
Inner-Mongolia	88,228.44	2	52,568.74	3	817,342.66	2	42,530.06	2
Shaanxi	108,517.61	2	29,579.6	3	1,430,490.5	2	49,998.64	2
Shanxi	69,682.36	1	40,420.05	3	1,412,879.9	2	73,845.33	2
Henan	156,672.64	1	38,999.83	3	4,162,523.9	2	207,239.86	2
Shandong	217,203.87	2	48,659.22	2	15,923,150	3	-	-

Note: The color block, from light to dark, indicates the sorting results from the best to the worst.

5.3. Comparative Analysis of the Overall Evolutionary Trends in the Yellow River Basin under Different Scenarios

The key to the global governance of the Yellow River Basin is to change the traditional situation of “governing the Yellow River in nine provinces and managing each section”. According to the analysis of the above provinces’ development status and evolution process, each province’s development stages and work priorities are different. The evolution trends of economic development in the upper, middle, and lower reaches, and the whole Yellow River Basin under different scenarios are shown in Figure 4.

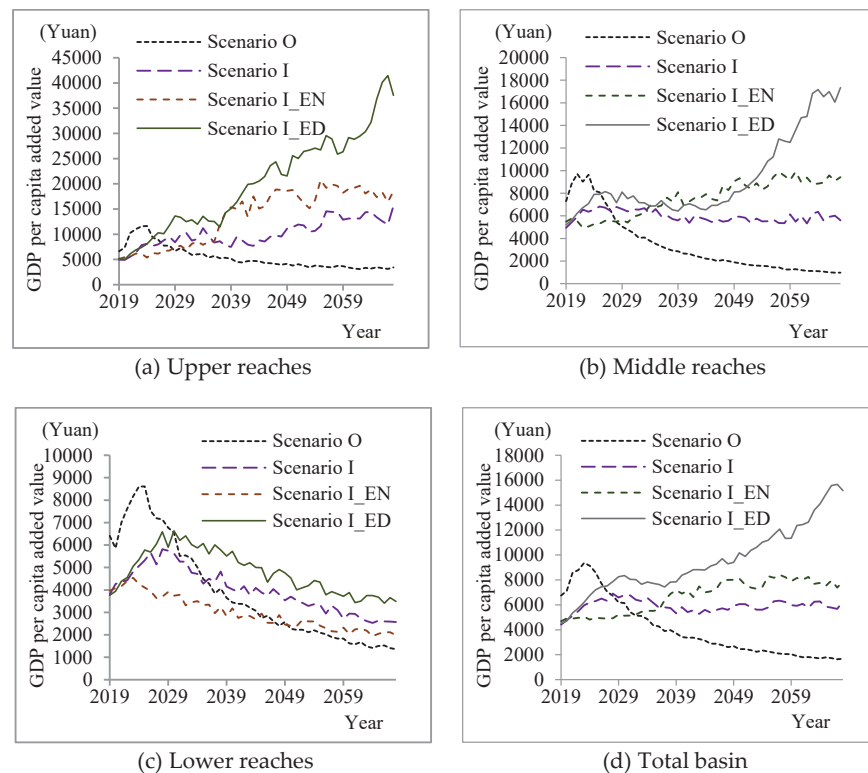


Figure 4. Simulation of the evolution of the overall economic development of the Yellow River Basin under different scenarios.

As seen in Figure 4, from the long-term evolutionary trend, scenario I_ED (green innovation with differentiated ecological and environmental constraint patterns in the upper, middle, and lower reaches) is optimal in terms of the overall economic development of the Yellow River Basin, with the exception of the lower reaches, for which scenario I_EN (green innovation with basin-wide undifferentiated ecological and environmental constraint model) is significantly better than scenario I (green innovation development model with economic incentives). In the context of green innovation, the strict ecological and environmental protection has a positive effect on the economic development of the Yellow River Basin, especially from a basin-wide perspective. The phased-control ecological and environmental protection strategy is far superior to other scenarios in terms of economic development. As the high-quality development of the middle and upper reaches of the Yellow River Basin is constrained by the business environment, human living environment, salary and benefits, and development space, it faces enormous competitive pressure regarding green innovation. Further analysis of the overall resource consumption in the Yellow River Basin under different scenarios is shown in Figure 5.

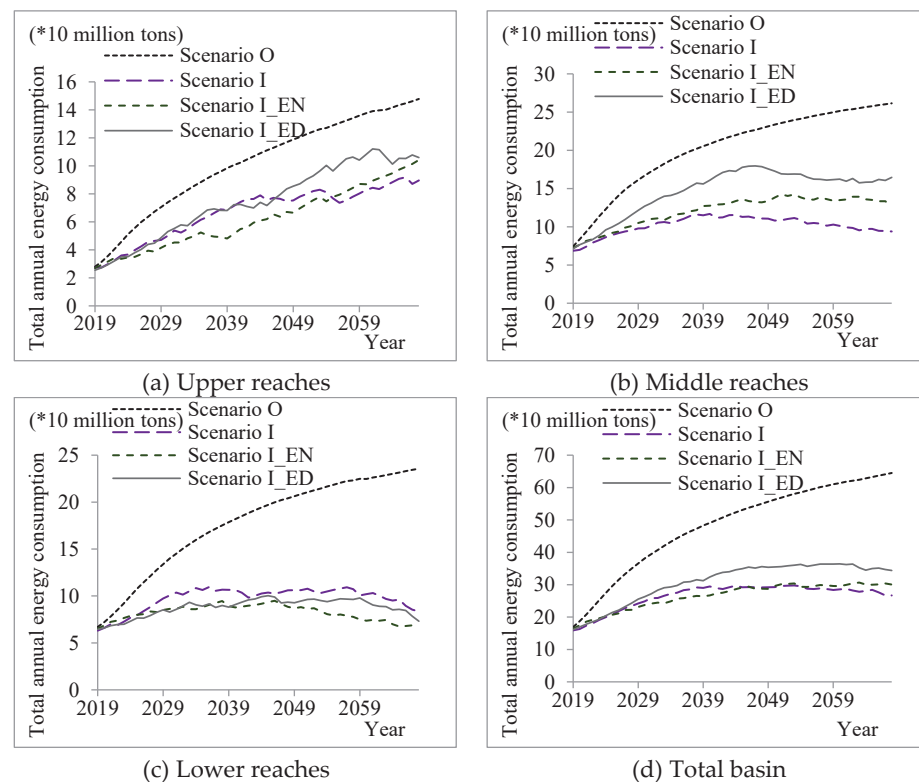


Figure 5. Simulation of the evolution of the overall resource consumption in the Yellow River Basin under different scenarios.

As shown in Figure 5, green innovation can better reduce resource consumption in the Yellow River Basin, and, overall, the effect of resource consumption reduction under the scenario without adding strict ecological and environmental constraints is generally better than that of scenario I_ED with the segmented control of region-wide ecological and environmental constraints; specifically, under scenario I_ED (the green innovation and differentiated ecological and environmental constraints model for upper, middle, and lower reaches), because the regions in the middle and upper reaches of the Yellow River Basin enforce stricter ecological and environmental constraints than the lower reaches, which reduces the regional green innovation capacity to a certain extent, in the long run, the middle and upper reaches can better reduce their resource consumption under scenario I (the green innovation development model with economic incentives) regarding green innovation. Meanwhile, for the lower reaches of the Yellow River Basin, under scenario

I_EN (the green innovation with basin-wide undifferentiated ecological and environmental constraints model), and according to Figure 5c, it can be seen that the unified strict ecological and environmental constraints in the upper and lower reaches constrain economic growth; therefore, the resource consumption under this scenario is optimal. Further analysis of the resource consumption per unit of GDP under different scenarios is shown in Figure 6.

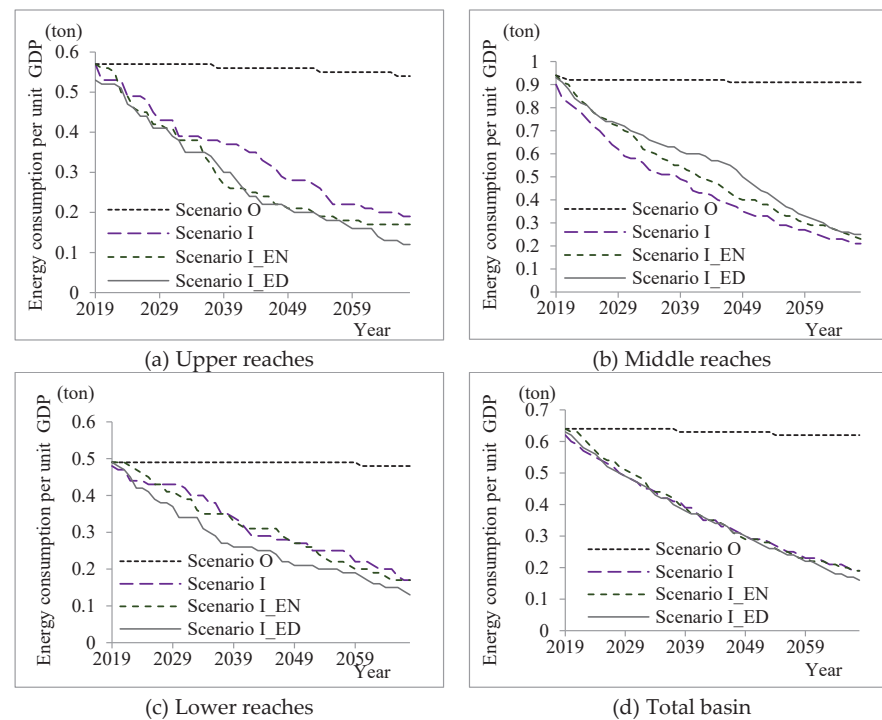


Figure 6. Simulation of the evolution of the resource consumption per unit of GDP in the Yellow River Basin under different scenarios.

As seen in Figure 6, overall, the three scenarios with the inclusion of green innovation have similar effects on the reduction of the resource consumption per unit of GDP in the Yellow River Basin. However, scenario I_ED (green innovation and differentiated ecological and environmental constraints model for upper, middle, and lower reaches) is optimal in the upper and lower reaches, while scenario I (the green innovation development model with economic incentives) is optimal in the middle reaches. Further analysis of the overall ecological environment in the Yellow River basin under different scenarios is shown in Figure 7.

As seen in Figure 7, overall, scenario I_EN (the green innovation with basin-wide undifferentiated ecological constraint model) and scenario I_ED (the green innovation and differentiated ecological and environmental constraints model for upper, middle, and lower reaches) are optimal in terms of ecological and environmental protection in the Yellow River basin; specifically, under scenario I_ED, the upper reaches adopt more stringent ecological environment constraints than the middle and lower reaches, such that the ecological environment of the upper reaches is the best under this scenario. Further analysis of the pollution emissions per unit of GDP under different scenarios is shown in Figure 8.

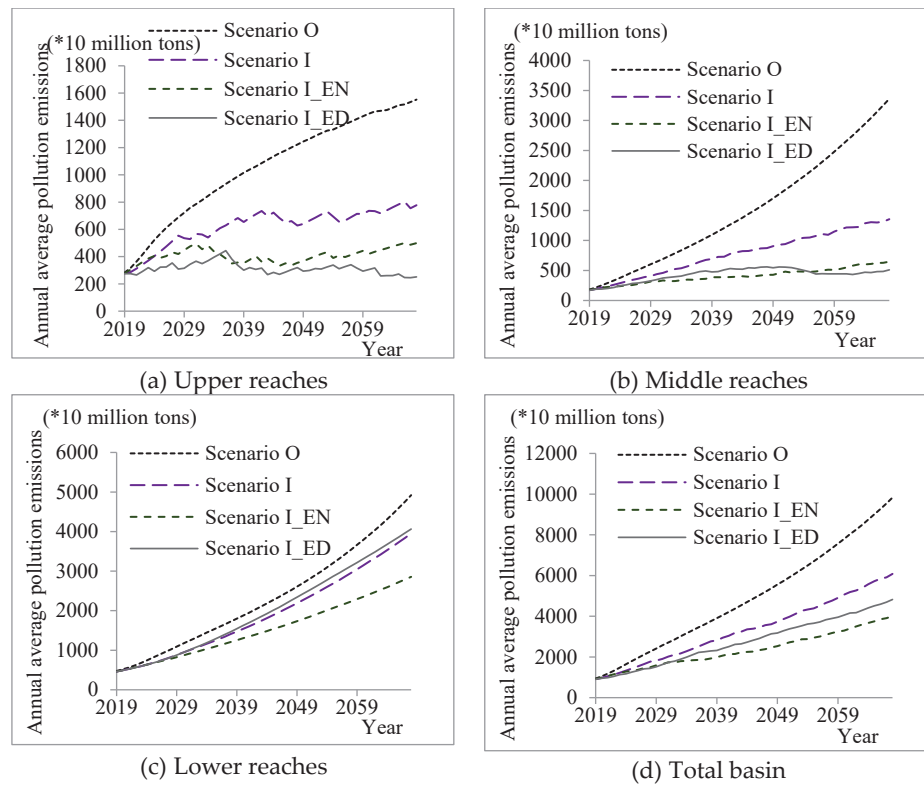


Figure 7. Simulation of the evolution of the overall ecological situation in the Yellow River basin under different scenarios.

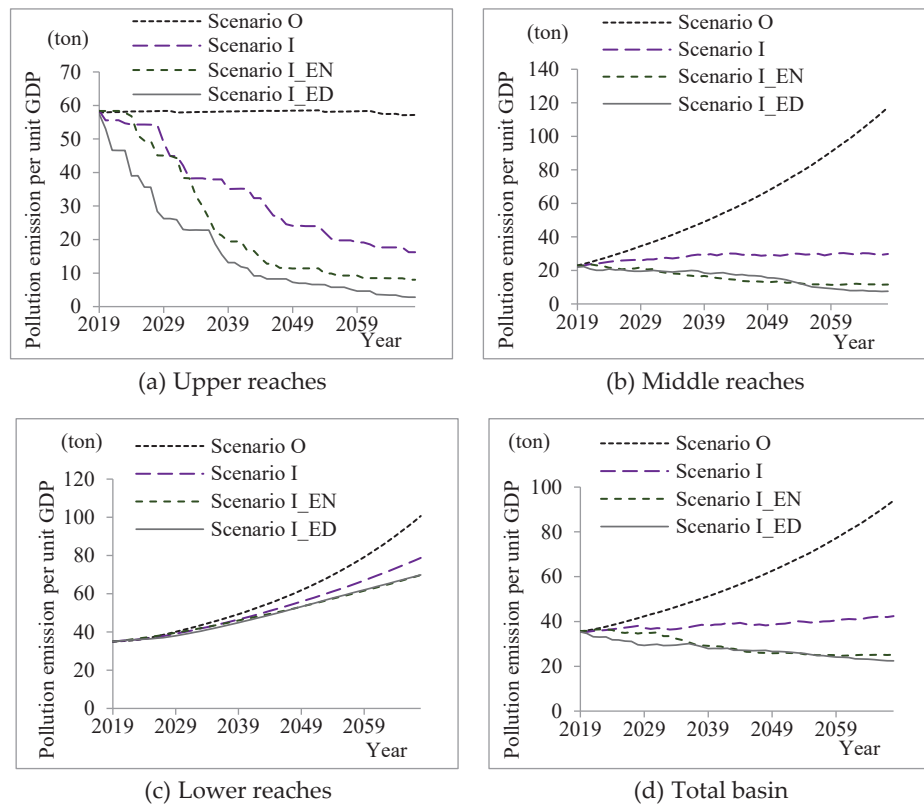


Figure 8. Simulation of the evolution of the pollution emissions per unit of GDP in the Yellow River Basin under different scenarios.

As shown in Figure 8, overall, scenario I_ED (the green innovation and differential ecological and environmental constraint model for upper, middle, and lower reaches) has the best effect on the reduction of pollution emissions per unit of GDP, especially in the upper reaches of the Yellow River Basin, where the pollution emissions per unit of GDP under scenario I_ED are significantly better than those in other scenarios. Combining economic development, resource consumption, and the ecological environment, the implementation of segmented-control ecological and environmental constraint strategies for the upper, middle, and lower reaches of the Yellow River basin, and the appropriate strengthening of ecological and environmental protection in the middle and upper reaches, are important in order to promote the overall high-quality development of the Yellow River Basin.

5.4. Policy Implications of the Research Results

The simulation results show that under the existing development model, the economic development of all of the provinces in the Yellow River Basin will be subject to different degrees of resource and ecological constraints, and different policy scenarios significantly affect the evolutionary trends of the economic development, resource consumption, and environmental pollution in each province in the Yellow River Basin, showing different mechanisms to approach the bottleneck of resource and ecological constraints. The effects of the same policy scenario in different provinces also vary. The following policy implications are based on the research results.

(i) Green innovation economic incentive policies have significant effects on the reduction of resource consumption, the optimization of the ecological environment, and the reduction of the lower reaches' impact of environmental pollution in the region, especially reducing resource consumption and the ecological environment. However, from the perspective of promoting economic growth, the Shandong, Henan, and Ningxia provinces are generally seen to lag behind the crude development model under a single green innovation incentive model (see Table 3).

(ii) Strict ecological constraints do not always harm the economy. From the economic growth trends simulated over 50 years of evolution, Inner Mongolia and Shaanxi instead achieve optimal economic growth under the scenario that imposes strict environmental constraints; at the same time, resource consumption under this scenario is much better than under the crude development model, and a comparison with the green innovation economic incentive scenario reveals that the scenario with a dual combination of green innovation and ecological and environmental protection yields better results in terms of the promotion of economic development and the reduction of resource consumption. This scenario is also significantly better than the crude development model and the green innovation incentive model in terms of the ecological environment and the impact on the lower reaches (see Table 4).

(iii) The implementation of the segmented-control ecological and environmental protection strategy has a much better impact on economic growth than other scenarios, but in terms of the reduction of resource consumption, this scenario is significantly inferior to the green innovation incentive scenario and the combined innovation and environmental constraint scenario; in terms of ecological environment and the impact on the lower reaches, this scenario is significantly better than the crude development model and the green innovation incentive scenario, but is not significantly different from the combined innovation and environmental constraint scenario. The difference between this scenario and the combined innovation and environmental constraint scenario is not significant (see Table 5). It can be seen that the implementation of a segmented-control strategy for ecological and environmental protection across the Yellow River Basin can better guarantee long-term economic growth. Because the high-quality development of the middle and upper reaches of the Yellow River Basin is constrained by the business environment, human living environment, salary and welfare, development space, and other conditions, and faces huge competitive pressure regarding green innovation, implementing a synergistic development model with upper and lower reach linkages, complementary advantages,

and a reasonable division of labor can not only achieve sustainable economic growth but also reduce resource consumption and environmental pollution more efficiently. In the long term, this could better promote the high-quality development of the whole Yellow River Basin.

6. Conclusions and Discussion

Based on an agent-based model, this study took the empirical data of 115 prefecture-level cities in nine provinces and regions of the Yellow River Basin from 2010 to 2018 as a sample, and took the coordinated development of economy–resources–environment–society as the goal, constructing 115 agent models with different attributes, constraints, behavior rules, interaction rules, and autonomous response capabilities. It used computational experiment methods derived from the social sciences to simulate the evolutionary path of eco-protection and high-quality development under different policy scenarios, such as green innovation, ecological environment constraints, ecological compensation, and so on. The simulation results show that under the existing development model, the economic development of all of the provinces in the Yellow River Basin will be subject to different degrees of resource and ecological constraints, and different policy scenarios significantly affect the evolutionary trends of economic development, resource consumption, and environmental pollution in each province in the Yellow River Basin, showing different mechanisms to approach the bottleneck of resource and ecological constraints.

Existing research on the high-quality development of the Yellow River Basin is mostly based on the evaluation of multiple indicators. These studies mostly use empirical data to carry out the comparative analysis of different temporal and spatial dimensions, and rarely involve the prediction of future evolution trends under different scenarios. With the combination of empirical research with the SD method, Jiang et al. (2021) [32] simulated the dynamic process of system development. Jiang's model was based on the indicators of the evaluation system and the causal relationship between the indicators. The advantages of such a method come from the intuitive modeling method; the easy-to-understand, clear causal relationship between the variables; and the ease of reflection of the error of simulation by comparing the simulation results of each index with empirical data. However, the research objects and conclusions of such methods remain at the macro level, which is difficult to reveal the microdrivers of variant changes, and it is also difficult to reflect the impact of individual heterogeneity, individual decision-making uncertainty and individual interaction on the macro level of the system.

On the other hand, from the perspective of scenario analysis, Jiang's model simulated the evolution results under three different scenarios, including economic growth priority, environmental protection, and equal emphasis on economic development and environmental protection. The above scenarios are essentially one or more dimensions that constitute the evaluation system, and the simulation results only reflect the linkage and coupling relationship between the dimensions. Our model benefits from the flexibility of multi-agent attributes and behavior rules. It focuses on the possible policy scenarios of high-quality development in the Yellow River Basin, and explores the optimization space of policy design, which can more deeply reveal the action mechanism and macro-level effect of a specific policy. Compared with existing research, the proposed model reveals the microdrivers of the macro changes. Its outstanding advantage is that it is convenient for researchers to analyze the motivation at the micro level and observe the overall emergence at the macro level. In this way, it is possible to visually simulate the development and evolution of a complex system under different scenarios, based on empirical data and with computers as tools. The virtual–real linkage provides a guarantee for the reliability of research. Researchers can verify and adjust the attributes or rules of agents at any time by comparing the simulation results with empirical data. This helps the constructed artificial system to map the real system well, on the one hand, and provides more abundant scenarios than the real system, on the other hand.

Scenario modeling and evolutionary simulation based on multiple agents are very effective tools in bottom-up research; however, there may be limitations in the modeling process, which should be continuously improved and expanded in future research. In this paper, 115 prefecture-level cities were used as agents for simulation. The study did not consider the behavioral characteristics and interactions of more micro-level individuals, such as different industries, specific enterprises, residents, and so on. In the future, the interaction research of agents at different levels should be strengthened. In addition, this study took language probability as the basis of agent decision-making. It did not consider the mutation problem of the agent itself. In the future, it will be necessary to enrich the agent rule-learning algorithms, such as the genetic algorithm, particle swarm optimization, and ant colony algorithm, etc. In addition, it is also important to strengthen the integration of different models, consider the complexity of interaction between agents, and expand the scope of application of the model.

Author Contributions: Conceptualization and methodology, A.Z.; original draft preparation, J.W.; data curation Z.S.; review and editing, H.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by “the Major Projects of National Social Science Fund”, China (Grant No. 19ZDA080).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sun, Q.; Zhang, X.; Zhang, H.; Niu, H. Coordinated development of a coupled social economy and resource environment system: A case study in Henan Province, China. *Environ. Dev. Sustain.* **2018**, *20*, 1385–1404. [[CrossRef](#)]
2. Ma, H.M.; Ding, Y.; Yang, Q. Regional Ecology-Economy-Society Coordinated Development Evaluation Model and Application. *Stat. Decis.* **2019**, *35*, 75–79. [[CrossRef](#)]
3. Shi, B. Study on the development path of central cities in the Yellow River Basin. *J. Humanit.* **2020**, *1*, 5–9. [[CrossRef](#)]
4. Ren, B.P.; Zhang, Q. The Strategic Design and Supporting System Construction of High-quality Development in the Yellow River Basin. *Reform* **2019**, *10*, 26–34.
5. Wang, Y.Q.; Li, G.P. Sustainable simulation of ecological environment and socio-economic development in the Yellow River Basin based on the SD model. *Arid Land Geogr.* **2022**, *45*, 901–911.
6. Geng, Y.W.; Chen, W.Q.; Zhang, J.X.; Wang, K.; Zhang, F.; Ma, Y.H. Evolution Characteristics and Scenario Simulation of Social-Ecological-Productional Landscape Resilience in Areas Along Mainstream of Yellow River in He’nan Province. *Bull. Soil Water Conserv.* **2021**, *41*, 181–190. [[CrossRef](#)]
7. Fang, C.; Liu, H.; Li, G. International progress and evaluation on interactive coupling effects between urbanization and the eco-environment. *J. Geogr. Sci.* **2016**, *26*, 1081–1116. [[CrossRef](#)]
8. Sun, Z.; Zhu, X.; Pan, Y.; Zhang, J.; Liu, X. Drought evaluation using the GRACE terrestrial water storage deficit over the Yangtze River Basin, China. *Sci. Total Environ.* **2018**, *634*, 727–738. [[CrossRef](#)]
9. Srinivasan, V.; Seto, K.C.; Emerson, R.; Gorelick, S.M. The impact of urbanization on water vulnerability: A coupled human-environment system approach for Chennai, India. *Glob. Environ. Chang.* **2013**, *23*, 229–239. [[CrossRef](#)]
10. Ma, T.; Chen, M.; Wang, C.; Mao, Z.; Jiang, M. Study on the environment-resource-economy comprehensive efficiency evaluation of the biohydrogen production technology. *Int. J. Hydrog. Energy* **2013**, *38*, 13062–13068. [[CrossRef](#)]
11. Bastianoni, S.; Coscieme, L.; Pulselli, F.M. The input-state-output model and related indicators to investigate the relationships among environment, society and economy. *Ecol. Model.* **2016**, *325*, 84–88. [[CrossRef](#)]
12. Estoque, R.C.; Murayama, Y. Social-ecological status index: A preliminary study of its structural composition and application. *Ecol. Indic.* **2014**, *43*, 183–194. [[CrossRef](#)]
13. Li, Y.; Li, Y.; Zhou, Y.; Shi, Y.; Zhu, X. Investigation of a coupling model of coordination between urbanization and the environment. *J. Environ. Manag.* **2012**, *98*, 127–133. [[CrossRef](#)] [[PubMed](#)]
14. Aldieri, L.; Vinci, C.P. Climate change and knowledge spillovers for cleaner production: New insights. *J. Clean. Prod.* **2020**, *271*, 122729. [[CrossRef](#)]
15. Zhao, Y.; Wang, S.; Zhou, C. Understanding the relation between urbanization and the eco-environment in China’s Yangtze River Delta using an improved EKC model and coupling analysis. *Sci. Total Environ.* **2016**, *571*, 862–875. [[CrossRef](#)]

16. Xing, L.; Xue, M.; Hu, M. Dynamic simulation and assessment of the coupling coordination degree of the economy–resource–environment system: Case of Wuhan City in China. *J. Environ. Manag.* **2019**, *230*, 474–487. [[CrossRef](#)]
17. Shi, Y.; Zhu, X.; Sun, X.; Li, Y.; Wei, T. A dynamic assessment for the coordination between economic development and the environment: A case study of Lianyungang, China. *Acta Ecol. Sin.* **2010**, *30*, 4119–4128. [[CrossRef](#)]
18. Han, R.; Tong, L.; Zhu, S.; Lu, Z. The Coordinated Development of Economy and Environment Based on ARMA Model in Shenyang Economic Zone. *Sci. Geogr. Sin.* **2014**, *34*, 32–39. [[CrossRef](#)]
19. Li, Y.; Wu, Y.X.; Zeng, Z.X.; Guo, L. Research on Forecast Model for Sustainable Development of Economy-Environment System Based on PCA and SVM. In Proceedings of the 2006 International Conference on Machine Learning and Cybernetics, Dalian, China, 13–16 August 2006; pp. 3590–3593. [[CrossRef](#)]
20. Turner, K. Economics and wetland management. *Ambio* **1991**, *20*, 59–63.
21. Zhao, L.; Zhang, G. Evaluation and welfare effect of coordinated ecological development of the Beijing–Tianjin–Hebei region. *Chin. J. Popul. Resour.* **2021**, *19*, 283–290. [[CrossRef](#)]
22. Deng, C.; Li, H.; Peng, D.; Liu, L.; Zhu, Q.; Li, C. Modelling the coupling evolution of the water environment and social economic system using PSO-SVM in the Yangtze River Economic Belt, China. *Ecol. Indic.* **2021**, *129*, 108012. [[CrossRef](#)]
23. Kaplan, R.S.; Norton, D.P. The Balanced Scorecard—Measures that Drive Performance. *Harv. Bus. Rev.* **1992**, *70*, 71–79. [[PubMed](#)]
24. Kaplan, R.S.; Norton, D.P. *The Execution Premium: Linking Strategy to Operations for Competitive Advantage*; Harvard Business School Press: Boston, MA, USA, 2008; ISBN 9781422121160.
25. Kaplan, R.S.; Norton, D.P. Using the balanced scorecard as a strategic management system. *Harv. Bus. Rev.* **1996**, *74*, 75–85.
26. Ahn, H. Applying the balanced scorecard concept and experiential report. *Long Range Plan.* **2001**, *34*, 441–461. [[CrossRef](#)]
27. Acosta, L.A.; Rounsevell, M.; Bakker, M. An Agent-Based Assessment of Land Use and Ecosystem Changes in Traditional Agricultural Landscape of Portugal. *Int. J. Inf. Manag.* **2015**, *6*, 55–80. [[CrossRef](#)]
28. Xu, Q.; Yang, K.; Wang, G.; Yang, Y. Agent-based modeling and simulations of land-use and land-cover change according to ant colony optimization: A case study of the Erhai Lake Basin, China. *Nat. Hazards* **2015**, *75*, 95–118.
29. Liu, R.; Jiang, D.; Shi, L. Agent-based simulation of alternative classroom evacuation scenarios. *Front. Archit. Res.* **2016**, *5*, 111–125. [[CrossRef](#)]
30. Zhang, N.; Lu, Y.; Chen, J.; Hwang, B. An agent-based diffusion model for residential photovoltaic deployment in Singapore: Perspective of consumers’ behavior. *J. Clean. Prod.* **2022**, *367*, 132793. [[CrossRef](#)]
31. Macal, C.M.; North, M.J. Tutorial on agent-based modelling and simulation. *J. Simul.* **2010**, *4*, 151–162. [[CrossRef](#)]
32. Jiang, L.; Zuo, Q.; Ma, J.; Zhang, Z. Evaluation and prediction of the level of high-quality development: A case study of the Yellow River Basin, China. *Ecol Indic.* **2021**, *129*, 107994. [[CrossRef](#)]

Article

Practical Formalism-Based Approaches for Multi-Resolution Modeling and Simulation

Jang Won Bae¹ and Il-Chul Moon^{2,*}¹ School of Industrial Management, Korea University of Technology and Education, Cheonan 31253, Korea² Department of Industrial and Systems Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34129, Korea

* Correspondence: icmoon@kaist.ac.kr

Abstract: Multi-resolution modeling (MRM) has been considered as an ideal form of simulation to acquire low-resolution scalability as well as high-resolution modeled details. Although both practical and theoretical interests exist in MRM, actual implementations were quite different in terms of cases and methods. Specifically, MRM implementations range from parameter-based interoperation to model exchanges with different resolutions, yet it is difficult to observe a method that focuses on both of these aspects. To this end, this paper introduces a formalism or multi-resolution translational Discrete Event System Specification (MRT-DEVS). Focusing on the practical perspective, MRT-DEVS intends to ease the implementation's difficulty and reduce the simulation's execution costs. Specifically, MRT-DEVS embeds state and event translation functions into the model's specifications so that it enables MRM with less complex mechanisms in terms of operations. Using the provided case study and a reduction to other MRM methods, the theoretical soundness of the proposed method is supported. Moreover, we discussed the pros and the cons of the proposed method from various MRM perspectives. We expect that with all the provided information, MRMS users would consider the proposed method as a practical option to implement their models.

Citation: Bae, J.W.; Moon, I.-C.Practical Formalism-Based Approaches for Multi-Resolution Modeling and Simulation. *Systems* **2022**, *10*, 174. <https://doi.org/10.3390/systems10050174>

Academic Editors: Philippe Mathieu, Juan M. Corchado, Alfonso González-Briones and Fernando De la Prieta

Received: 19 July 2022

Accepted: 23 September 2022

Published: 29 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: modeling and simulation; practical formal method; multi-resolution; discrete event system specification

1. Introduction

This paper introduces a formalism for multi-resolution translational discrete event system specification (MRT-DEVS) to enable MRMs with less complex mechanisms with respect to operations. The proposed method is based on discrete event system specification (DEVS) formalism [1]. The previous DEVS-based formalisms are solid and sound, with complete expressions in MRM scenarios. However, we note that their hidden challenge is the difficulty that field engineers face when implementing multi-resolution (MR) models by following such formalisms. For example, MR modeling with previous formalisms requires components for different resolution models and resolution conversions, which eventually increases the total implementation's complexity. The proposed MRT-DEVS intends to ease the implementation difficulty by relaxing an assumption of the DEVS. Specifically, MRT-DEVS absorbs the resolution conversion into the models for each resolution, and the state information is converted within the model, which does not require additional models for the conversion's purpose. This reduces the practical implementation burden, but relaxing the assumption can cause a trade-off relationship, which is investigated from various perspectives.

The utility of simulation often originates from the emergence of target systems that is difficult to anticipate before the execution of actual system operations as well as the comprehensive description of systems. To embody these two aspects, simulation practices need to be comprehensive for the expression and scalable for the emergence [2,3].

For example, if only a handful of objects are involved in a simulation, the complexity that lies in the system would be difficult to observe in emergent phenomena. On the other hand, to simulate many objects, the descriptions of the objects may be over-abstracted and even simplified, which would fail to deliver a comprehensive description. Therefore, the objectives of comprehensive expression and scaled emergence result in a dilemma [4,5] when we assume limited resource in modelers, computing resources, available data, etc.

To confront this dilemma, researchers proposed a form of multi-resolution modeling (MRM) that describes a target system with different resolution components; thus, the model's details can change according to their compositions [6–8]. This approach can resolve the dilemma by implementing a high-resolution model for completeness and a low-resolution model for scalability [9]. Moreover, the resolution levels of model components change during simulation executions using internal or external triggers (e.g., events coming from external or inner states) through an arbiter that is implemented by users [10,11]. Ideally, MRM simulations may execute the modeled details with accelerated or simplified behaviors that intervene in periods. This ideal framework has drawn interest from practitioners and researchers; thus, works related to the MRM approach have been published.

Although both practical and theoretical interests exist in MRM, actual implementations were quite different in terms of cases and methods. The implementation of MRM ranges from a parameter-based interoperation to model exchanges among different resolutions. In particular, model exchange has been conducted in rather complex ways because the objects and the associated information, such as model states, should be properly converted with respect to a new resolution. In this conversion process, an ad hoc approach is often applied due to its lower development costs [12–15]. However, we argue that the conversion should be rather disciplined so that the developed models can be maintainable, flexible, and transparent. A number of proposals have followed such a suggestion, especially the formalism-based method: MRM formalisms [10,11] generally specify models for different resolutions and require an additional model to deal with the conversion process.

After introducing the MRT-DEVS formalism, we argue its expression in two ways: First, we show an illustrative example with a dynamic resolution-change sequence to demonstrate how MRT-DEVS models MR scenarios; second, we reduce MRT-DEVS formalism to previous MRM formalisms so that MRT-DEVS can convey the same amount of information that was expressed in previous formalisms. Lastly, we discuss trade-off relationships in MRT-DEVS.

2. Previous Research

This section provides a survey of the previous research studies on MRM. Specifically, we examine how MRM works, particularly in MRM applications and methods.

2.1. Multi-Resolution Modeling Applications

Before diving into the survey, we intend to begin with a formal definition of MRM. In modeling and simulation (M&S), the resolution of a model refers to the level of an abstraction of a target system, which could differ according to the objective of simulations performed with the model. For example, when we simulate a squad-level engagement, its objectives could vary with respect to the modeler's interests: For instance, one may have an interest in either squad maneuver tactics or individual engagement tactics in the squad. In this sense, the model's resolution has become a frequent topic in discussions among M&S experts [16–18].

MRM, as a concept extended from model resolution, describes a target system as a set of models with different resolution levels. More formally, MRM was defined as “*building a single model with alternative user models that have different levels of resolution for the same phenomena*” [19,20]. To meet the potential capability expected from the definition, many MRM research studies have been conducted and developed in various domains.

Rauscher et al. conducted a large number of aqua planet experiments using the multi-resolution model for predication across different scales of hydrostatic dynamical cores [21]. Jeschke and Uhrmacher applied MRM methods to develop a molecular crowding simulation model in which a combination of individual and lattice-population-based algorithms were used to manage macromolecular crowding phenomena [22]. Tan et al. dealt with MRM problems, such as aggregation and disaggregation among federates of different resolution levels, in a high level architecture (HLA) environment [23]. Zeigler and Kim proposed an efficient approach for MRM, particularly for UAV-based service systems using DEVS-based system entity structures (SEs) [24].

Although MRM is used to capture various aspects of a target system, there is another method used for applying MRM, such as securing simulation efficiency. Yoon et al. [25] developed grid models of focused ultrasound propagation, and they assessed their computational efficiency by various combinations of resolution grid settings. Ringler et al. [26] proposed an MRM method for the global ocean system, and the simulation based on the proposed method was efficiently evaluated by using MRM concepts. Choi et al. [27] applied an MRM method to improve simulation execution speeds while reducing errors and increasing future model reuses.

2.2. Multi-Resolution Modeling Methods

Although MRM applications are widespread, many researchers have shown an interest in using different methods to develop MR models efficiently. In MRM contexts, maintaining the consistency between different levels of resolution models is important, because models of various resolution levels have to be exchanged during the simulation's execution [28–30]. Consistency maintenance is mainly about current model information, such as the model's states and external inputs to the model, and the conversion process on this information is referred to as state and event translation in this paper.

Although a number of ways to realize MRM exist, we focus on the formalism-based methods. Considering the importance of consistency maintenance in MRM, we also argue that a method for MRM should provide a clear view of the conversion process so that users can understand how it works and eventually manipulate it for what they want to develop. Among M&S formalisms, DEVS is often considered as one of the remarkable candidates because of its systematic modeling capability. Hence, a number of MRM methods have been based on DEVS [11,31–33], and the proposed method is another variant of DEVS.

Among past DEVS variants, we focused on Baohong's [11] and Hong's [33] studies because their approaches are similar to ours. Baohong proposed a formal specification for MRM based on DEVS. To this end, he proposed a concept of an MR model family (MF) where different resolution models and their relations are involved. Technically, he made two efforts for his proposed MRM formalism: (1) he introduced a set of model resolutions (γ) and the associated functions (ψ , π , and χ) for embedding model resolutions into his work; (2) he adapted the specifications from dynamic structure DEVS [34] for describing more resolution changes via a change in model structure.

Hong also proposed a formalism for MR modeling and simulation, and it focused on the method of implementing MR simulations with existing models (e.g., federates in HLA) and not methods for specifying the actual MR model. In any case, the main body of her formalism was formed using a structure similar to Baohong's, yet Hong introduced a concept of resolution-conversion protocols (using C_R and Y_R) that enables triggering a model's resolution changes from its inner components.

Although it is obvious that the above-mentioned MRM methods provide sound theoretical frameworks for MRM, two drawbacks cause hesitancy in their applications in actual model development. The first is that their formalisms possess structure that is too and have too many elements that should be specified by users. This trait may oppose the direction of DEVS because DEVS has become one of the most popular formalisms in M&S due to its simple and explicit expression. The second drawback is that according to their complex structure, its simulation would be inefficiently executed. Moreover, in handling

resolution changes, they introduced additional components, such as a converter (χ) or additional event exchange protocols (C_R and Y_R), and these efforts require more costs, including increased simulation time and resource usage.

To tackle these problems, this paper proposes an MRM formalism. The proposed method is an extension of DEVS formalism, but the modification is minimized to convey the easy modeling resulting from DEVS. Moreover, the proposed method hides most of the converter and event-exchange protocols for model resolution changes, which were enabled by releasing modular characteristics in DEVS. Through this improvement, the cost of MRM development and its simulation in practice would be reduced. The details on the pros and the cons of the proposed method are discussed in a later section.

3. Multi-Resolution Translational DEVS Formalism

This section introduces the proposed formalism, MR translational discrete event system specification formalism (MRT-DEVS), with respect to two aspects: the first is the definition and semantics of the formalism; the second is the operation of the specified MR system following the MRT-DEVS formalism.

3.1. Specification of MRT-DEVS Formalism

In this section, we introduce the specifications of the MRT-DEVS formalism, which extends the DEVS formalism with a minimal variation in order to enable MR modeling. MRT-DEVS consists of two definitions: the atomic model and the coupled model. Equations (1) and (2) define the atomic model and the coupled model, respectively.

$$\begin{aligned}
 AM = & \langle X, Y, S_M, S_R, \delta_{ext}, \delta_{int}, \delta_{res}, \lambda, ta \rangle \\
 & X \text{ and } Y \text{ are sets of input and output events} \\
 & S_M \text{ and } S_R \text{ are sets of model and resolution states} \\
 & \delta_{ext} : Q \times X \rightarrow S_M, \text{ external transition function} \\
 & \quad \text{where } Q = \{(s, e) | s \in S_M, 0 \leq e \leq ta(s)\} \\
 & \delta_{int} : S_M \rightarrow S_M, \text{ internal transition function} \\
 & \delta_{res} : S_M \times S_R \rightarrow S_R, \text{ resolution transition function} \\
 & \lambda : S_M \rightarrow Y, \text{ output function} \\
 & ta : S_M \rightarrow R^+, \text{ time advance function}
 \end{aligned} \tag{1}$$

Equation (1) enumerates the atomic model tuple. In detail, X is a set of input events, Y is a set of output event, and S_M and S_R are sets of state variables representing the model and the resolution states, respectively. δ_{ext} is the external transition function that determines the next S_M with the current S_M and the input event instance; δ_{int} is the internal transition function that determines the next S_M with the current S_M ; δ_{res} is the resolution transition function that determines the next S_R with the current S_R and the current S_M ; λ is the output function to generate an output event instance based on the model state S_M ; ta specifies the lifespan of a model state (i.e., next time advance time), $t_{adv} \in [0, \infty)$, by depending on S_M .

Compared to the original DEVS formalism, the atomic model of MRT-DEVS adds S_R and δ_{res} , which are the resolution state and resolution transition function. This formalism design intends to minimize the modifications of potentially existing atomic models, and particularly, the formalism limits alterations on existing transition functions of δ_{ext} and δ_{int} because these are often considered as key features for modeling discrete-event systems. For example, the proposed formalism minimizes the modification on DEVS by adding two tuple elements regarding the model's resolution to the DEVS atomic model.

$$\begin{aligned}
CM = & \langle X, Y, M, S_R, RMS, \delta_{res}, \delta_{strans}, \delta_{etrans} \rangle \\
& X \text{ and } Y \text{ are sets of input and output events} \\
& M \text{ is a set of model components} \\
& S_R \text{ is a set of resolution states} \\
& RMS \subseteq S_R \times \Sigma_r (r \in S_R), \text{ resolution model structure} \\
& \text{ where } \Sigma_r = \{M_r, EIC_r, EOC_r, IC_r, Select_r\}, \\
& \text{ coupling relations at model resolution state } (r) \tag{2} \\
\delta_{res} : & S_R \times \cup_{m \in \Sigma_r.M_r} m.S \rightarrow S_R, \\
& \text{ resolution transition function} \\
\delta_{strans} : & S_R \times \cup_{m \in \Sigma_r.M_r} m.S \rightarrow \cup_{m \in \Sigma_r.M_r} m.S, \\
& \text{ state translation function} \\
\delta_{etrans} : & S_R \times X \rightarrow m.X, \text{ where } m \in \Sigma_r.M_r \\
& \text{ event translation function}
\end{aligned}$$

Equation (2) enumerates the coupled model tuple: X is the input events, Y is the output events, and M is a set of model components, which are identical to the DEVS coupled model. To embed the model resolution concept in the coupled model, MRT-DEVS introduces a resolution state (S_R) and the associated transition functions (i.e., δ_{res} , δ_{strans} , and δ_{etrans}), which are not allowed in the classic DEVS coupled model. Moreover, MRT-DEVS holds a resolution model set (RMS), which is defined as a set of activated models and their coupling relations for a certain resolution state. Specifically, at a certain resolution state denoted by S_R , RMS becomes a tuple of the coupling information according to the resolution state, Σ_r . In this paper, Σ_r consists of activated models (M_r) and a union set of external input couplings, EIC_r ; external output couplings, EOC_r ; internal couplings, IC_r ; and a tie-breaking function, $Select_r$, in the classic DEVS coupled model. We denote this union of coupling information as Σ by following the notation of the DS-DEVS [34].

MRT-DEVS proposes a transition function, δ_{res} , for S_R . By definition, δ_{res} accesses information from two separate sources: the specifying model's resolution state, S_R , and the union of the currently activated model's state, $\cup_{m \in \Sigma_r.M_r} m.S$. Here, it should be noted that we are suggesting that a coupled model accesses the states of its components. This has been regarded as a violation of the black-box assumption, or modular modeling, which has been prohibited in DEVS. We will return to this discussion in later sections in this paper. If we accept the definition of RMS , the definition of δ_{res} becomes trivial because it changes the resolution state's information, S_R , by the state of the child models that is affected by input events or the expiration time.

While accepting the definition of δ_{res} requires the read privilege of the child models, or components, the state translation function of δ_{strans} requires writing privileges. δ_{strans} is the function to change the state information in the activated models at a certain resolution state r ($\Sigma_r.M_r$). Previous approaches used event messages to change the state information of the activated models, but we note that this increases modeling and simulation costs, such as increased message passing counts and the modeling concerns for handling them. Specifically, Lee and Kim [35] identified that message passing is the most influential factor for simulation overheads. Hence, we relaxed the black-box model assumption to alter the state information of the child models directly. The justification for this assumption violation will follow in the Discussion section. The main role of this state translation is data aggregation and disaggregation because of resolution changes. For instance, a low-resolution model will generate a set of state information for a high-resolution model through data disaggregation, and data aggregation denotes the scenario from a high resolution to a low resolution. Eventually, every piece of information is stored as state information; thus, data aggregation/disaggregation can be realized by manipulating state information.

Finally, MRT-DEVS requires a function for event translation, δ_{etrans} , between resolution changes. An event from the outside model assumes a certain resolution of a receiver model,

but the receiver model might change its resolution by alternatively using a different set of RMS. Therefore, the outside event needs to be adjusted by the activated model resolution that is determined by S_R . As a notation, δ_{etrans} accepts S_R and the input event from the outside X , which is indifferent to the resolution state of the receiver model, and δ_{etrans} turns the input event into inputs of the activated components, which is called an event translation adjusted to fit the model of a certain resolution, S_R . In previous MR methods, this was implemented as another converter model, and this method requires exchanges of event messages as well as the resolution setting message. Therefore, we removed the converter model by violating the black-box model assumption; more specifically, we replaced the converter model with a function of the coupled model.

3.2. Operation of MRT-DEVS Formalisms

Once we define the MRT-DEVS formalism, it is important to understand how the specified models work (i.e., their operational procedures). The proposed MRT-DEVS defines the resolution's state transition in both atomic and coupled models; thus, this paper introduces the operation of both models in turn.

Algorithm 1 illustrates the operation flow of the atomic model in MRT-DEVS in contrast to the classic DEVS (additional parts for MRT-DEVS are presented in bold in Algorithm 1). In the classic DEVS, the atomic model is introduced for modeling behaviors in discrete event systems. Once users design atomic models, the DEVS simulation algorithm supports the simulation of system behaviors with respect to a discrete event system. For example, setting up the last event time (tl) and next event time (tn) is essential in discrete event simulation.

Algorithm 1 Simulation Algorithm for MRT-DEVS Atomic Models

Require: *parent* {parent model},
 tl {last event time},
 tn {next event time}

- 1: **while** not at the end of simulation **do**
- 2: **if** receiving an initialization message at time t **then**
- 3: $tl = t - e$ { e : elapsed time}
- 4: $tn = tl + ta(S_M)$ { S_M : model state}
- 5: **else if** receiving a state transition event at time t **then**
- 6: $y = \lambda(S_M)$
- 7: send an output event to *parent* (y, t)
- 8: $S_M = \delta_{int}(S_M)$
- 9: $S_R = \delta_{res}(S_M, S_R)$ { S_R : resolution state}
- 10: $tl = t$
- 11: $tn = tl + ta(S_M)$
- 12: **else if** receiving an input event x at time t **then**
- 13: $e = t - tl$
- 14: $S_M = \delta_{int}(S_M, e, x)$
- 15: $S_R = \delta_{res}(S_M, S_R)$
- 16: $tl = t$
- 17: $tn = tl + ta(S_M)$
- 18: **end if**
- 19: **end while**

As Algorithm 1 shows, the simulation algorithm for the DEVS atomic model handles events, or messages, from two perspectives. In one case, it involves receiving a state transition message from the parent model where the atomic model is involved. In this case, the atomic model generates an output event based on its current state (λ), changes its model state (δ_{int}), and sets up a time duration for staying in the changed state (time advance, ta). In the other case, it receives an input event from the outside. In this case, the atomic model changes its state depending on the current state and the input event (δ_{ext}),

and similarly to the previous case, it updates the time duration for the new state with its time advance function.

As mentioned above, the proposed MRT-DEVS atomic model does not alter the classic one much and introduces resolution states (S_R) and the associated transition function (δ_{res}). Similarly, the simulation algorithm for the MRT-DEVS adds calling the resolution state-transition function after the model's state transitions (δ_{int} (line 9) and δ_{ext} (line 15), which are marked in bold in Algorithm 1) into the classic state.

Algorithm 2 illustrates the simulation algorithm of the coupled model in MRT-DEVS, which is also based on DEVS (additional parts are marked in bold). In DEVS, the coupled model is for reflecting the structure of a target system. However, the simulation algorithm for the coupled model coordinates the simulation's execution within it and its components; specifically, it helps synchronize simulation times and manage event exchanges among the coupled model and its components.

Setting aside the initialization, the simulation algorithm of the DEVS-coupled model considers three event cases (refer to Algorithm 2). The first case is when the coupled model receives a state transition event, which means any component of the coupled model is ready to change its state (either or both of its model and resolution states). In this case, the simulation algorithm sends a state transition event to an imminent child, m^* , and its time advance is identical to its next event time, tn , and the imminent child, if it is an atomic model, changes its state following a case in which it receives a state transition message in Algorithm 1. Then, the time information, such as tl and tn , is updated by considering the results of the state transition.

Second, when the associated coupled model obtains an input event, x , the simulation algorithm finds a set of models, $M_{receiving}$, that is related with the input event. To this discovery, the coupling information takes the role of condition (Σ_r). Then, the simulation algorithm requests every model in $M_{receiving}$ to handle the input event (refer to when receiving an input event case in Algorithm 1). Considering the result of the followed state transitions, the simulation algorithm update its time information.

Lastly, when the coupled model receives an output event, y , generated from its components, the simulation algorithm defines $M_{receiving}$ of the output event using coupling relations. Similarly, output event y is forwarded to every model in $M_{receiving}$. Specifically, in a case in which the parent model is identified as a receiving model (i.e., the parent model is included in $M_{receiving}$), the simulation algorithm sends y as an output event of the parent model; otherwise, it sends y as an input event of the component models. Then, an update of the time information follows.

Based on the above explanation, the modifications for the simulation algorithm for the MRT-DEVS-coupled model are conducted in three parts: resolution transition function (δ_{res}), state translation function (δ_{strans}), and event translation function (δ_{event}). Similarly to the MRT-DEVS atomic model case, the resolution transition function calls after resolution state transitions of the components. Specifically, such transitions can occur in the above three cases (i.e., when the cases of receiving a state transition (lines 11–14), an input event (lines 23–26), and an output event (lines 38–41) cases in Algorithm 2). After the resolution state function, the state translation function always follows to update it and the component's states (including model and resolution states). However, the event translation function is called only when an input event is forwarded to components (line 19 in Algorithm 2). Through the event translation function, the input event is transformed into a new form of an input event that is more proper to the resolution state of the receiving component.

The operation flows of MRT-DEVS modify the simulation algorithms of the classic DEVS by adding several function calls for handling state and event conversions due to resolution changes. With such a small modification, MRT-DEVS can not only help enable MR modeling for discrete event simulations but can also aid many users by facilitating easy modeling with DEVS semantics.

Algorithm 2 Simulation Algorithm for the MRT-DEVS-Coupled Model**Require:** $parent, tl, tn$

```

1: while not at the end of simulation do
2:   if receiving an initialization message at time  $t$  then
3:     for  $m$  in  $M$  ( $M$ : children components) do
4:       send initialization message to  $m$ 
5:     end for
6:      $tl = \max\{tl_m | m \in M\}$ 
7:      $tn = \min\{tn_m | m \in M\}$ 
8:      $m^* = \operatorname{argmin}_{m \in M}\{tl_m\}$  ( $m^*$ : imminent child)
9:   else if receiving a state transition event at time  $t$  then
10:    send state transition event to  $m^*$ 
11:    if  $m^*$  changed its resolution state then
12:       $S_R = \delta_{res}(S_R, \cup_{m \in \Sigma_r.M_r} m.S)$ 
13:       $\cup_{m \in \Sigma_r.M_r} m.S =$ 
14:         $\delta_{strans}(S_R, \cup_{m \in \Sigma_r.M_r} m.S)$ 
15:    end if
16:     $tl = t, tn = \min\{tn_m | m \in M\}$ 
17:     $m^* = \operatorname{argmin}_{m \in M}\{tl_m\}$ 
18:   else if receiving an input event  $x$  at time  $t$  then
19:    define  $m_{receiving}$  ( $m_{receiving}$ : models receiving  $x$ ) according to coupling relations
20:    ( $\Sigma_r$ )
21:     $m.x = \delta_{etrans}(S_R, x)$ 
22:    for  $m$  in  $m_{receiving}$  do
23:      send an input event to  $m$  ( $m.x, t$ )
24:    end for
25:    if  $m^*$  changed its resolution state then
26:       $S_R = \delta_{res}(S_R, \cup_{m \in \Sigma_r.M_r} m.S)$ 
27:       $\cup_{m \in \Sigma_r.M_r} m.S =$ 
28:         $\delta_{strans}(S_R, \cup_{m \in \Sigma_r.M_r} m.S)$ 
29:    end if
30:     $tl = t, tn = \min\{tn_m | m \in M\}$ 
31:     $m^* = \operatorname{argmin}_{m \in M}\{tl_m\}$ 
32:   else if receiving an output event  $y$  from  $m \in M$  at time  $t$  then
33:    define  $m_{receiving}$  according to coupling relations ( $\Sigma_r$ )
34:    for  $m$  in  $m_{receiving}$  do
35:      if  $m$  is  $parent$  then
36:        send an output event to  $parent$  ( $y, t$ )
37:      else
38:        send an input event to  $m$  ( $y, t$ )
39:      end if
40:    end for
41:    if  $m^*$  changed its resolution state then
42:       $S_R = \delta_{res}(S_R, \cup_{m \in \Sigma_r.M_r} m.S)$ 
43:       $\cup_{m \in \Sigma_r.M_r} m.S =$ 
44:         $\delta_{strans}(S_R, \cup_{m \in \Sigma_r.M_r} m.S)$ 
45:    end if
46:     $tl = t, tn = \min\{tn_m | m \in M\}$ 
47:     $m^* = \operatorname{argmin}_{m \in M}\{tl_m\}$ 
48:   end if
49: end while

```

4. Case Study

This section presents a case study utilizing a model developed by the proposed MRT-DEVS. By examining this case study, we provide an example of the MRT-DEVS and address the way MRM is properly realized with its semantics.

4.1. Illustrative Example in MRT-DEVS

The example model used in the case study describes a squad-level engagement. Figure 1 shows the structure of the squad-level engagement model. The proposed MRT-DEVS primarily follows DEVS semantics, so the example model has a hierarchical structure: The highest level model consists of blue and red force models. Each force involves a commander and multiple squad models. The command model controls its subordinate squad models, such as in terms of squad maneuver, detection, and engagement. While the number of the subordinate squad models can be determined by users, the case study has two and three squad models in blue and red forces, respectively (see the numbers on the edges in Figure 1). By their combat circumstances, squad models are described at two resolution levels: one is at low-resolution level modeling with respect to squad maneuver and detection. At the low-resolution level, the squad behavior model (dark gray in Figure 1) is the only active component in the squad model; the other resolution level is the high-resolution modeling of squad engagement. At the high-resolution level, three squad member models (light gray Figure 1) are activated in the squad model. The resolution changes in the squad model are triggered by the detection of enemy forces. Squad behavior and squad member models are developed by the MRT-DEVS atomic model; others are developed by the coupled model.

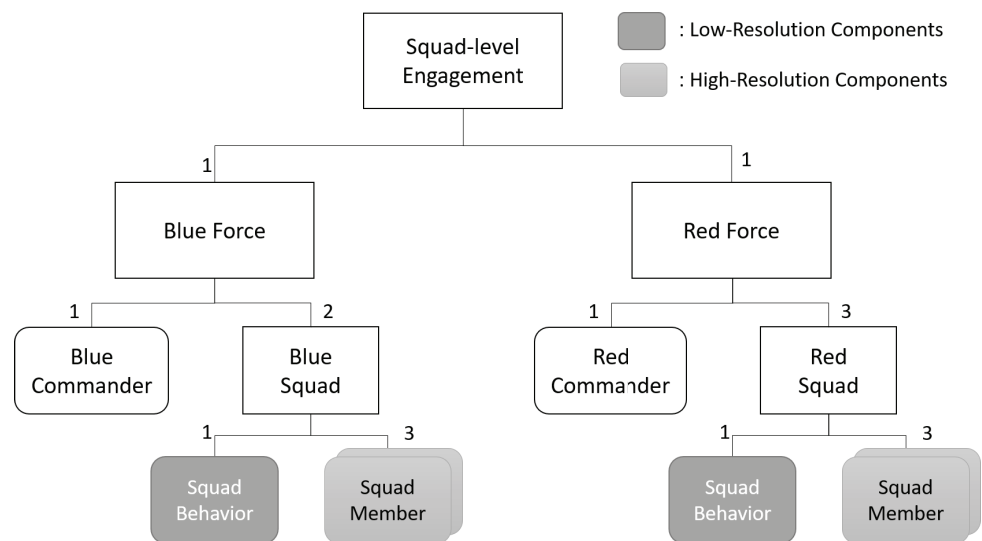


Figure 1. Model structure of a squad-level engagement model in the case study: blue and red squad models hold high- and low-resolution components, and their activation would be followed by the coupling relations of the resolution state $(\Sigma_r, r \in S_R)$.

Using the above example model, we designed a simulation scenario about an engagement between blue and red forces. The following is a brief illustration of the simulation scenario: (a) Three squads of red force and two squads of blue force are deployed to the north and south of a battlefield, respectively; (b) the squad models of the two forces approach each other, and they are at the squad level (i.e., low-resolution level, LR); (c) when they detect each other, each squad model turns into three squad member models, which are at the high-resolution level, HR, and they enter a firefight; (d) when the firefight ends, the remaining squad member models assemble into their squad models, and they continue to march as a low-resolution model. Figure 2 presents snapshots of the simulation's execution following the scenario, and they highlight two resolution changes in the squad models, which occur when each force recognizes its enemy (from low-resolution to high-resolution) and when they no longer detect an enemy after a gunfight (from HR to LR).

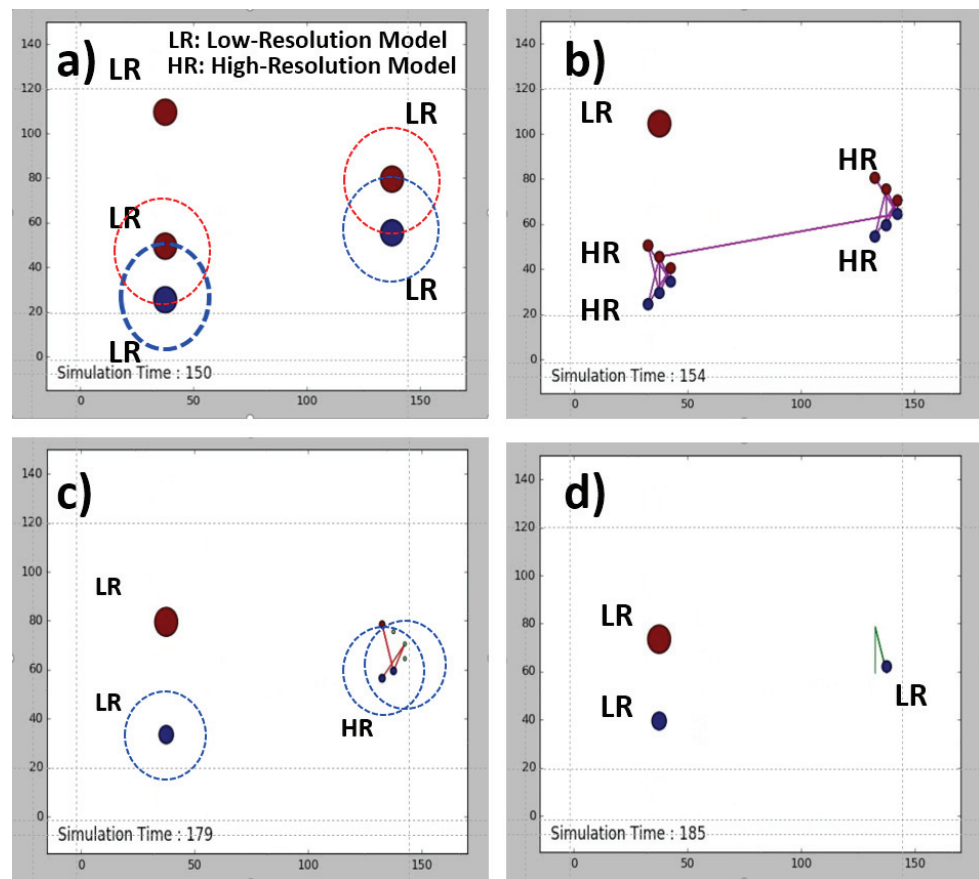
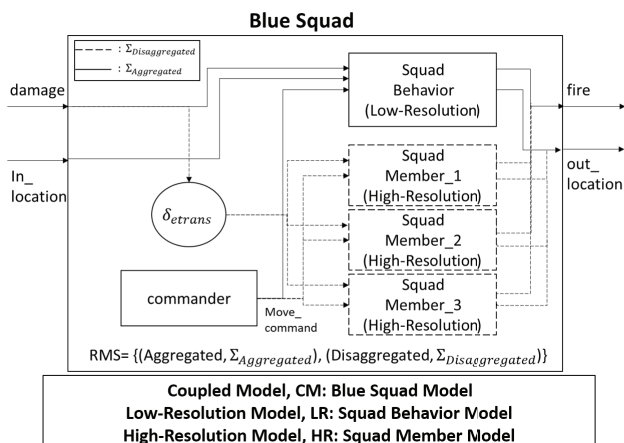


Figure 2. Screenshots of the simulation execution of the case study, particularly about resolution changes: (a,b) when an enemy has been detected (from low-resolution to high-resolution) and (c,d) when an engagement ends (from HR to LR).

4.2. Progress of Model Resolution Changes

Although we illustrated an example of MRM and the way MRM is specified via the proposed method, this subsection focuses on how resolution changes in MRM are realized through MRT-DEVS's semantics. In the above simulation scenario, the squad model changes its resolution level due to the detection of the enemy. For an improved understanding of this changing process, Figure 3 presents the structure of the blue squad model as an example. The blue squad model was developed based on the coupled model, and it has two resolution states (S_R): "Aggregated (as a low-resolution level)" and "Disaggregated (as a high-resolution level)". According to the resolution states, the RMS of the squad model was also specified (refer to the model diagram in Figure 3): (a) For the "Aggregated" state, the squad behavior model is set for activation, and its input and output events are connected with those of the squad models, which are specified in $\Sigma_{Aggregated}$. (b) For the "Disaggregated" state, three squad member models are set for activation, and their input and output events are connected with those of the squad models as well ($\Sigma_{Disaggregated}$). In particular, before being forwarded to squad members, an input event named "damage" with respect to the squad model is transformed by the event translation function (δ_{etrans}). Specifically, in the case study, the "damage" event of the squad model would be distributed into "damage" events of the three squad member models. We note that such event translations have relatively lower costs in the model development than previous methods that use a converter model (e.g., when taking an event, the converter model requires another simulation loop in the DEVS simulation algorithm [1]).



- Initialization of CM (Blue Squad model)
 - 1) RMS and initial state setting up
 - I. $CM.RMS = \{(Aggregated, \Sigma_{Aggregated}), (Disaggregated, \Sigma_{Disaggregated})\}$
 - II. $CM.S_R = Aggregated \rightarrow \Sigma_{Aggregated}$ is activated
- Resolution change flow (e.g., when detecting an enemy)
 - 1) Resolution state changes in LR (Squad Behavior model)
 - I. Model state changes by an external event
 - I. $LR.S_M(Detected) = LR.\delta_{ext}(LR.S_M(Move), LR.X(in_location))$
 - II. Resolution state change
 - I. $LR.S_R(Contact) = LR.\delta_{res}(LR.S_M(Detected), LR.S_R(Not_Contact))$
 - 2) RMS changes in CM
 - I. $CM.S_R(Disaggregated) = CM.\delta_{res}(CM.S_R(Aggregated), HR.S_R(Contact))$
 • $\rightarrow \Sigma_{Disaggregated}$ is activated
 - 3) State translation of HRs (Squad Member_1~3)
 - I. $HR.S(HP) = CM.\delta_{strans}(CM.S_R(Disaggregated), LR.S(HP))$
 - 4) Event translation from CM.X to HR.X
 - I. $LR.S(Damage) = CM.\delta_{etrans}(CM.S_R(Disaggregated), CM.X(Damage))$

Figure 3. Resolution-level changes in the blue squad model: when the squad model detects an enemy force, its resolution state changes from the “Aggregated” state with the low-resolution model (i.e., squad behavior model) to the “Disaggregated” state with a high-resolution model (i.e., squad member models).

Based on the above specifications of the squad model, we examine detailed procedures of its resolution changes in the case study. The right side of Figure 3 illustrates setting the initial resolution state and changing the resolution state of the squad model (coupled model, CM). Following the simulation scenario, the squad model conducts maneuver and scouting operations, so its initial resolution state is at a low-level resolution according to the model’s specifications. More specifically, the resolution state of the squad model (S_R) is set as “Aggregated”, and the associated RMS ($\Sigma_{Aggregated}$), which includes a squad behavior model (low-resolution component, LR), is also activated. While conducting operations, the squad model changes its resolution state when it detects an enemy squad. When the enemy is detected, the squad model is still at the low-resolution level: The detection of the enemy is determined by the external transition function (δ_{ext}) of the squad behavior model. After detecting an enemy, the squad behavior model changes its resolution state to “Contact”. When the component model changes its resolution state, the resolution transition function (δ_{res}) of its coupled model is also conducted (refer to line 26 of Algorithm 2). Hence, the squad model also changes its resolution state from “Aggregated” to “Disaggregated” for a gunfight against the detected enemy. After its resolution state changes, the associated RMS ($\Sigma_{Disaggregated}$), which includes three squad member models (high-resolution component, HR), is activated; otherwise, the RMS of “Aggregate” state ($\Sigma_{Aggregated}$) is deactivated. During resolution level changes, some states of the low resolution model need to be transferred to those of high-resolution models. For example, in the case study, the HP state of the squad’s behavior (LR) should be translated into the HP state of squad members (HR), and this translation is conducted by the state transition function (δ_{strans}) of the squad model (CM). After existing at high-resolution states, the two forces fall into an engagement. During the engagement, damage from the enemy would come to the squad model as its input event. This damage input is translated by the event translation function (δ_{etrans}) of the squad model before it reaches the squad members. The translation result represents, for example, the damage of each squad members, so a squad member that receives too much damage would die. After all squad members of one force have died, the remaining squad changes its resolution state from “Disaggregated” to “Aggregated” to carry on the maneuver and scout operations, and similar operations would follow.

5. Discussion

MRT-DEVS shares a theoretical background with the classic DEVS formalism, yet as we mentioned before, it relaxes a black box assumption of DEVS to achieve lower costs in MRMS. From various perspectives, including the relaxation, this section investigates strong

and weak points of the MRT-DEVS. Before diving into details, we analyze the functional analogy of MRT-DEVS, which was conducted by comparing the expressiveness power of MRT-DEVS with previous studies, such as Baohong's [11] and Hong's [33] studies.

Table 1 presents the relationships between the features of multi-resolution modeling (MRM) and the elements of the associated formalisms, including previous (Baohong's and Hong's) and the proposed one (MRT-DVEVS). Table 1 shows that the elements of MRT-DEVS have more matched elements compared to past MRM methods, which demonstrates two benefits of the proposed method: The first benefit, conservatively, is that the proposed MRT-DEVS is reducible to past MRM methods, which means that the proposed method is theoretically sufficient for MRM as previous ones provide; the second benefit is that both previous methods fail to match some parts of the MRM's features (e.g., the model state translation function in Hong's and the resolution change event structure in Baohong's), and the proposed method, however, covers it. As such, the proposed method has a larger coverage on MRM features, so we argue that from this point of view, the proposed method provides more applicabilities in MRMS.

Table 1. Relations between the features of multi-resolution modeling (MRM) and the elements from the associated formalisms (previous (Baohong's and Hong's) and proposed ones (MRT-DVEVS)).

MRM Features	Baohong's	Hong's	MRT-DEVS
Resolution State	γ	r	S_R
Resolution Model Family	M_k	RM_i	RMS
Model Structure Function/Info	M_ψ	ψ, ρ	δ_{res}
Model State Translation Function	π	-	δ_{strans}
Event Conversion Function	π	$MREI$	δ_{etrans}
Resolution Change Event Structure	-	C_R, Y_R	(Embedded in simulation algorithm)
Resolution Change Controller	χ	$MRCI$	(Embedded in simulation algorithm)

Let us address the details of the efficiency of the MRT-DEVS using the case study. We argue that the contribution of this research is in providing an option for MRM from a practical perspective, and this practicality comes from the ease in MRM development and lower costs for simulation executions. Having said that, we note that it is difficult, or even infeasible, to quantitatively compare the efficiency of the proposed method due to the following reasons: (1) The ease of the model's development is difficult to measure and even strongly dependent on modelers, which means that it cannot be appropriately used as a performance measure for the comparison; (2) to quantitatively compare the efficiency of the proposed method, comparison targets are required (i.e., the implementations of Baohong's and Hong's idea). However, to the best of our knowledge, their implementations are not available to the public.

As such, we rather provide another method to prove our outperformance by using an abstract comparison. Specifically, for the ease of model developments, we consider a practical problem in MRM development practices: In recent practices, MRM users should consider not only MRM for their target systems but also for handling resolution changes (e.g., as the resolution converter). This means that there is a high hurdle in MR's model development. However, at this point, the proposed method helps the modelers to focus only on modeling itself, setting aside additional MR considerations.

For the simulation's efficiency, we can analyze their simulation efficiencies at an abstract level and eventually compare their expected performance. The main feature of the proposed method is embedding state and event translation functions into the coupled model, so the resolution conversion process is performed within the simulation algorithm (Algorithm 2). Due to embedded MR functions, modelers eventually use less efforts in MR modeling. Specifically, compared to past methods that adapt these translation functions at another component (e.g., implementing a converter model), the proposed method permit

disregarding the details of the resolution's change. Developing the converter model may not require a heavy cost, but establishing and maintaining connections between the converter model with other models (e.g., responding to the resolution changes) definitely require heavy costs; the converter model should take a role for transferring all input/output events associated with the entire model's components during the simulation, so its connection structure becomes complex and requires expensive costs. More specifically, considering the method of DEVS simulation execution [1], such translations through a converter model are implemented by additional event handling and state transitions of a DEVS atomic model, which was identified as a critical factor for delaying DEVS simulation execution times in various studies [35–39]. In contrast, the proposed MRT-DEVSs permit event and state translations by functions in the coupled model. Such treatments require no extra event exchanges and, thus, generate no negative effects with respect to the simulation algorithm, which eventually lessens costs in both model development and simulation execution.

Having said that, we also admitted that the proposed method involves a potential shortcoming induced by removing the black-box assumption. The black box assumption, or modularity, enables the construction of a structured model by an integration of a number of small blocks (which are DEVS atomic and coupled models). Such a trait makes it easier to not only develop a complex model but also maintains the developed model, such as reuses in another model development [5]. Due to the loss of the modular property, the proposed method holds a limitation on the development of simulation models across various domain systems in which a number of model reuses can happen. Nonetheless, we still hold that it cannot induce much damage on MRMS. As we have argued throughout this paper, increasing the practicality of MRMS is the important motivation of this study. The importance of the model's reusability in modeling and simulation has been discussed in M&S communities [5,40], yet its practical examples over various domains are rarely observed. Still, we observe that there is a trade-off relationship between efficiency and model reuses in MRMS, and we note that the proposed method offers a practical option that is worthy of consideration for MRMS users.

6. Conclusions

Multi-resolution modeling and simulation (MRMS) is a useful option for gleaning insights with respect to target systems from various resolution levels. Many methods have been developed to support MRMS, but contrary to their theoretical completeness, their practical usages are rarely observed. We see that this distance is derived from inefficiencies, such as such indigestible model specifications and expensive simulation costs. The proposed MRT-DEVS tackles such inefficiencies by embedding state and event translation functions into the model's specification. The provided case study illustrates how the proposed method is applied to MRM and the detailed procedures of resolution changes via the suggested model specifications. Moreover, by engaging in discussions on the proposed method, we offered more considerations about MRMS to users. With all the provided information, we expect that MRMS users would consider the proposed method as a practical option to implement their models.

Author Contributions: Conceptualization, J.W.B. and I.-C.M.; methodology, I.-C.M.; software, I.-C.M.; validation, J.W.B.; formal analysis, J.W.B.; investigation, J.W.B.; resources, I.-C.M.; data curation, I.-C.M.; writing—original draft preparation, J.W.B. and I.-C.M.; writing—review and editing, J.W.B.; visualization, J.W.B.; supervision, I.-C.M.; project administration, J.W.B.; funding acquisition, J.W.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Agency For Defense Development by the Korean Government(UD210008DD).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Zeigler, B.P.; Praehofer, H.; Kim, T.G. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*; Academic Press: Cambridge, MA, USA, 2000.
2. Macal, C.M. Agent-based modeling and artificial life. In *Complex Social and Behavioral Systems: Game Theory and Agent-Based Models*; Springer: Berlin, Germany, 2020; pp. 725–745.
3. Rainey, L.B. Introduction and Overview for Engineering Emergence: A Modeling and Simulation Approach. In *Engineering Emergence*; CRC Press: Boca Raton, FL, USA, 2018; pp. 3–10.
4. Parry, H.R. Agent-based modeling, large-scale simulations. In *Complex Social and Behavioral Systems: Game Theory and Agent-Based Models*; Springer: Berlin, Germany, 2020; pp. 913–926.
5. Bae, J.W.; Moon, I.C. LDEF formalism for agent-based model development. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *46*, 793–808. [[CrossRef](#)]
6. Lee, K.; Lee, G.; Rabelo, L. A systematic review of the multi-resolution modeling (MRM) for integration of live, virtual, and constructive systems. *Information* **2020**, *11*, 480. [[CrossRef](#)]
7. Maffeo, C.; Aksimentiev, A. MrDNA: A multi-resolution model for predicting the structure and dynamics of DNA systems. *Nucleic Acids Res.* **2020**, *48*, 5135–5146. [[CrossRef](#)] [[PubMed](#)]
8. Khan, H.; Bazaz, M.A.; Nahvi, S.A. Adaptive multi-resolution framework for fast simulation of power electronic circuits. *IET Circuits Devices Syst.* **2020**, *14*, 537–546. [[CrossRef](#)]
9. Fishwick, P.A. The role of process abstraction in simulation. *IEEE Trans. Syst. Man Cybern.* **1988**, *18*, 18–39. [[CrossRef](#)]
10. Hong, S.Y.; Kim, T.G. A resolution converter for multi-resolution modeling/simulation on HLA/RTI. In *Systems Modeling and Simulation*; Springer: Berlin, Germany, 2007; pp. 289–293.
11. Baohong, L. A formal description specification for multi-resolution modeling based on DEVS formalism and its applications. *J. Def. Model. Simul.* **2007**, *4*, 229–251. [[CrossRef](#)]
12. Wei, L.; Wenjia, Z.; Heming, Z. Cross Level Switching Technology for Multi-resolution Model of Complex Products. *J. Syst. Simul.* **2022**, *34*, 944.
13. Wang, Y.; Qu, Q.; Yao, Y. Research on a Model Preaggregation Method for Multi-resolution Modeling. In Proceedings of the 2nd International Conference on Artificial Intelligence and Advanced Manufacture, Manchester, UK, 15–17 October 2020; pp. 155–158.
14. Schiller, M.; Dupius, M.; Krajzewicz, D.; Kern, A.; Knoll, A. Multi-resolution traffic simulation for large-scale high-fidelity evaluation of VANET applications. In *Simulating Urban Traffic Scenarios*; Springer: Berlin, Germany, 2019; pp. 17–36.
15. Liu, B.; Verbraeck, A. Multi-resolution modeling based on quotient space and DEVS. *Simul. Model. Pract. Theory* **2017**, *70*, 36–51. [[CrossRef](#)]
16. Moon, I.C.; Hong, J.H. Theoretic interplay between abstraction, resolution, and fidelity in model information. In Proceedings of the 2013 Winter Simulation Conference (WSC), Washington, DC, USA, 8–11 December 2013; pp. 1283–1291.
17. Mitchell, B.; Yilmaz, L. Symbiotic adaptive multisimulation: An autonomic simulation framework for real-time decision support under uncertainty. *ACM Trans. Model. Comput. Simul. (TOMACS)* **2009**, *19*, 1–31. [[CrossRef](#)]
18. Benjamin, P.; Erraguntla, M.; Delen, D.; Mayer, R. Simulation modeling at multiple levels of abstraction. In Proceedings of the 1998 Winter Simulation Conference, Proceedings (Cat. No. 98CH36274), Washington, DC, USA, 13–16 December 1998; Volume 1, pp. 391–398.
19. Davis, P.K.; Bigelow, J.H. *Experiments in Multiresolution Modeling (MRM)*; Technical Report; Rand Corp.: Santa Monica, CA, USA, 1998.
20. Davis, P.K.; Huber, R.K. *Variable-Resolution Combat Modeling: Motivations, Issues, and Principles*; Technical Report; Rand Corp.: Santa Monica, CA, USA, 1992.
21. Rauscher, S.A.; Ringler, T.D.; Skamarock, W.C.; Mirin, A.A. Exploring a global multiresolution modeling approach using aquaplanet simulations. *J. Clim.* **2013**, *26*, 2432–2452. [[CrossRef](#)]
22. Jeschke, M.; Uhrmacher, A.M. Multi-resolution spatial simulation for molecular crowding. In Proceedings of the 2008 Winter Simulation Conference, Miami, FL, USA, 7–10 December 2008; pp. 1384–1392.
23. Tan, G.; Ng, W.N.; Moradi, F. Aggregation/disaggregation in HLA multi-resolution distributed simulation. In Proceedings of the IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications, Cincinnati, OH, USA, 13–15 August 2001; p. 0076.
24. Zeigler, B.P.; Kim, D. Multi-resolution modeling for adaptive UAV service systems. In Proceedings of the 2019 Spring Simulation Conference (SpringSim), Tucson, AZ, USA, 29 April–2 May 2019; pp. 1–12.
25. Yoon, K.; Lee, W.; Croce, P.; Cammalleri, A.; Yoo, S.S. Multi-resolution simulation of focused ultrasound propagation through ovine skull from a single-element transducer. *Phys. Med. Biol.* **2018**, *63*, 105001. [[CrossRef](#)]
26. Ringler, T.; Petersen, M.; Higdon, R.L.; Jacobsen, D.; Jones, P.W.; Maltrud, M. A multi-resolution approach to global ocean modeling. *Ocean. Model.* **2013**, *69*, 211–232. [[CrossRef](#)]

27. Choi, S.H.; Lee, S.J.; Kim, T.G. Multi-fidelity modeling & simulation methodology for simulation speed up. In Proceedings of the 2nd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation, Denver, CO, USA, 18–21 May 2014; pp. 139–150.
28. Drewry, D.T.; Reynolds, P.F.; Emanuel, W.R. An optimization-based multi-resolution simulation methodology. In Proceedings of the Winter Simulation Conference, San Diego, CA, USA, 8–11 December 2002; Volume 1, pp. 467–475.
29. Natrajan, A.; Reynolds, P.F., Jr.; Srinivasan, S. MRE: A flexible approach to multi-resolution modeling. *ACM Sigsim Simul. Dig.* **1997**, *27*, 156–163. [[CrossRef](#)]
30. Reynolds, P.F., Jr.; Natrajan, A.; Srinivasan, S. Consistency maintenance in multiresolution simulation. *ACM Trans. Model. Comput. Simul. (TOMACS)* **1997**, *7*, 368–392. [[CrossRef](#)]
31. Gholami, S. Combined DEVS multiresolution simulation and model checking. In Proceedings of the 2017 Winter Simulation Conference (WSC), Las Vegas, NV, USA, 3–6 December 2017; pp. 4646–4647.
32. Goo, B.; Chung, H.; Han, S. Layered discrete event system specification for a ship production scheduling model. *Simul. Model. Pract. Theory* **2019**, *96*, 101934. [[CrossRef](#)]
33. Hong, S.Y.; Kim, T.G. Specification of multi-resolution modeling space for multi-resolution system simulation. *Simulation* **2013**, *89*, 28–40. [[CrossRef](#)]
34. Barros, F.J. Modeling formalisms for dynamic structure systems. *ACM Trans. Model. Comput. Simul. (TOMACS)* **1997**, *7*, 501–515. [[CrossRef](#)]
35. Lee, W.B.; Kim, T.G. Performance evaluation of concurrent system using formal model: Simulation speedup. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2003**, *86*, 2755–2766.
36. Bae, J.W.; Bae, S.W.; Moon, I.C.; Kim, T.G. Efficient flattening algorithm for hierarchical and dynamic structure discrete event models. *ACM Trans. Model. Comput. Simul. (TOMACS)* **2016**, *26*, 1–25. [[CrossRef](#)]
37. Muzy, A.; Nutaro, J.J. Algorithms for efficient implementations of the DEVS & DSDEVS abstract simulators. In Proceedings of the 1st Open International Conference on Modeling & Simulation (OICMS), Clermont-Ferrand, France, 12–15 June 2005; pp. 273–279.
38. Kwon, S.J.; Kim, T.G. Design and implementation of event-based DEVS execution environment for faster execution of iterative simulation. In Proceedings of the SpringSim (TMS-DEVS), Orlando, FL, USA, 26–29 March 2012; p. 14.
39. Fishwick, P.A. DEVS Formalism for Modeling of Discrete-Event Systems. In *Handbook of Dynamic System Modeling*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2007; pp. 107–120.
40. Robinson, S.; Nance, R.E.; Paul, R.J.; Pidd, M.; Taylor, S.J. Simulation model reuse: Definitions, benefits and obstacles. *Simul. Model. Pract. Theory* **2004**, *12*, 479–494. [[CrossRef](#)]

Article

Multi-Category Innovation and Encroachment Strategy Evolution of Composite E-Commerce Platform Based on Multi-Agent Simulation

Ziyan Wang¹ and Tianjian Yang^{1,2,*}

¹ School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing 100876, China

² School of Modern Post (School of Automation), Beijing University of Posts and Telecommunications, Beijing 100876, China

* Correspondence: frankytj@bupt.edu.cn

Abstract: An in-depth study of the product encroachment behavior on the composite e-commerce platform is of great significance to standardize the platform economy. This paper studies product encroachment behavior of composite e-commerce platforms with double-differentiated multi-product competition and constructs a game model of product innovation by an independent seller and product encroachment by the platform owner. Using multi-agent simulation, we simulate the bounded rational decision-making and interaction process of multiple agents in multiple periods and analyze the main parameters' influence. Results indicate the following: (1) In dual-differentiated multi-product competition, the third-party seller is more willing to invest in innovating high-quality category P, and the profit-driven platform owner only encroaches on the new variants of category P. (2) The larger consumers' platform owner preference can encourage the third-party seller to innovate high-quality new products. The increase in vertical differentiation of categories can enhance the third-party seller's innovation motivation for the traffic-attracting category. (3) A reasonable commission rate set by the platform owner can ensure the variety of variants of various categories, thereby expanding the sales scope of the composite e-commerce platform. Diseconomies of scale of category diversity management costs hinder the growth of product variety in the online marketplace.

Keywords: composite e-commerce platform; dual differentiated; product innovation; product encroachment; multi-agent simulation

Citation: Wang, Z.; Yang, T. Multi-Category Innovation and Encroachment Strategy Evolution of Composite E-Commerce Platform Based on Multi-Agent Simulation. *Systems* **2022**, *10*, 215. <https://doi.org/10.3390/systems10060215>

Academic Editors: Philippe Mathieu, Juan M. Corchado, Alfonso González-Briones and Fernando De la Prieta

Received: 5 October 2022

Accepted: 8 November 2022

Published: 11 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the context of the COVID-19 pandemic, online marketplaces are becoming more popular [1], and a series of issues related to the development of the e-commerce ecosystem are constantly emerging. At present, e-commerce platforms are increasingly not only trading places between customers and third-party sellers, but the platform owners also usually act as prevailing sellers in their own platforms. For example, JD, Dangdang.com, and Amazon are all composite e-commerce platforms [2]. Generally speaking, the types of products available on the composite e-commerce platform will be far more than that in the physical store. For example, there are more than 8000 digital cameras displayed and sold on Amazon, while a Walmart physical store can only display around 30 kinds of products. Excluding products with high sales volumes, most varieties are “long tail” products with relatively low sales volumes. For the platform owner, it may be uneconomical to sell varieties with a low sales volume; accordingly, for example, Amazon will leave up to 93% of categories to its independent third-party sellers for sale. However, with the rapid development of third-party sellers, these product categories that help third-party sellers achieve revenue will attract high attention from platform owners. In order to expand product categories and create higher revenue, platform owners will encroach on

the product space of third-party sellers, and procure and sell products directly. Statistics show that Amazon enters three percent of third-party sellers' product space over a ten-month period. Platform owners have an information advantage: they are actually closely observing category sales and perform category evaluation of third-party sellers. Profit drives them to encroach on the third-party sellers' successful categories with "blockbuster" sales. Platform owners also have advantages in product display. For example, Amazon exhibits "Similar Items to Consider" ads directly above an item's shopping cart link, thereby promoting its own products before consumers add third-party sellers' products to their shopping carts. Because of the dominant market position of the platform owner, this kind of category encroachment often damages the market position of the independent third-party sellers or even makes them exit the market. Therefore, understanding the category encroachment behavior of the platform owner and making strategic adjustments is often key to the survival of third-party sellers. In addition to responding to the encroachment by adjusting pricing [3,4] and marketing [5] strategies, third-party sellers should continue to bring innovative products, create a particularity that is distinguishable from other similar products to attract customers, and strive for the favorable position of market competition based on product differences. However, the platform owner is not at complete liberty to take any action he desires—generally speaking, the platform owner usually carefully balances the short-term profit encroachment affords them and the damage to product innovation incurred by independent sellers due to his excessive encroachment [6]. The latter usually compromises the diversity of products and thus the health of the entire platform.

In addition, with the improvement of economic level, consumers' pursuit of product quality and variety is increasing. In order to cater to more consumer segments, there is an increasing variety of products on the platform. Statistics show that from May 2015 to May 2016, taking shoes, clothing, and jewelry as an example, Amazon's self-operated product variety increased by 83%, and third-party sellers' product variety increased by 84%. At present, product vertical and horizontal differentiation strategies are often used as an important means for sellers to segment and expand the market. Vertical differentiation mainly refers to the difference in product quality, while horizontal differentiation refers to the difference in product color, size, taste, and other aspects. For example, the Philips portable battery has 10,000 mah and 20,000 mah battery capacities and comes in a variety of color variations, such as pure black and blue-black. From the core level of the product, the main goal of product innovation is to realize multiple differentiation through technological innovation and product serialization, which is also an effective way to cope with competition [7].

In view of this, this paper takes dual-differentiated multi-product competition as the starting point and constructs a multi-period game model of product innovation by an independent seller and product encroachment by the platform owner. Based on this, the optimal product innovation decision of the third-party seller and the optimal product encroachment decision of the platform owner are discussed. Furthermore, this paper combines analysis and multi-agent simulation. By simulating the heuristic process of some bounded rational decision-making of merchants, in addition to the analytical results, many emergent results can be produced [8–10]. Additionally, the influence of the main parameters on both players' decision-making and profit is analyzed. Specifically, this paper aims to answer the following questions: (1) What is the optimal horizontal differentiation innovation decision of the third-party seller for categories with different qualities in the face of possible product encroachment behavior of the platform owner? (2) What is the platform owner's optimal product encroachment decision for the third-party seller's innovative products? (3) What is the evolution law of the optimal decision-making of both players under the competitive interaction of multiple periods? (4) How do the main parameters affect the evolution results?

2. Literature Review

On a composite e-commerce platform, the platform owner is both an athlete and a referee and has a wealth of information; third-party sellers, meanwhile, enjoy unique category innovation, and operation and management capabilities. The key to the healthy development of e-commerce platforms depends on the perfect integration of multiple participants [11]. Parker and Van Alstyne [12] studied the motivation of platform owners to encroach on the product space of more successful third-party sellers, while Gawer and Henderson [13] found that platform owners may also worry about the deterioration of the health of the platform ecosystem and its long-term interests and choose to avoid direct competition with complementary third-party sellers, or only compete with those with service problems in order to maintain a good “fair” reputation without harming third-party sellers and richness of categories. A very representative recent study comes from Zhu and Liu [14]. They used empirical methods to systematically test Amazon’s category encroachment behavior against third-party sellers and explore the significance of influencing factors such as price, variety, commission fees, distribution costs, demand levels, customer reviews, and seller size. They found that third-party sellers should focus on less prominent products or on categories that require extensive platform-side investment to be successful, and suggested maintaining the ability to develop new products. Consistent with Zhu and Liu, Li et al. [15] found that with the development of the platform, due to the risk of developing new categories and the expectation to free ride on the platform owner’s best-selling products, third-party sellers that originally focused on “niche products” may also encroach on the platform owner’s product range. They used empirical methods to study the encroachment strategies of third-party sellers on platform products and found that third-party sellers will choose products that have a low price, high demand, low return rate, low operating cost, abundant supply sources, uniqueness, and high exposure. Moreover, encroachment by large third-party sellers will reduce the sales of the platform owner but increase the sales of the entire platform. The above representative works are all based on empirical or case analysis methods. Other studies on the category encroachment of composite e-commerce platforms have adopted the method of game analysis. There are relatively few such studies, which we discuss below.

The seed paper for analytical analysis comes from the research of Jiang et al. [6]. They examined product information disclosure to model Amazon’s product encroachment behavior against independent third-party sellers and constructed a two-period game model. Jiang et al. obtained the judgment conditions for “long-tail” and “short-tail” products, identified the conditions for achieving pool equilibrium and separation equilibrium in “middle-tail” products, and explained the internal relationship between Amazon’s product encroachment, product demand, and platform commission fees. Hagiue et al. [16] studied the competition–cooperation model of the coexistence of platform self-operated sales and third-party seller sales from the perspective of consumer surplus. They concluded that consumers can benefit from the platform’s dual role and pointed out that the platform’s self-interested purpose and category copycat behavior may bring about inefficiencies. Etro [17] modeled and compared the various sales models existing on the Amazon platform, namely the private label sales model, first party sales model, and third-party seller sales model. The conclusion shows that when third-party sellers have the characteristics of lower customer conversion rate, higher distribution cost, and lower market voice, or when the product has the characteristics of low value-added and high-demand elasticity, the platform owner tends to encroach on the market space of third-party sellers. In addition, Etro introduced the third-party seller’s product innovation and the platform’s category copycat behavior into the game model, and identified the third-party seller’s optimal innovation investment level and the platform’s optimal copycat probability. The above studies on product encroachment ideally assume that the third-party seller sells a single category or the categories sold are independent and unrelated, and when the platform side encroaches, the third-party seller immediately exits the market. None of them takes into account the fact that multiple products continue to coexist and differentiated competition occurs after the platform owner

encroaches. Feng et al. [18] discussed the behavior of third-party sellers encroaching on the platform owner's market share and assumed the coexistence and competition of multiple product categories after the encroachment. However, they only considered the vertical differentiation of categories, not the horizontal differentiation of categories. Moreover, they focused on the impact of two-way network externalities and did not combine the issue of category encroachment with that of category innovation.

With the increasing competition, product innovation and the introduction of differentiated products have become an important way for enterprises to cope with invasion challenges and gain competitive advantages. Wu and Lai [19] constructed a horizontal differentiation competition model and explored pricing and product launching strategies in a multistage game between two asymmetric firms. Yi and Chen [20] constructed a duopolistic competition game model consisting of a large manufacturer and a small manufacturer with imitation function and studied the product quality attributes decision-making of both manufacturers. Baron [21] studied the product positioning and innovation strategies of two competing firms under the coexistence of innovative products and initial products. He concluded that the incumbent firm would offer an additional product to forestall entry by narrowing the quality gap. Based on the following product encroachment and manufacturers' R&D modes (in-house R&D versus outsourcing R&D), Li et al. [22] constructed game models under oligopoly and oligopolistic competition, respectively, and discussed the influence of product encroachment on innovation quality. The above studies on product innovation strategies in the face of encroachment threat only study from one dimension of product horizontal and vertical differentiation and fail to consider the coexistence of the multiple differentiation of categories.

The direct source of the dual-differentiated product competition model in this paper is the research of Zhang et al. [23], who introduced the competition of dual-differentiated products (different product models exist and each model has multiple variants); however, they took horizontal and vertical differentiation as a given condition and considered the impact on information disclosure strategies of the intermediary and competitive sellers without paying attention to the product encroachment and innovation. In addition, many scholars have studied the differentiation strategy of dual-differentiated products. Shang-guan et al. [24] studied the two-dimensional product differentiation design and pricing strategies of a manufacturer. Jalali et al. [25] studied the optimal product development strategy (platform-based versus independent development) and the product differentiation strategy (horizontally versus vertically differentiated products) of a monopolistic manufacturer for quality and feature-sensitive customers, and emphasized the impact of operational cost parameters on the optimal differentiation strategy. Tian et al. [26] considered both the horizontal differentiation of channels and the vertical differentiation of products and analyzed the influence of consumer free-riding behavior on the optimal differentiation strategy. Lv [27] examined a two-dimensional differentiation model of both vertical product preferences and horizontal coupon preferences and investigated how couponing affects firms' promotion strategies and profits. Although these papers studied product differentiation decisions in different scenarios, they did not include product encroachment and product innovation under encroachment.

To highlight the contributions of this study, we contrast our study with other related works (as shown in Table 1). It can be concluded that this paper is different from existing literatures in the following three aspects: (1) Most analytical studies on the category encroachment of composite e-commerce platforms assume that third-party sellers withdraw from the market after the platform owner's category encroachment occurs, but this is not realistic. Moreover, only the assumption of a single category is made or only the vertical differentiation of categories is considered—the horizontal differentiation and the coexistence of multiple product variants are not considered. This paper considers two vertically differentiated categories with multiple variants, and multiple product variants continue to coexist after the platform owner's product encroachment. In addition, most of studies are based on single-stage and two-stage game analysis. This paper combines multi-agent

simulation to simulate and observe the emergent results of multi-period competition and evolution, which is a new manifestation of the computing-driven supply chain in category encroachment analysis, and provides new ideas for research on e-commerce platform ecosystem-related issues. (2) Most studies on category innovation strategies under the threat of encroachment only study from one dimension of product horizontal and vertical differentiation. From the above background, it can be seen that vertical and horizontal differentiation of categories are prominent in real life. Therefore, it is necessary to study product encroachment and innovation strategies considering the dual differentiation of categories. (3) By reviewing the literature in the field of category dual-differentiation, it is found that some scholars take category dual-differentiation as a given condition to study the impact on pricing [28], information disclosure, and other strategies, while others are concerned about products themselves and focus on the product differentiation strategy. However, there is no research that combines the competition of multiple differentiated products with product invasion and product innovation. This paper introduces the dual-differentiated multi-product competition into the encroachment problem and considers both the product innovation behavior of the independent seller and the product encroachment behavior of the platform owner. This paper not only analyzes the impact of category differentiation on both players' decision-making, but also studies the optimal horizontal differentiation strategy of both players for different quality categories, which has theoretical significance for regulating category encroachment behavior on composite e-commerce platforms. The research results of this paper have theoretical significance for regulating product encroachment behavior on composite e-commerce platforms.

Table 1. Comparative summary of related studies.

References	Category Encroachment	Category Innovation	Vertical Differentiation	Horizontal Differentiation	Multi-Period	Game Analysis	Multi-Agent
Gawer and Henderson [13]	✓						
Zhu and Liu [14]	✓						
Li et al. [15]	✓						
Parker and Van Alstyne [12]	✓					✓	
Jiang et al. [6]	✓					✓	
Hagiu et al. [16]	✓	✓				✓	
Etro [17]	✓	✓				✓	
Feng et al. [18]	✓		✓			✓	
Wu and Lai [19]	✓	✓		✓	✓	✓	
Yi and Chen [20]	✓	✓	✓			✓	
Baron [21]	✓	✓	✓			✓	
Li et al. [22]	✓	✓	✓			✓	
Zhang et al. [23]			✓	✓		✓	
Feng et al. [28]			✓	✓		✓	
Shangguan et al. [24]			✓	✓		✓	
Jalali et al. [25]		✓	✓	✓		✓	
Tian et al. [26]			✓	✓		✓	
Lv [27]			✓	✓		✓	
This study	✓	✓	✓	✓	✓	✓	✓

3. Model Formulation

3.1. Problem Description and Basic Assumptions

Consider a composite e-commerce platform that includes a platform owner (seller 1) and an independent third-party seller (seller 3). They can sell both of the two vertically differentiated categories at the same time: one is a high-quality profitable product (category P), and the other is a low-quality traffic-attracting product (category F). Each category has multiple variants, such as different colors or sizes, etc. The third-party seller can develop new variants to obtain horizontal differentiation advantages of categories, and the platform owner can increase revenue by copycat third-party innovative products.

This study can be regarded as examining a multi-period problem in which both types of sellers need to decide their retail prices and marketing efforts of the two categories in each period. In order to expand their market share, the third-party seller may invest in innovating category variants. The number of variants affects customer demand and incurs the cost of product diversification. In addition, the third-party seller understands that the platform owner may encroach when new variants are launched and needs to decide whether to invest in innovating products and how much to invest in the current period, and then puts new variants into the market in the next period. After observing the category innovation of the third-party seller, the platform owner decides whether to copycat new products in the same period. At the same time, each consumer has a potential demand for one unit of the product in each period.

In this paper, a multi-period game model is constructed, as shown in Figure 1. The game sequence is as follows: in period 0, the platform owner and the third-party seller aim to maximize their respective profits and decide their retail prices and marketing efforts of categories. After that, according to the prediction of the encroachment behavior of the platform owner, the third-party seller decides whether to invest in innovation and how many variants to innovate. Finally, consumers make their purchase in this period. In period 1, the product innovation of the third-party seller is declared a success or failure. At the same time, the platform owner chooses whether to encroach or not. Then, both players decide on category prices and marketing efforts at the same time. Subsequently, the third-party seller, based on the prediction of the platform owner’s encroachment behavior and the current category diversity situation, decides whether to continue to invest in innovation and how many variants to develop; finally, consumers make their purchase in this period. Next, the actions of period 1 are repeated for each period.

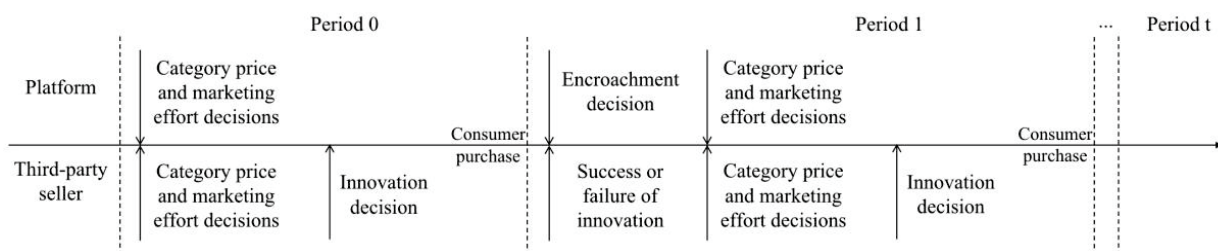


Figure 1. Decision sequence.

The assumptions of this paper are as follows: (1) Consumers’ willingness θ to pay for categories is heterogeneous. Let θ follow a uniform distribution in the interval $[0, \theta^+]$ and, without loss of generality, normalize $[0, \theta^+]$ to $[0, 1]$. (2) The category quality is an exogenous variable. The quality ratio of category P to category F is $\alpha q:q$ ($\alpha > 1$). α reflects the degree of vertical differentiation between the two categories. The larger the value of α is, the higher the consumer’s valuation of category P is and vice versa. (3) Consumers have a higher quality estimate δ ($\delta > 1$) of the platform’s self-operated products, but consumers’ valuation of the platform self-operated low-quality category F is lower than the valuation of the independent seller’s high-quality category P; that is, $\alpha > \delta$. (4) In order to promote the category and increase the willingness to pay of consumers, sellers implement a marketing effort level of $e_{ij}^{(t)} > 0$ and a resulting marketing cost of $\frac{1}{2}e_{ij}^{(t)2}$ ($i = 1, 3; j = F, P$) [5] in period t. (5) Both types of sellers face diseconomies of scale in the management of product diversity [29]. The management cost of each category is a quadratic function of the number of the category variants; therefore, the category management cost per period is $\frac{\beta}{2}n_{ij}^{(t)2}$ ($i = 1, 3; j = F, P$), where β ($\beta > 0$) is the diseconomies of scale coefficient, $n_{ij}^{(t)}$ represents the number of variants of seller i ’s category j in period t, and $n_{ij}^{(0)} = 0$. (6) The platform charges commission based on the sales of the third-party seller, and the commission rate is r ($0 < r < 1$).

Information such as product price, quality, marketing level, and seller type obtained by consumers through the platform search engine affects consumers' purchasing decisions [30]. In addition, considering that the horizontal differentiation of categories also has a positive impact on consumer demand, the utility that consumers obtain from purchasing category j ($j = F,P$) from seller i ($i = 1, 3$) in period t is:

$$U_{ij}^{(t)} = \begin{cases} \theta\delta q_j - p_{1j}^{(t)} + e_{1j}^{(t)} + n_{1j}^{(t)} & (i = 1) \\ \theta q_j - p_{3j}^{(t)} + e_{3j}^{(t)} + n_{3j}^{(t)} & (i = 3) \end{cases} \quad (1)$$

where q_j represents the quality of category j , and $p_{ij}^{(t)}$ represents the retail price of seller i 's category j in period t .

Consumers choose to buy a product that can afford them the maximum utility in each period, that is, $\max(U_{ij}^{(t)}, 0)$. The market segmentation of consumers is shown in Figure 2.

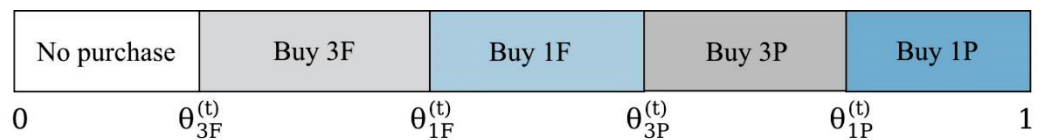


Figure 2. Consumer market segmentation.

Let $\theta_{3F}^{(t)}$, $\theta_{1F}^{(t)}$, $\theta_{3P}^{(t)}$, and $\theta_{1P}^{(t)}$ be the indifference points of consumer purchase utility in each period, and satisfy $0 \leq \theta_{3F}^{(t)} \leq \theta_{1F}^{(t)} \leq \theta_{3P}^{(t)} \leq \theta_{1P}^{(t)} \leq 1$. Among them, $\theta_{3F}^{(t)}$ represents the indifference threshold between consumers buying 3F-type and not buying; $\theta_{1F}^{(t)}$ represents the indifference threshold between consumers purchasing 1F-type and purchasing 3F-type; $\theta_{3P}^{(t)}$ represents the indifference threshold between consumers purchasing 3P-type and purchasing 1F-type; $\theta_{1P}^{(t)}$ represents the indifference threshold between consumers purchasing 1P-type and purchasing 3P-type. Based on this, the indifference point of the utility of consumers buying different categories from different sellers needs to satisfy the following:

$$\theta_{3F}^{(t)}q - p_{3F}^{(t)} + e_{3F}^{(t)} + n_{3F}^{(t)} = 0 \quad (2)$$

$$\theta_{1F}^{(t)}q - p_{3F}^{(t)} + e_{3F}^{(t)} + n_{3F}^{(t)} = \theta_{1F}^{(t)}\delta q - p_{1F}^{(t)} + e_{1F}^{(t)} + n_{1F}^{(t)} \quad (3)$$

$$\theta_{3P}^{(t)}\delta q - p_{1F}^{(t)} + e_{1F}^{(t)} + n_{1F}^{(t)} = \theta_{3P}^{(t)}\alpha q - p_{3P}^{(t)} + e_{3P}^{(t)} + n_{3P}^{(t)} \quad (4)$$

$$\theta_{1P}^{(t)}\alpha q - p_{3P}^{(t)} + e_{3P}^{(t)} + n_{3P}^{(t)} = \theta_{1P}^{(t)}\delta\alpha q - p_{1P}^{(t)} + e_{1P}^{(t)} + n_{1P}^{(t)} \quad (5)$$

From the above, the following can be solved:

$$\theta_{3F}^{(t)} = \frac{p_{3F}^{(t)} - e_{3F}^{(t)} - n_{3F}^{(t)}}{q} \quad (6)$$

$$\theta_{1F}^{(t)} = \frac{p_{1F}^{(t)} - p_{3F}^{(t)} - e_{1F}^{(t)} + e_{3F}^{(t)} - n_{1F}^{(t)} + n_{3F}^{(t)}}{q(-1 + \delta)} \quad (7)$$

$$\theta_{3P}^{(t)} = \frac{p_{3P}^{(t)} - p_{1F}^{(t)} - e_{3P}^{(t)} + e_{1F}^{(t)} - n_{3P}^{(t)} + n_{1F}^{(t)}}{q(\alpha - \delta)} \quad (8)$$

$$\theta_{1P}^{(t)} = \frac{p_{1P}^{(t)} - p_{3P}^{(t)} - e_{1P}^{(t)} + e_{3P}^{(t)} - n_{1P}^{(t)} + n_{3P}^{(t)}}{\alpha q(-1 + \delta)} \quad (9)$$

Therefore, the profit functions of the third-party seller and the platform owner in period t are:

$$\pi_3^{(t)} = (1 - r)((\theta_{1F}^{(t)} - \theta_{3F}^{(t)})p_{3F}^{(t)} + (\theta_{1P}^{(t)} - \theta_{3P}^{(t)})p_{3P}^{(t)}) - \sum_{j=F,P} (\frac{e_{3j}^{(t)2}}{2} + \frac{\beta}{2}n_{3j}^{(t)2}) \tag{10}$$

$$\pi_1^{(t)} = r((\theta_{1F}^{(t)} - \theta_{3F}^{(t)})p_{3F}^{(t)} + (\theta_{1P}^{(t)} - \theta_{3P}^{(t)})p_{3P}^{(t)}) + (\theta_{3P}^{(t)} - \theta_{1F}^{(t)})p_{1F}^{(t)} + (1 - \theta_{1P}^{(t)})p_{1P}^{(t)} - \sum_{j=F,P} (\frac{e_{1j}^{(t)2}}{2} + \frac{\beta}{2}n_{1j}^{(t)2}) \tag{11}$$

3.2. Multi-Category Innovation and Encroachment Decision Analysis

Assuming that the innovation success probability ρ increases with the increase in the innovation investment of the third-party seller, and the innovation investment amount is a convex function of the innovation success probability [17], thus the innovation investment of the third-party seller in period t is:

$$I^{(t)} = \frac{1 + \gamma}{1 + \sigma} \rho^{(t+1)1+\sigma} \tag{12}$$

where γ represents the marginal cost of innovation investment and σ represents the sensitivity of product innovation success to investment.

When the third-party seller’s product innovation is successful in period $t + 1$ and the platform owner encroaches, that means $n_{3F}^{(t)}$ and $n_{3P}^{(t)}$ will increase by $\Delta n_F^{(t+1)'}$ and $\Delta n_P^{(t+1)'}$, respectively, and $n_{1F}^{(t)}$ and $n_{1P}^{(t)}$ will do the same. Thus, the demand of each category of each seller is as follows:

$$D_{3F}^{(t+1)'} = \frac{p_{1F}^{(t+1)'} - p_{3F}^{(t+1)'} - e_{1F}^{(t+1)'} + e_{3F}^{(t+1)'} - n_{1F}^{(t)} + n_{3F}^{(t)}}{q(-1 + \delta)} - \frac{p_{3F}^{(t+1)'} - e_{3F}^{(t+1)'} - n_{3F}^{(t)} - \Delta n_F^{(t+1)'}}{q} \tag{13}$$

$$D_{3P}^{(t+1)'} = \frac{p_{1P}^{(t+1)'} - p_{3P}^{(t+1)'} - e_{1P}^{(t+1)'} + e_{3P}^{(t+1)'} - n_{1P}^{(t)} + n_{3P}^{(t)}}{q\alpha(-1 + \delta)} - \frac{-p_{1F}^{(t+1)'} + p_{3P}^{(t+1)'} - e_{3P}^{(t+1)'} + e_{1F}^{(t+1)'} + n_{1F}^{(t)} + \Delta n_F^{(t+1)'} - n_{3P}^{(t)} - \Delta n_P^{(t+1)'}}{q(\alpha - \delta)} \tag{14}$$

$$D_{1F}^{(t+1)'} = \frac{-p_{1F}^{(t+1)'} + p_{3P}^{(t+1)'} - e_{3P}^{(t+1)'} + e_{1F}^{(t+1)'} + n_{1F}^{(t)} + \Delta n_F^{(t+1)'} - n_{3P}^{(t)} - \Delta n_P^{(t+1)'}}{q(\alpha - \delta)} - \frac{p_{1F}^{(t+1)'} - p_{3F}^{(t+1)'} - e_{1F}^{(t+1)'} + e_{3F}^{(t+1)'} - n_{1F}^{(t)} + n_{3F}^{(t)}}{q(-1 + \delta)} \tag{15}$$

$$D_{1P}^{(t+1)'} = 1 - \frac{p_{1P}^{(t+1)'} - p_{3P}^{(t+1)'} - e_{1P}^{(t+1)'} + e_{3P}^{(t+1)'} - n_{1P}^{(t)} + n_{3P}^{(t)}}{q\alpha(-1 + \delta)} \tag{16}$$

where $p_{ij}^{(t+1)'}$ and $e_{ij}^{(t+1)'}$ represent the retail price and the marketing effort of ij -type when the platform owner encroaches in period $t + 1$, respectively.

At this time, the optimization problem of the third-party seller and the platform owner selling two categories is:

$$\begin{aligned} \max_{p_{3j}^{(t+1)'}, e_{3j}^{(t+1)'}, \Delta n_j^{(t+1)'}} \pi_{3(I,E)}^{(t+1)} &= \sum_{j=F,P} ((1 - r)(D_{3j}^{(t+1)'} p_{3j}^{(t+1)'}) - \frac{e_{3j}^{(t+1)2}}{2} - \frac{\beta}{2}(n_{3j}^{(t)} + \Delta n_j^{(t+1)'})^2) \\ \text{s.t. } e_{3j}^{(t+1)'} &\geq 0, \Delta n_j^{(t+1)'} \geq 0 \end{aligned} \tag{17}$$

$$\begin{aligned} \max_{p_{ij}^{(t+1)'}, e_{ij}^{(t+1)'}} \pi_{1(L,E)}^{(t+1)} &= \sum_{j=F,P} (rD_{3j}^{(t+1)'} p_{3j}^{(t+1)'} + D_{1j}^{(t+1)'} p_{1j}^{(t+1)'} - \frac{e_{ij}^{(t+1)'2}}{2} - \frac{\beta}{2} (n_{ij}^{(t)} + \Delta n_j^{(t+1)'})^2) \\ \text{s.t. } e_{ij}^{(t+1)'} &\geq 0 \end{aligned} \tag{18}$$

where (I,E) represents the situation in which the third-party seller’s product innovation is successful and the platform owner encroaches.

The innovation marginal profit of each seller is:

$$\Delta\pi_{i(L,E)}^{(t+1)} = \pi_{i(L,E)}^{(t+1)} - \pi_i^{(t)} \tag{19}$$

When the third-party seller’s product innovation is successful in period t + 1 and the platform owner does not encroach, that means $n_{3F}^{(t)}$ and $n_{3P}^{(t)}$ will increase by $\Delta n_F^{(t+1)'}$ and $\Delta n_P^{(t+1)'}$, respectively, and $n_{1F}^{(t)}$ and $n_{1P}^{(t)}$ will keep unchanged. Thus, the demand of each category of each seller is as follows:

$$D_{3F}^{(t+1)''} = \frac{p_{1F}^{(t+1)''} - p_{3F}^{(t+1)''} - e_{1F}^{(t+1)''} + e_{3F}^{(t+1)''} - n_{1F}^{(t)} + n_{3F}^{(t)} + \Delta n_{3F}^{(t+1)''}}{q(-1 + \delta)} - \frac{p_{3F}^{(t+1)''} - e_{3F}^{(t+1)''} - n_{3F}^{(t)} - \Delta n_{3F}^{(t+1)''}}{q} \tag{20}$$

$$D_{3P}^{(t+1)''} = \frac{p_{1P}^{(t+1)''} - p_{3P}^{(t+1)''} - e_{1P}^{(t+1)''} + e_{3P}^{(t+1)''} - n_{1P}^{(t)} + n_{3P}^{(t)} + \Delta n_{3P}^{(t+1)''}}{q\alpha(-1 + \delta)} - \frac{-p_{1F}^{(t+1)''} + p_{3P}^{(t+1)''} - e_{3P}^{(t+1)''} + e_{1F}^{(t+1)''} + n_{1F}^{(t)} - n_{3P}^{(t)} - \Delta n_{3P}^{(t+1)''}}{q(\alpha - \delta)} \tag{21}$$

$$D_{1F}^{(t+1)''} = \frac{-p_{1F}^{(t+1)''} + p_{3P}^{(t+1)''} - e_{3P}^{(t+1)''} + e_{1F}^{(t+1)''} + n_{1F}^{(t)} - n_{3P}^{(t)} - \Delta n_{3P}^{(t+1)''}}{q(\alpha - \delta)} - \frac{p_{1F}^{(t+1)''} - p_{3F}^{(t+1)''} - e_{1F}^{(t+1)''} + e_{3F}^{(t+1)''} - n_{1F}^{(t)} + n_{3F}^{(t)} + \Delta n_{3F}^{(t+1)''}}{q(-1 + \delta)} \tag{22}$$

$$D_{1P}^{(t+1)''} = 1 - \frac{p_{1P}^{(t+1)''} - p_{3P}^{(t+1)''} - e_{1P}^{(t+1)''} + e_{3P}^{(t+1)''} - n_{1P}^{(t)} + n_{3P}^{(t)} + \Delta n_{3P}^{(t+1)''}}{q\alpha(-1 + \delta)} \tag{23}$$

where $p_{ij}^{(t+1)''}$ and $e_{ij}^{(t+1)''}$ represent the retail price and the marketing effort of ij-type when the platform owner does not encroach in period t + 1, respectively.

At this time, the optimization problem of the third-party seller and the platform owner selling two categories is:

$$\begin{aligned} \max_{p_{3j}^{(t+1)'}, e_{3j}^{(t+1)'}, \Delta n_{3j}^{(t+1)'}} \pi_{3(L,N)}^{(t+1)} &= \sum_{j=F,P} ((1 - r)(D_{3j}^{(t+1)''} p_{3j}^{(t+1)'}) - \frac{e_{3j}^{(t+1)'2}}{2} - \frac{\beta}{2} (n_{3j}^{(t)} + \Delta n_j^{(t+1)'})^2) \\ \text{s.t. } e_{3j}^{(t+1)'} &\geq 0, \Delta n_{3j}^{(t+1)'} \geq 0 \end{aligned} \tag{24}$$

$$\begin{aligned} \max_{p_{1j}^{(t+1)'}, e_{1j}^{(t+1)'}} \pi_{1(L,N)}^{(t+1)} &= \sum_{j=F,P} (rD_{3j}^{(t+1)''} p_{3j}^{(t+1)'} + D_{1j}^{(t+1)''} p_{1j}^{(t+1)'} - \frac{e_{1j}^{(t+1)'2}}{2} - \frac{\beta}{2} n_{1j}^{(t)2}) \\ \text{s.t. } e_{1j}^{(t+1)'} &\geq 0 \end{aligned} \tag{25}$$

where (I,N) represents the situation in which the third-party seller’s production innovation is successful and the platform owner does not encroach.

The innovation marginal profit of each seller is:

$$\Delta\pi_{i(L,N)}^{(t+1)} = \pi_{i(L,N)}^{(t+1)} - \pi_i^{(t)} \tag{26}$$

Therefore, the expected marginal profit of product innovation of the third-party seller in period t + 1 is:

$$E(\Delta\pi_3^{(t+1)}) = \rho^{(t+1)} \left[f^{(t+1)} \Delta\pi_{3(L,E)}^{(t+1)} + (1 - f^{(t+1)}) \Delta\pi_{3(L,N)}^{(t+1)} \right] - \frac{1 + \gamma}{1 + \sigma} \rho^{(t+1)1 + \sigma} \tag{27}$$

where $f^{(t+1)}$ represents the product encroachment probability of the platform owner in period $t + 1$.

The expected marginal profit of product innovation of the third-party seller is equal to the probability of success of innovation multiplied by the weighted average of the marginal profits of innovation under the condition of platform encroachment and non-encroachment, minus the innovation investment amount.

In order to obtain the maximum expected marginal profit for the third-party seller, let $\frac{\partial E(\Delta\pi_3^{(t+1)})}{\partial \rho^{(t+1)}} = 0$, and the optimal success probability of innovation in period $t + 1$ is:

$$\rho^{(t+1)} = \min\left\{\max\left\{\left[\frac{f^{(t+1)}\Delta\pi_{3(I,E)}^{(t+1)} + (1 - f^{(t+1)})\Delta\pi_{3(I,N)}^{(t+1)}}{1 + \gamma}\right]^{\frac{1}{\sigma}}, 0\right\}, 1\right\} \in [0, 1] \quad (28)$$

The expected profit change brought about by the successful product innovation for the platform owner in period $t + 1$ is:

$$E(\Delta\pi_1^{(t+1)}) = \rho^{(t+1)} \left[f^{(t+1)}\Delta\pi_{1(I,E)}^{(t+1)} + (1 - f^{(t+1)})\Delta\pi_{1(I,N)}^{(t+1)} \right] \quad (29)$$

The expected profit change of the platform owner is equal to the probability of success of innovation multiplied by the weighted average of profit changes under encroachment and non-encroachment conditions. Let $\frac{\partial E(\Delta\pi_1^{(t+1)})}{\partial f^{(t+1)}} = 0$, and the optimal encroachment probability of platform owner in period $t + 1$ is:

$$f^{(t+1)} = \min\left\{\max\left[\frac{\Delta\pi_{1(I,N)}^{(t+1)}}{(1 + \sigma)(\Delta\pi_{1(I,N)}^{(t+1)} - \Delta\pi_{1(I,E)}^{(t+1)})} + \frac{\sigma\Delta\pi_{3(I,N)}^{(t+1)}}{(1 + \sigma)(\Delta\pi_{3(I,N)}^{(t+1)} - \Delta\pi_{3(I,E)}^{(t+1)})}, 0\right], 1\right\} \in [0, 1] \quad (30)$$

Therefore, the number of variants for each category of the third-party seller and the platform owner in period $t + 1$ are the cumulation of increments for each period:

$$n_{3j}^{(t+1)} = \sum_{t=0}^{t+1} \rho^{(t+1)} (f^{(t+1)}\Delta n_j^{(t+1)'} + (1 - f^{(t+1)})\Delta n_j^{(t+1)''}) \quad (31)$$

$$n_{1j}^{(t+1)} = \sum_{t=0}^{t+1} \rho^{(t+1)} f^{(t+1)} \Delta n_j^{(t+1)'} \quad (32)$$

3.3. Multi-Agent Simulation Model Establishment

Based on the above strategy, this paper establishes a multi-agent simulation model based on a genetic algorithm (GA) and observes the emergent results of multiple periods of competition and evolution by simulating the heuristic process of some bounded rational decision-making of merchants. As a parallel algorithm, GA has been used for seeking the global optimum and widely applied to solve the game equilibrium solution [8,9]. The complex constraints and objective functions are only used to check the feasibility and quality of the GA solution. In view of the short-term decision-making with the goal of maximizing the respective profits of both sellers in this paper, it can be regarded as a dual-objective optimization problem. Therefore, this paper uses GA to determine the optimal category retail price and marketing efforts in each period and nests GA into the game model of third-party seller category innovation and platform category encroachment to solve the optimal pricing, marketing, and innovation decisions under different encroachment situations of the platform.

This paper uses multi-agent simulation to dynamically simulate the decision-making and interaction process of each agent in multiple periods. The multi-period, multi-product

innovation and encroachment decision-making process of a composite e-commerce platform ecosystem is shown in Figure 3. The process can be expressed as the following steps:

Step 0: Enter the number of variants for each category of each seller in the initial state, $n_{ij}^{(0)} = 0$ ($i = 1, 3; j = F, P$).

Step 1: Based on Equations (6)–(9), update the demand function of each category of each seller.

Step 2: Use GA to determine $p_{ij}^{(t)}$ and $e_{ij}^{(t)}$ according to the following sub-steps.

Step 2-1: Initialize the population, set the variable range, and generate individual genes according to the variable range.

Step 2-2: Determine the fitness function [31] and calculate the fitness of each individual.

Step 2-3: Use the roulette wheel method to select the parents. Select excellent individuals with large fitness values for chromosome cross-combination and mutation.

Step 2-4: Repeat Step 2-2 to Step 2-3 until the number of iterations is reached, then jump out of the loop.

Step 2-5: Select the individual with the largest fitness as the optimal solution.

Step 3: Calculate $f^{(t+1)}$ and $\rho^{(t+1)}$ according to the following sub-steps.

Step 3-1: Based on Equations (13)–(16) and (20)–(23), update the demand function in the case of platform encroachment and non-encroachment.

Step 3-2: Use GA to determine the optimal strategies under different encroachment situations. Use Equations (19) and (26) to calculate $\Delta\pi_{i(L,E)}^{(t+1)}$ and $\Delta\pi_{i(L,N)}^{(t+1)}$.

Step 3-3: Use Equation (30) to calculate the platform owner’s optimal encroachment probability of innovative products in period $t + 1$.

Step 3-4: Use Equation (28) to calculate the optimal probability of successful innovation of the third-party seller in period $t + 1$.

Step 4: Determine whether the probability of successful innovation in period $t + 1$ is equal to 0. If $\rho^{(t+1)} = 0$, it means that the third-party seller fails to innovate and no longer invests in innovation; if $\rho^{(t+1)} > 0$, it means that the third-party seller can still successfully innovate new products, and it does not achieve an equilibrium yet.

Step 5: Based on Equations (31) and (32), update the number of product variants of each category of each party in period $t + 1$ and use it as the input for the next cycle. If Step 5 determines that the third-party seller’s product innovation success probability decreases to 0, the iteration ends.

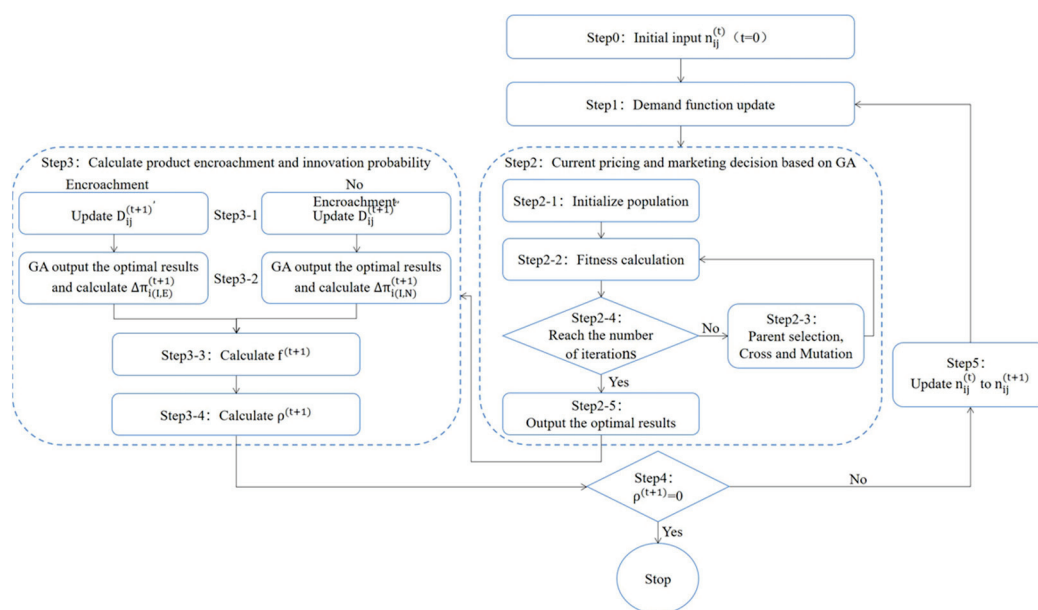


Figure 3. Schematic diagram of multi-period, multi-product innovation-and-encroachment decision-making process of composite e-commerce platform ecosystem.

4. Model Simulation and Analysis

This paper conducts multi-agent simulation experiments on Anylogic 8.7.5 software (Software Source: Russian XJ Technologic) to explore the evolution law of the optimal decision-making of the composite e-commerce platform system and analyzes the influence of category vertical differentiation, consumer channel preference, scale diseconomies, and platform commission rate on equilibrium decision-making. The model parameters and variables are configured as shown in Table 2.

Table 2. Model parameters and variable value settings.

PARAM, VAR	Meaning	Value	Assignment Rules
θ^+	Scope of consumers' quality assessment of categories	100	Constant
q_F	Quality of Category F	1000	Constant
α	Category vertical differentiation	2.0, 2.1, 2.2, etc.	Constant, set according to the experimental situation
δ	Platform owner operated channel preference of customers	1.5, 1.6, 1.7, etc.	Constant, set according to the experimental situation
β	Diseconomies of scale in product management costs	0.8, 1.0, 1.2, etc.	Constant, set according to the experimental situation
r	The commission rate charged by the platform	0.4, 0.5, 0.6, etc.	Constant, set according to the experimental situation
σ	Sensitivity of product innovation success to investment	0.3	Constant
γ	Marginal cost of product innovation investment	70	Constant
$p_{ij}^{(t)}$	Retail price of category j of seller i	[0, 1500]	Automatically adjust settings from GA training results
$e_{ij}^{(t)}$	Marketing efforts of category j by seller i	[0, 100]	Automatically adjust settings from GA training results
$n_{ij}^{(t)}$	The number of variants of seller i's category j	Initial value $n_{ij}^{(0)} = 0$	Automatically adjust settings from GA training results

4.1. Changes in Multi-Period Equilibrium Decision

When $t = 1$, the third-party seller develops new products successfully, and the platform owner implements the encroachment strategy. After that, the changes in the equilibrium decision of both sellers are shown in Figures 4–7.

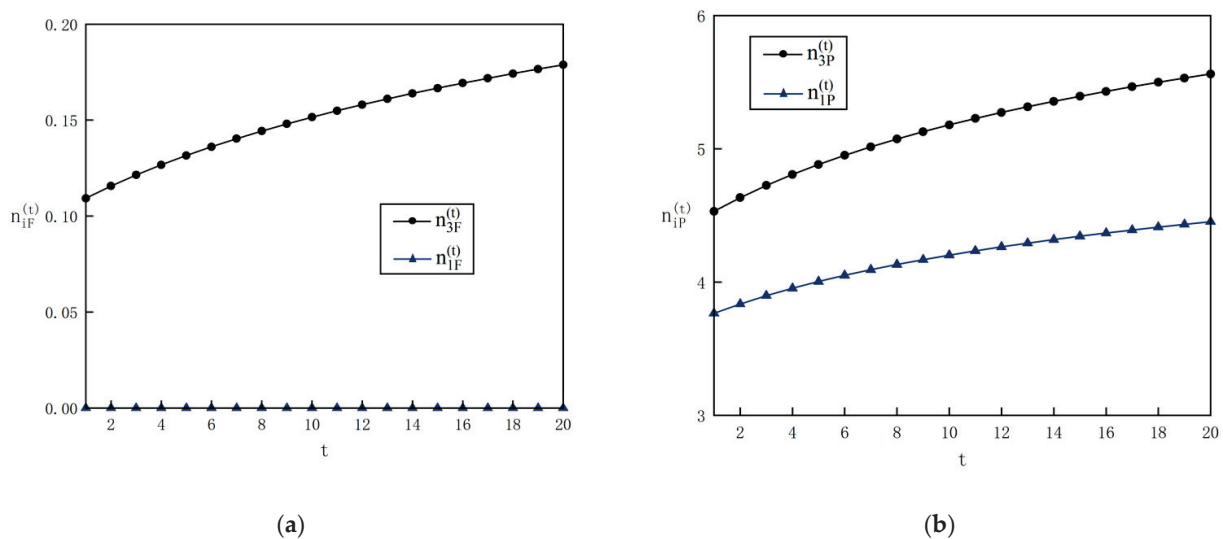
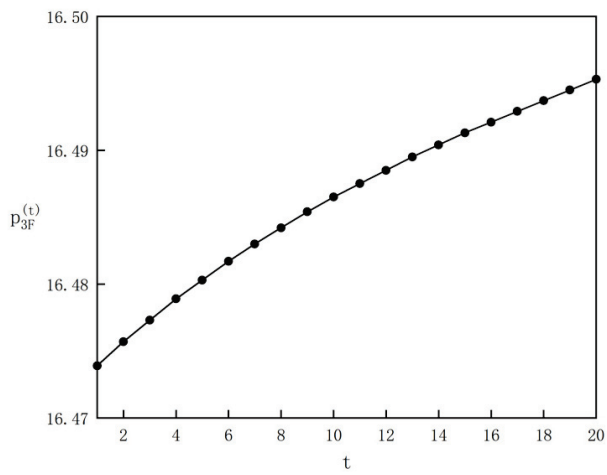
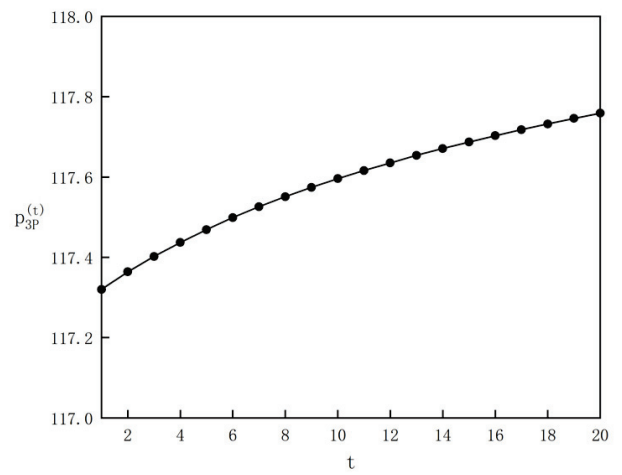


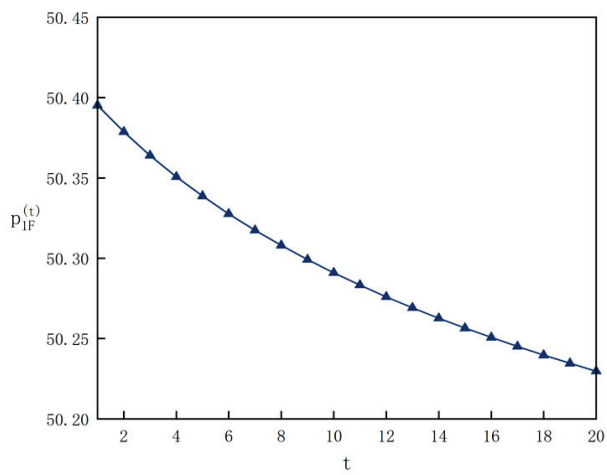
Figure 4. Number of sub-variants of each category of each seller during multiple periods (a) number of sub-variants of category F of each seller during multiple periods and (b) number of sub-variants of category P of each seller during multiple periods.



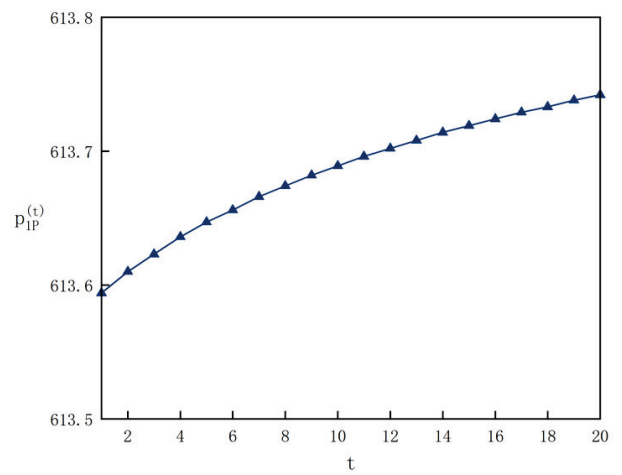
(a)



(b)

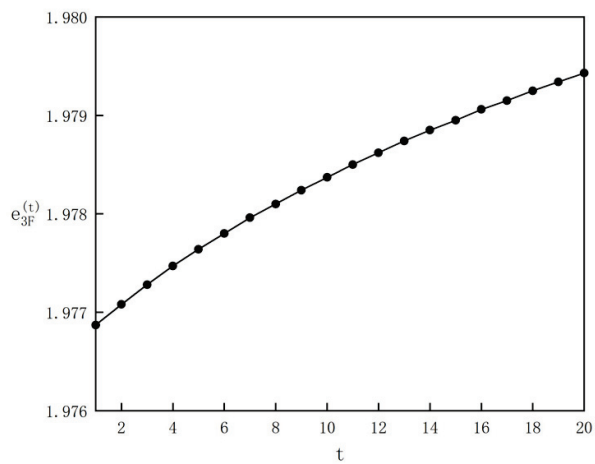


(c)

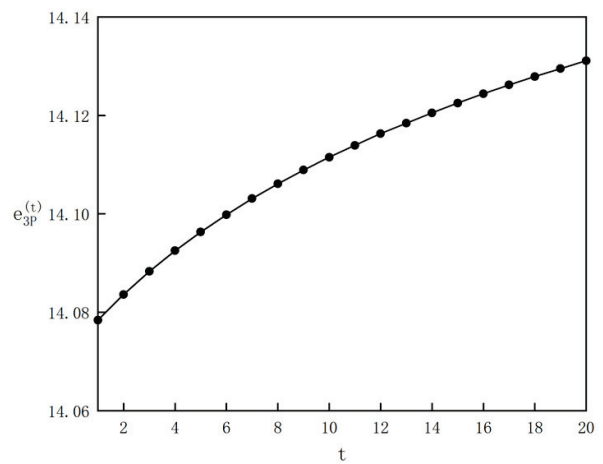


(d)

Figure 5. Retail price for each category of each seller during multiple periods: (a) retail price for 3F-type during multiple periods, (b) retail price for 3P-type during multiple periods, (c) retail price for 1F-type during multiple periods, and (d) retail price for 1P-type during multiple periods.

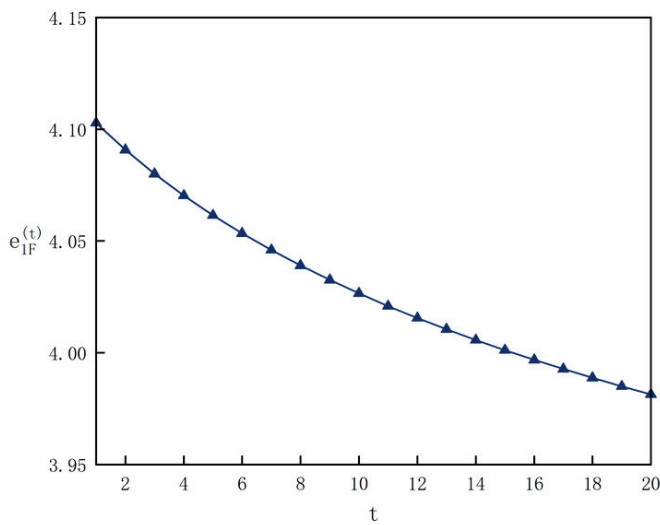


(a)

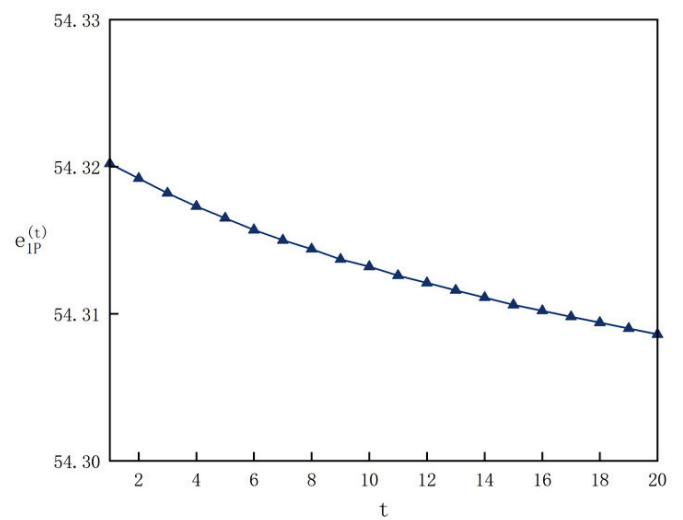


(b)

Figure 6. Cont.

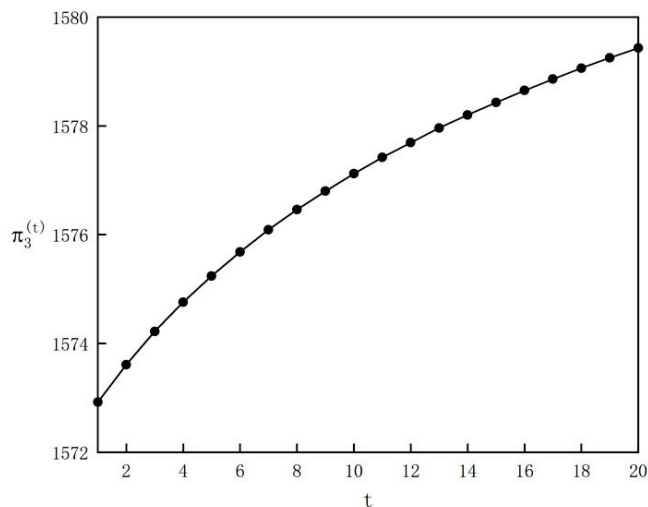


(c)

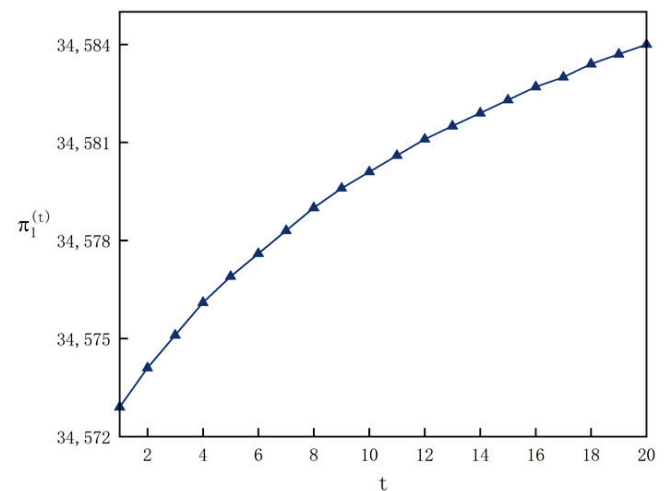


(d)

Figure 6. Marketing effort level of each category by each seller during multiple periods: (a) marketing effort level of 3F-type during multiple periods; (b) marketing effort level of 3P-type during multiple periods; (c) marketing effort level of 1F-type during multiple periods; and (d) marketing effort level of 1P-type during multiple periods.



(a)



(b)

Figure 7. Profit of each seller during multiple periods: (a) profit of the third-party seller during multiple periods and (b) profit of the platform owner during multiple periods.

It can be seen from Figure 4 that the number of innovated variants of 3P-type is much higher than that of 3F-type, and the platform owner only encroaches on category P, not category F. Figure 4 shows that the independent seller will innovate variants for both categories at the same time, but the platform owner almost only copycats high-end category variants. The independent seller consistently develops and maintains more product variants to stay competitive and strives to survive in the category P market.

It can be seen from Figure 5 that the prices of both categories of the third-party seller increase; the price of the platform owner’s category F decreases, while the price of the platform owner’s category P increases. As the third-party seller continues to innovate category variants, the retail price of his products will increase. The price of 1P-type that

the platform owner constantly copycats will also increase, but the price of 1F-type that the platform owner chooses not to copycat will decrease as the competitiveness of the independent seller increases.

Figure 6 shows that with the passage of time, compared with the platform owner, the third-party seller has the motivation to improve marketing efforts. Although the marketing effort level of the platform owner will gradually decrease while the marketing effort level of the independent seller gradually increases, since the independent seller has more products to sell, the marketing effort level will actually be lower than that of the platform owner; on the contrary, the platform owner needs to maintain a higher marketing effort level because of fewer product variants.

It can be seen from Figure 7 that the profits of both the third-party seller and the platform owner increase. Although the platform owner may encroach on the innovative products, the increase in product diversity meets more consumer demand, and there are more innovative products in the high-end market, increasing profits for both sellers.

4.2. Influence of δ and α

Figures 8–11 show the influence of the consumers' platform owner preference δ and the category vertical differentiation degree α on the equilibrium decision-making of both sellers (taking $t = 20$ as an example).

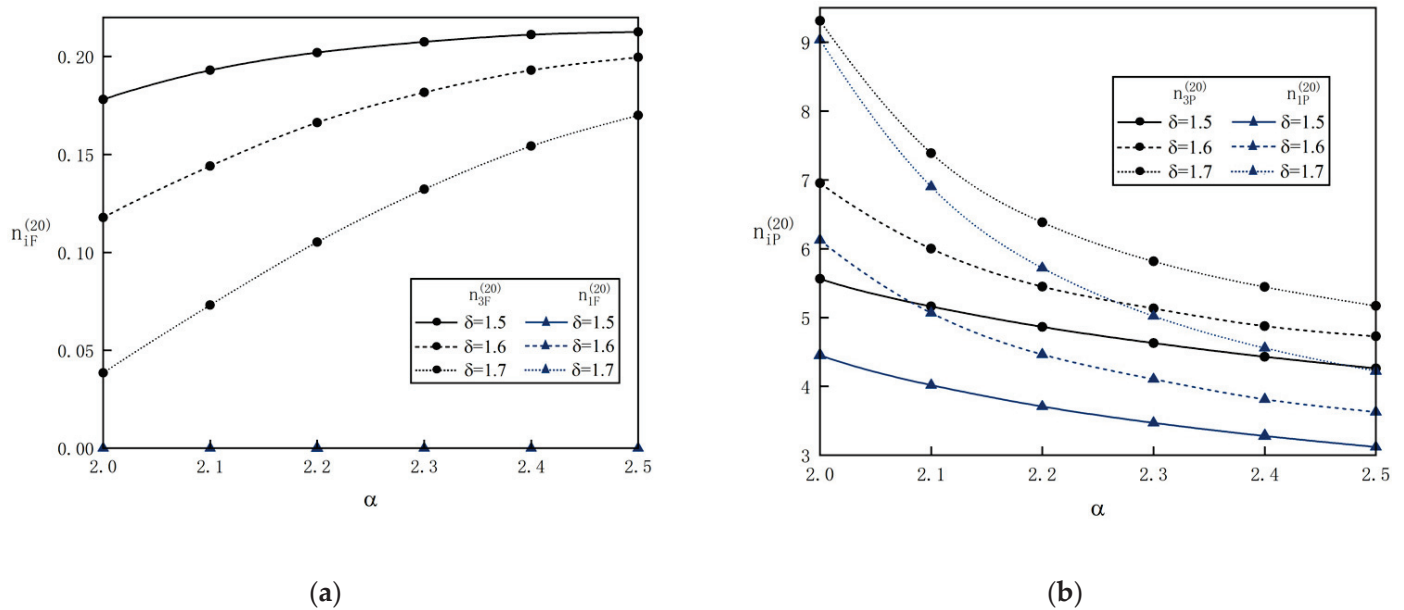
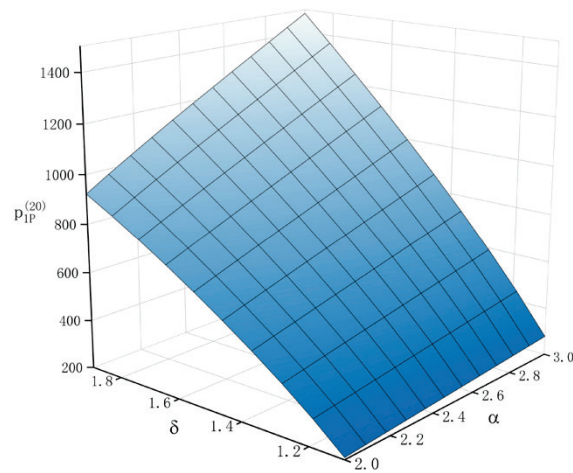
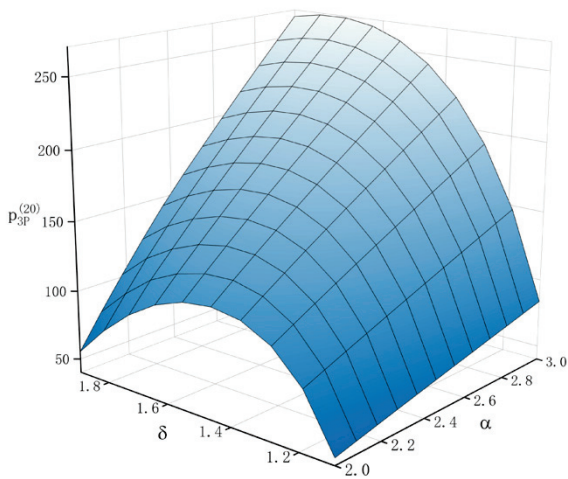


Figure 8. The variation in the number of variants of each category of each seller with δ and α : (a) the variation in the number of variants of category F of each seller with δ and α , and (b) the variation in the number of variants of category P of each seller with δ and α .

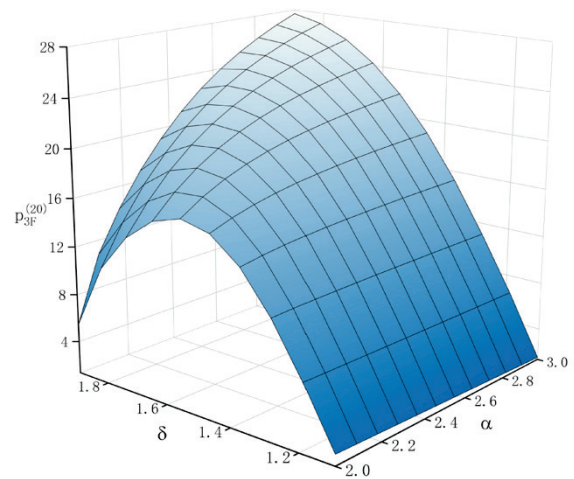
It can be seen from Figure 8 that the number of innovative variants of 3F-type is positively correlated with α ; in contrast, the number of innovative variants of 3P-type and the number of copycat variants of 1P-type are negatively correlated with α . For the category F, when the quality difference between it and the category P is large, the third-party seller will usually increase the number of variants of the category F to make up for the lack of quality value identification. On the contrary, when the quality of the category P is significantly different from that of the category F, both types of sellers perceive less necessity to maintain a high volume of variants in the category P.



(a)



(b)



(c)

Figure 9. The variation in retail price of types 1P, 3P, and 3F with δ and α : (a) the variation in retail price of 1P-type with δ and α , (b) the variation in retail price of 3P-type with δ and α , and (c) the variation in retail price of 3F-type with δ and α .

In addition, it can be seen from Figure 8 that for the category F, if consumers' platform owner preference is high, the number of variants of 3F-type will be reduced. This is because the platform owner has no new variant of the category F, there is less competition, and the third-party seller will reduce variants to accommodate the actual reduction in demand. For the category P, if consumers' platform owner preference is high, it will cause the third-party seller and the platform owner to increase variants of the category P at the same time. This is because there is a strong competitive relationship at this time, and both types of sellers take the decision to actively expand the number of category variants.

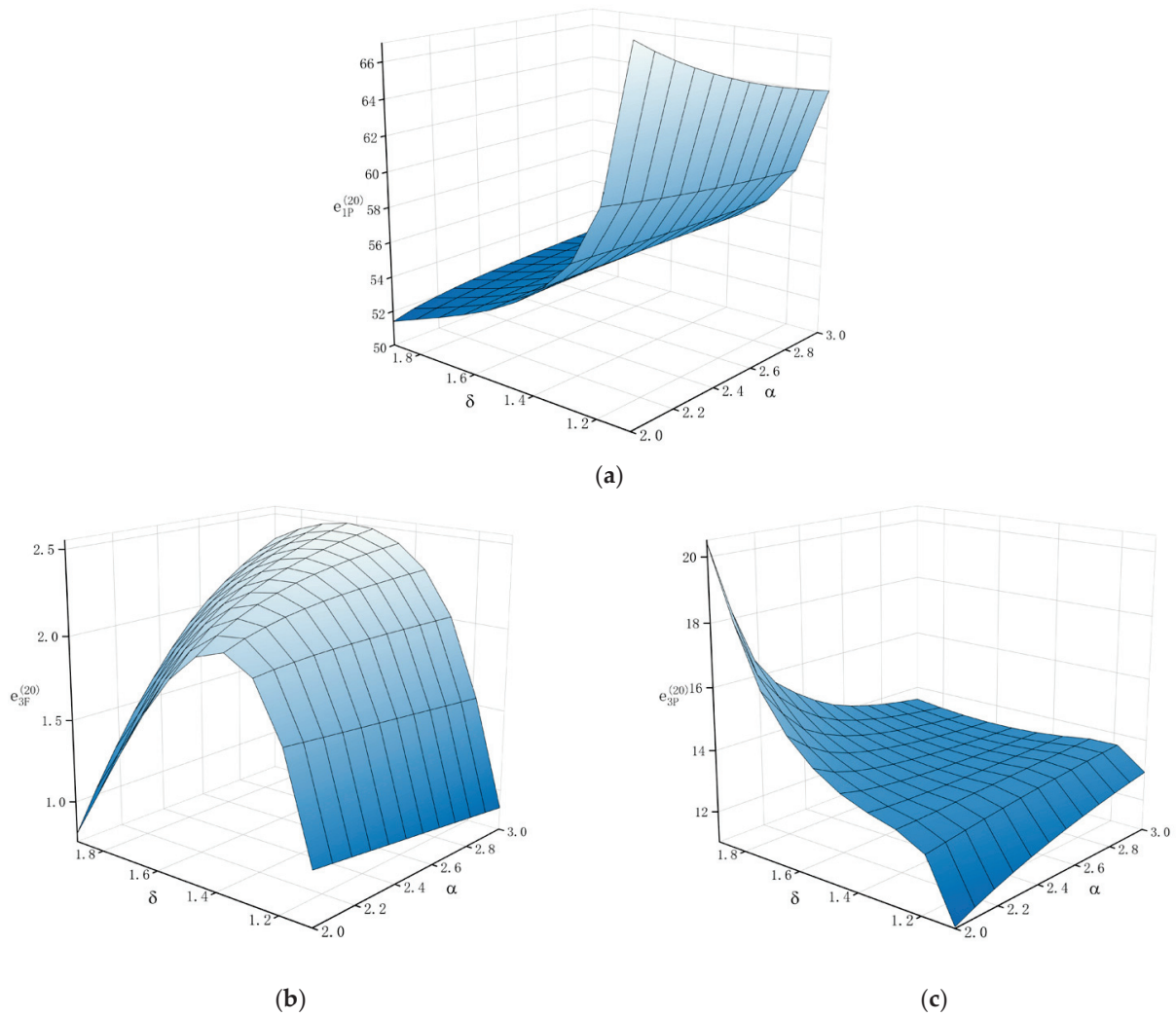


Figure 10. The variation in marketing effort level of types 1P, 3F, and 3P with δ and α : (a) the variation in marketing effort level of 1P-type with δ and α , (b) the variation in marketing effort level of 3F-type with δ and α , and (c) the variation in marketing effort level of 3P-type with δ and α .

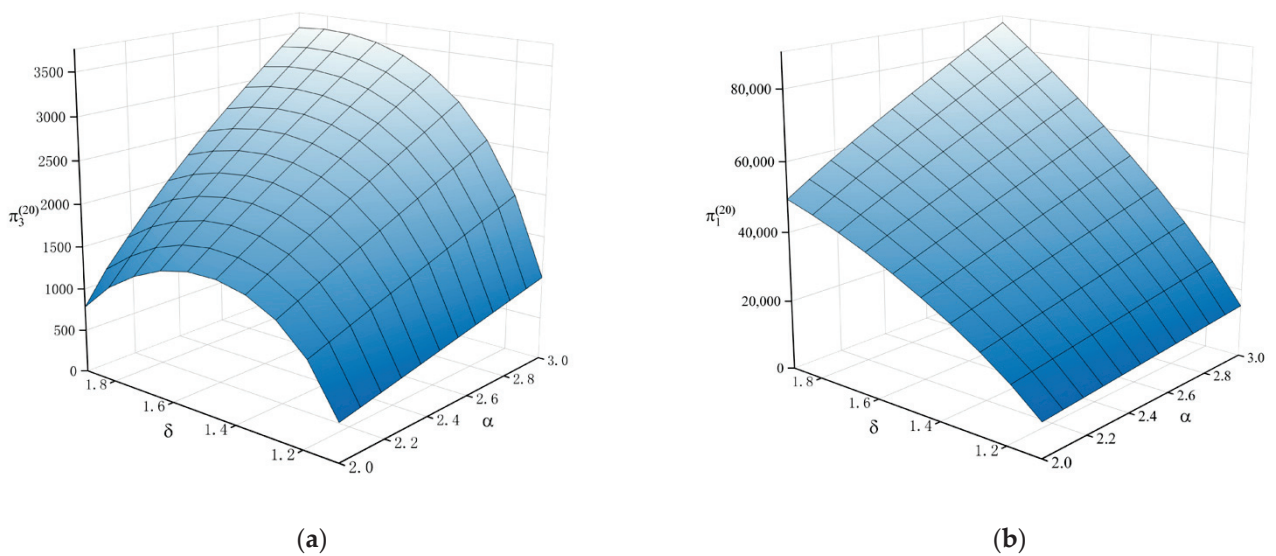


Figure 11. The variation of profit of each seller with δ and α : (a) the variation of profit of the third-party seller with δ and α , and (b) the variation of profit of the platform owner with δ and α .

For the 1P-type, when the quality difference between category P and category F increases, or consumers' platform owner preference increases, the price of 1P-type will increase almost linearly (see Figure 9a). For the 3P-type, when the quality gap between the category P and the category F increases, the price of 3P-type will monotonically increase. In addition, when consumers' platform owner preference increases, at first the third-party seller can free ride the rapid rise in the pricing of 1P-type and increase the price of his own category P. However, as consumers' platform owner preference exceeds the "free-riding window", the pricing of 3P-type will fall. Therefore, it can be seen from Figure 9b that the retail price of 3P-type first increases and then decreases with the increase in δ .

It can be seen from Figure 9c that the retail price of 3F-type also first increases and then decreases with the increase in δ . This is because the high-end category whose price starts to fall will form a crowding-out effect on the low-end category. In addition, when δ approaches 1.0, it can be seen from Figure 9c that the price of 3F-type will gradually decrease as the quality of the category P greatly exceeds that of the category F. However, when δ is at a high level, we can see that the price of 3F-type may not decrease as α increases. The independent seller also has the potential to increase 3F-type pricing by innovating more variants that create value for consumers. This is not surprising, as the platform owner actually does not encroach innovative variants of category F.

For the 1P-type, if consumers' platform owner preference is high, the platform owner sees no need to provide more marketing efforts for his products. Therefore, the marketing effort level of 1P-type decreases monotonically with δ (see Figure 10a). In addition, Figure 10a shows that when δ approaches 1.0, as the quality gap between the category P and the category F increases, the platform owner reduces the marketing effort level due to the reduction of copycat. However, when δ is at a high level, the platform owner promotes the third-party seller to innovate high-quality category variants through enhanced marketing efforts as α increases.

It can be seen from Figure 10b that the marketing effort level of 3F-type first increases and then decreases with the increase in δ . This is because when δ increases, the third-party seller will increase his marketing efforts for category F at first; however, if δ is larger, the third-party seller will shift more marketing efforts to category P with more innovative variants. In addition, Figure 10b shows that when δ approaches 1.0, the marketing effort level of 3F-type decreases as α increases. However, when δ is at a high level, the third-party seller will provide more marketing efforts for the increased number of 3F-type's innovative variants as α increases.

It can be seen from Figure 10c that the marketing effort level of 3P-type increases monotonically with δ when $\alpha \leq 2.4$. For the 3P-type, when α is small, the third-party seller will provide more marketing efforts for the increased number of innovative variants as δ increases. However, Figure 10c shows that the effect of δ on the marketing effort level of 3P exhibits a positive N-shaped characteristic of "increase first, then decrease and then increase" when $\alpha > 2.4$. This is because the increasing δ has to some extent discouraged the marketing enthusiasm of the third-party seller, but with the decline of the platform owner's marketing efforts, the third-party seller will seize this opportunity to promote his products. In addition, Figure 10c shows that when δ approaches 1.0, the third-party seller attracts consumers by increasing the marketing effort level as α increases; when δ is at a high level, the third-party seller reduces marketing investment due to the weakening competition in the high-end market as α increases.

It can be seen from Figure 11 that the third-party seller's profit first increases and then decreases with the increase in δ , and the platform's profit increases monotonically with δ . This is because increasing δ is conducive to promoting the horizontal innovation of category P, and the third-party seller can benefit more from free rides at first. However, as δ exceeds the "free-riding window", the 3P-type's actual demand and retail price decrease, resulting in lower profits for the third-party seller, while the platform owner's profit will increase monotonously. In addition, when α increases, the market demand and pricing of

category P increase, and since the marginal profit of category P is higher, the profits of both sellers increase monotonously with α (see Figure 11).

4.3. Influence of β and r

Figures 12–15 show the influence of diseconomies of scale β and commission rate r on the equilibrium decision-making of both sellers (taking $t = 20$ as an example).

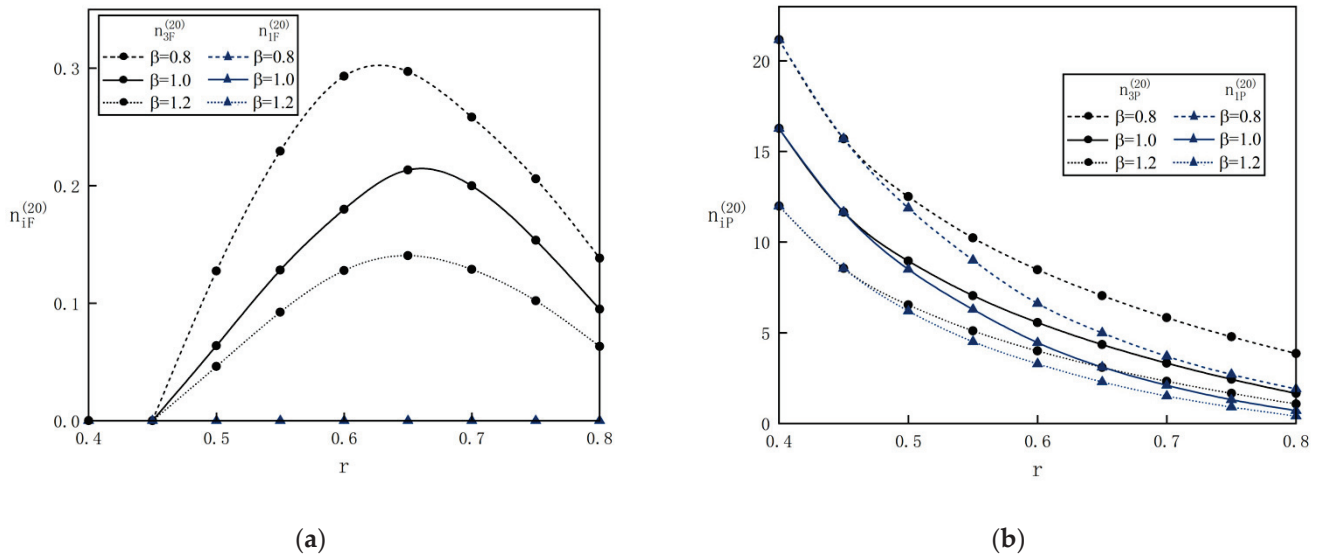


Figure 12. The variation in the number of variants of each category of each seller with β and r : (a) the variation in the number of variants of category F of each seller with β and r , and (b) the variation in the number of variants of category P of each seller with β and r .

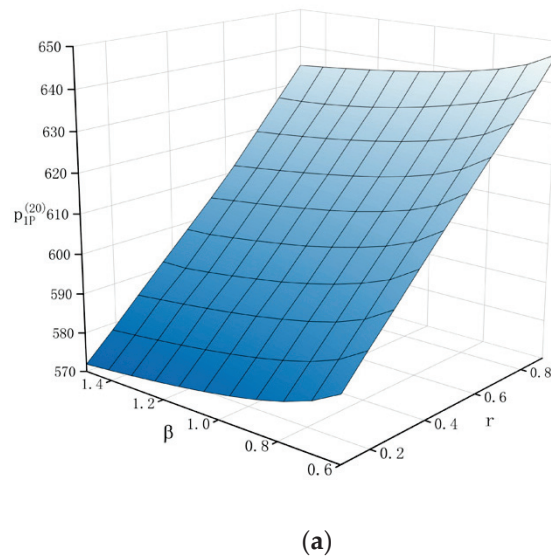


Figure 13. Cont.

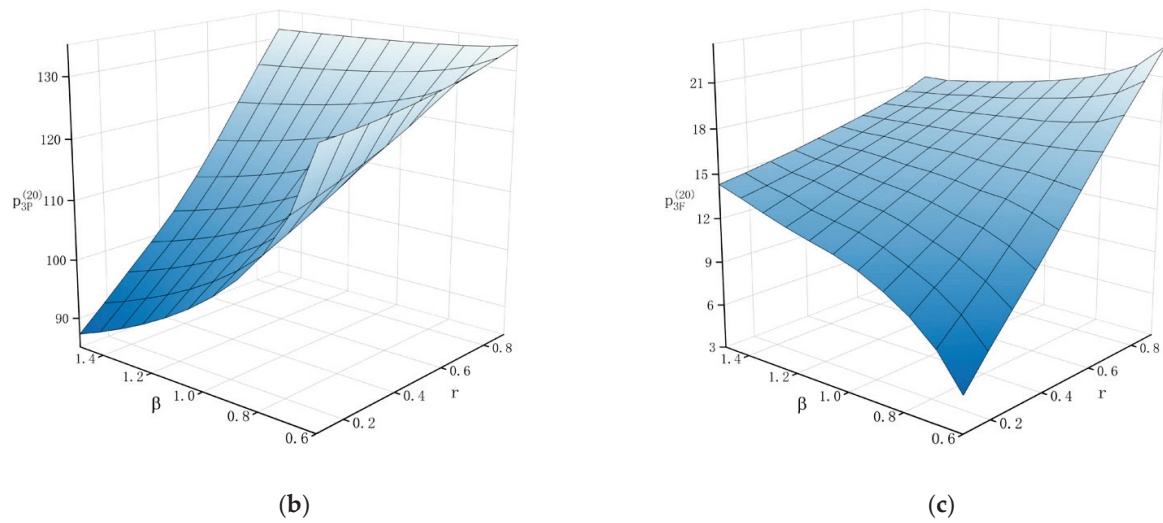


Figure 13. The variation in retail price of types 1P, 3P, and 3F with β and r : (a) the variation in retail price of 1P-type with β and r , (b) the variation in retail price of 3P-type with β and r ; and (c) the variation in retail price of 3F-type with β and r .

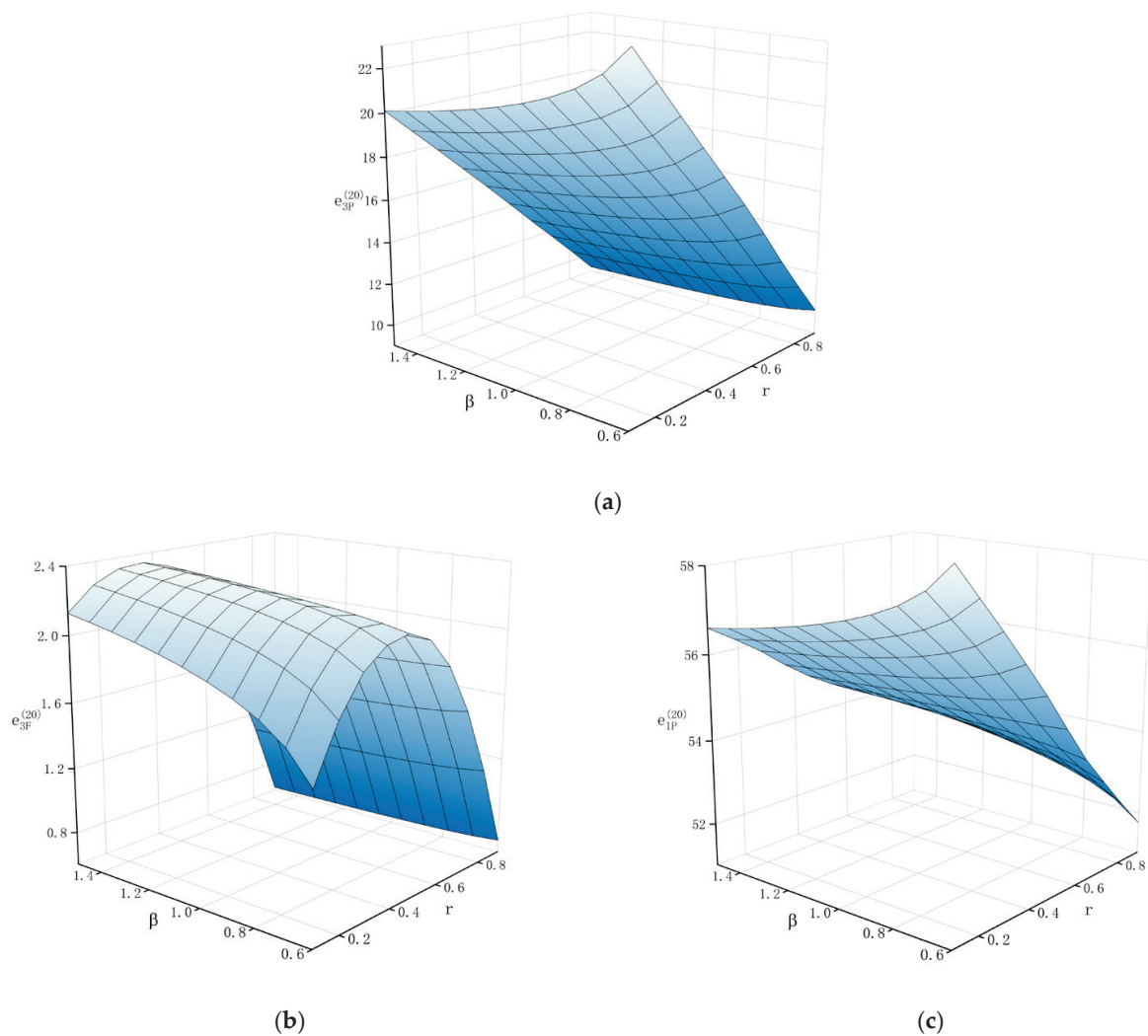


Figure 14. The variation in marketing effort level of types 3P, 3F, and 1P with β and r : (a) the variation in marketing effort level of 3P-type with β and r , (b) the variation in marketing effort level of 3F-type with β and r , and (c) the variation in marketing effort level of 1P-type with β and r .

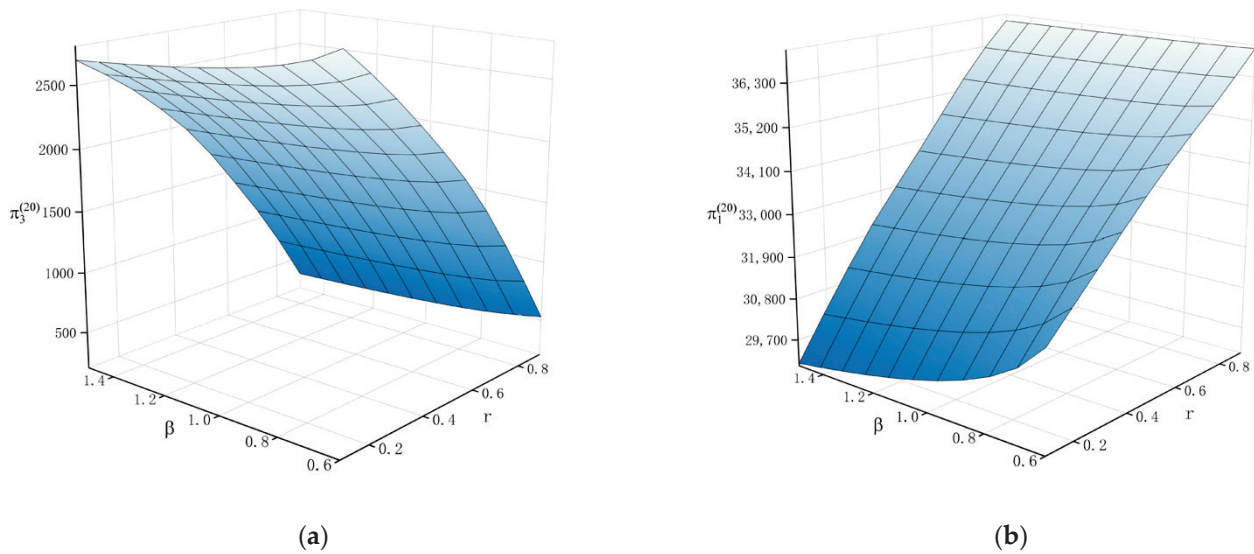


Figure 15. The variation of profit of each seller with β and r : (a) the variation of profit of the third-party seller with β and r , and (b) the variation of profit of the platform owner with β and r .

As can be seen from Figure 12, the number of innovative variants of 3F-type first increases and then decreases with the increase in r , the number of innovative variants of 3P-type is negatively correlated with r , and the number of copycat variants of 1P-type is negatively correlated with r . When the commission rate is quite low, the third-party seller does not innovate variants of category F and can gain more profits by innovating variants of category P; as r increases, the third-party seller starts expecting to increase profits by innovating lower-end products that the platform owner will not encroach on; however, when the r is quite high, the third-party seller is even forced to leave the platform, naturally reducing the investment in innovation. For the platform owner, when r is small, the platform owner is willing to directly benefit from his own business, resulting in an increase in the number of copycat variants. On the contrary, when r is large, the shared revenue (commission fee) is more important to the platform owner and the platform owner will reduce the encroachment of innovative products. In addition, Figure 12 shows that the higher the diseconomies of scale, the fewer the category variants in the online marketplace.

When β increases, the number of variants of category P decreases, and the price consumers are willing to pay for it decreases (see Figure 13a,b). It can be seen from Figure 13c that the retail price of 3F increases monotonically with β when $r \leq 0.5$ and decreases monotonically with β when $r > 0.5$. This is because when r is small, the third-party seller hardly innovates new variants of category F. If β increases, the competitive pressure from category P decreases, which increases the retail price of category F. However, when r is large, the third-party seller innovates variants of category F. If β increases, the innovative variants of category F reduce, resulting in a lower retail price. Moreover, the 3P-type with a significantly lower retail price also has a crowding-out effect on low-end 3F-type.

In addition, it can be seen from Figure 13 that the retail price of all categories is proportional to r . This is because the larger r is, the more the platform owner relies on shared revenue. In order to prevent the excessive decline of shared revenue, the platform owner pushes the third-party seller to set higher retail prices by actively raising the prices of self-operated products.

When β increases, the innovation investment of the third-party seller in category P decreases, and naturally, the marketing efforts on category P also decrease (see Figure 14a). It can be seen from Figure 14b that the marketing effort level of 3F-type increases monotonically with β when $r \leq 0.5$ and decreases monotonically with β when $r > 0.5$. As mentioned earlier, when r is small, the third-party seller hardly innovates new variants of category F. If β increases, the third-party seller will expand the low-end market by increasing his

marketing efforts to 3F-type. However, when r is large, the third-party seller innovates variants of category F. If β increases, the third-party seller reduces marketing efforts for category F with fewer innovative variants.

It can be seen from Figure 14c that the marketing effort level of 1P-type decreases monotonically with β when $r \leq 0.5$ and increases monotonically with β when $r > 0.5$. This is because when r is small, if β increases, the platform owner reduces the marketing effort level due to the reduction of copycat. However, when r is large, if β increases, the platform owner will motivate the third-party seller to be innovative by improving marketing efforts. In addition, it can be seen from Figure 14 that the marketing effort level of the above types varies with r in the same way as the number of variants with r .

It can be seen from Figure 15 that the third-party seller's profit and the platform owner's profit decrease monotonically with β . This is because when β increases, the horizontal innovation of each category decreases, resulting in lower profits for both sellers. In addition, Figure 15 shows that the third-party seller's profit decreases monotonically with r , while the platform owner's profit increases monotonically with r . For the third-party seller, when r increases, although the retail price of the category increases, the horizontal innovation degree of category P with higher marginal profit decreases, and the commission paid increases. Therefore, the overall profit of the third-party seller decreases. For the platform owner, when r increases, the commission charged increases, and the retail price of 1P-type increases. Therefore, the profit of the platform owner increases.

4.4. Managerial Insights

The following managerial insights based on the research results:

From the perspective of third-party sellers, third-party sellers must be wary of the platform owner's product encroachment, which leads to a reduction in the differentiation of the category, reducing their innovation margin profit. Therefore, in order to avoid the excessive decline of the innovation marginal profit caused by the platform owner's product encroachment, firstly, third-party sellers should adjust the innovation investment amount in each period according to the possible product encroachment behavior of the platform owner and increase the retail price and marketing effort level of innovative products. Secondly, as the platform owner focuses on the profitable category, third-party sellers can sell a variety of vertically differentiated categories at the same time. While competing fiercely with the platform owner in the high-end market, third-party sellers can also gain some profits by expanding their share of the low-end market. Thirdly, third-party sellers can also expand the vertical differentiation between the traffic-attracting category and the profitable category by improving the level of production technology and adding new variations of the traffic-attracting category to attract more consumers.

From the perspective of the platform owner, firstly, because the platform owner has a dominant market position, entering the third-party product market can bring more profits for himself. Therefore, the platform owner can choose to encroach on some products with high prices and deep product lines. Secondly, when considering whether to encroach on new products, the platform owner should not only weigh self-operated income and shared income, but also balance short-term profit through encroachment and damage to independent sellers' product innovation caused by excessive encroachment. To be precise, the platform's selective encroachment on new products can improve its own revenue while alleviating the inhibitory effect on the continuity of third-party sellers' category innovation. In addition, in order to avoid third-party sellers being forced out of the market by the platform owner's product encroachment, the platform owner can appropriately raise the retail prices of self-operated products and reduce marketing efforts to ease market competition. Thirdly, the platform owner can also improve consumers' platform owner preference by ensuring the high quality of platform services, thereby encouraging third-party sellers to invest in research and development of high-quality new products. Furthermore, the platform owner should set a reasonable commission rate to

ensure the variety of variations of various categories, thereby expanding the sales scope of the composite e-commerce platform.

5. Conclusions

This paper studies product encroachment by the e-commerce platform owner on independent third-party sellers' innovative products. The applied model considers the dual differentiation of categories and combines the method of multi-agent simulation to conduct a competitive dynamic simulation study. The following conclusions are obtained: (1) In the case where multiple categories are sold at the same time, the third-party seller will innovate variants for both the traffic-attracting category and the profitable category at the same time and invest more funds in innovative R&D of high-quality category P, and the profit-driven platform owner will only encroach on the new variants of the profitable category. (2) Consumers' platform owner preference and category vertical differentiation describe consumers' valuation of different categories, both of which affect the intensity of competition between categories and consumers' purchasing utility, thereby affecting the equilibrium decision-making of the third-party seller and the platform owner. When the categories of the platform owner have a greater valuation advantage, the third-party seller has a stronger incentive to innovate variants of category P, and the platform owner has a stronger incentive to encroach. When the valuation advantage of the high-quality category is obvious, the motivation of the third-party seller to innovate variants of category F increases. (3) The commission rate and diseconomies of scale directly affect the distribution of shared income and the marginal profit of category innovation, thus affecting the equilibrium decision-making of the third-party seller and the platform owner. If the commission rate is low, the third-party seller will invest in innovating variants of category P. If the commission rate is high, although the platform owner has a weak incentive to encroach, the third-party seller has little investment in product innovation. The diseconomies of scale of category diversity management costs hinder the growth of product variety in the online marketplace.

The research in this paper can be extended in the following directions: Firstly, the composite e-commerce platform model considered in this paper is relatively simple, with only the platform owner and one third-party seller. In the future, it can be extended to study the case of multiple independent sellers. Secondly, this paper assumes that consumers are rational and seek to maximize utility and does not consider consumers' strategic behaviors. Future studies should explore whether strategic consumers will guide the product innovation behavior of independent sellers and the product encroachment behavior of the platform owner through their own first-period purchases. Finally, this paper integrates product encroachment and product innovation and drives the platform owner's product encroachment with high profit. In the future, product encroachment can also be studied with the goal of regulating product quality.

Author Contributions: Conceptualization, T.Y. and Z.W.; methodology, Z.W.; software, T.Y.; validation, Z.W.; formal analysis, T.Y. and Z.W.; investigation, T.Y. and Z.W.; resources, T.Y. and Z.W.; data curation, Z.W.; writing—original draft preparation, Z.W.; writing—review and editing, T.Y. and Z.W.; visualization, Z.W.; supervision, Z.W.; project administration, T.Y.; funding acquisition, T.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Piezunka, H.; Katila, R.; Eisenhardt, K.M. Big fish or big pond? Seller's dilemma in intermediary selection. *AOM Proc.* **2016**, *1*, 10208. [[CrossRef](#)]
2. Khan, L.M. Amazon's Antitrust Paradox. *Yale Law J.* **2017**, *126*, 710–805.

3. Prince, J.T.; Simon, D.H. Do incumbents improve service quality in response to entry? Evidence from airlines' on-time performance. *Manag. Sci.* **2015**, *61*, 372–390. [[CrossRef](#)]
4. Ethiraj, S.; Zhou, Y.M. Fight or flight? Market positions, submarket interdependencies, and strategic responses to entry threats. *SMJ* **2019**, *40*, 1545–1569. [[CrossRef](#)]
5. Li, X.; Cai, X.Q.; Chen, J. Quality and Private Label Encroachment Strategy. *Prod. Oper. Manag.* **2021**, *31*, 374–390. [[CrossRef](#)]
6. Jiang, B.; Jerath, K.; Srinivasan, K. Firm Strategies in the 'Mid Tail' of Platform-Based Retailing? *Mark. Sci.* **2011**, *30*, 757–775. [[CrossRef](#)]
7. Ti, H.W.; Huang, L.; Pei, L.G. Amazon's Market Demand and Differential Product Selection Research. In Proceedings of the 2020 11th International Conference on E-Education, E-Business, E-Management, and E-Learning, Osaka, Japan, 10–12 January 2020; pp. 360–363.
8. Yu, Y.; Huang, G. Nash game model for optimizing market strategies, configuration of platform products in a Vendor Managed Inventory (VMI) supply chain for a product family. *Eur. J. Oper. Res.* **2010**, *206*, 361–373. [[CrossRef](#)]
9. Zhou, H.; Wang, S.Y.; Cheng, T.C.E. Competition and evolution in multi-product supply chains: An agent-based retailer model. *Int. J. Prod. Econ.* **2013**, *146*, 325–336.
10. Jiang, G.; Liu, S.; Liu, W.; Xu, Y. Agent-based modeling and simulation of the decision behaviors of e-retailers. *Ind. Manag. Data Syst.* **2018**, *118*, 1094–1113. [[CrossRef](#)]
11. Parker, G.; Van Alstyne, M.W. Two-sided network effects: A theory of information product design. *Manag. Sci.* **2005**, *51*, 1494–1504. [[CrossRef](#)]
12. Parker, G.; Van Alstyne, M.W. Innovation, openness, and platform control. *Manag. Sci.* **2014**, *64*, 2973–3468.
13. Gawer, A.; Henderson, R. Platform owner entry and innovation in complementary markets: Evidence from Intel. *J. Econ. Manag. Strat.* **2007**, *16*, 1–34. [[CrossRef](#)]
14. Zhu, F.; Liu, Q.H. Competing with Complementors: An Empirical Look at Amazon.com. *SMJ* **2018**, *39*, 2618–2642. [[CrossRef](#)]
15. Li, Q.; Wang, Q.S.; Song, P.J. Third-party sellers' product entry strategy and its sales impact on a hybrid retail platform. *Electron. Commer. Res. Appl.* **2021**, *47*, 101049. [[CrossRef](#)]
16. Hagiu, A.; Teh, T.H.; Wright, J. Should Platforms Be Allowed to Sell on Their Own Marketplaces? *Rand J. Econ.* **2022**, *53*, 297–327. [[CrossRef](#)]
17. Etro, F. Product selection in online marketplaces. *J. Econ. Manag. Strat.* **2021**, *30*, 614–637. [[CrossRef](#)]
18. Feng, N.; Chen, J.J.; Feng, H.Y.; Li, M.Q. Optimal product selection and pricing strategies for platform vendors under two-sided network effects. *Electron. Commer. Res. Appl.* **2020**, *43*, 100990. [[CrossRef](#)]
19. Wu, C.H.; Lai, J.Y. Dynamic pricing and competitive time-to-market strategy of new product launch under a multistage duopoly. *Eur. J. Oper. Res.* **2019**, *277*, 138–152. [[CrossRef](#)]
20. Yi, Y.Y.; Chen, Y.X. Product appearance and functionality quality competition model in toy manufacturers duopoly market. *CIMS* **2020**, *26*, 2216–2231.
21. Baron, D.P. Dynamic positioning, product innovation, and entry in a vertically differentiated market. *J. Econ. Manag. Strat.* **2021**, *30*, 287–307. [[CrossRef](#)]
22. Li, C.Y.; Zhang, C.H.; Li, Y.T. Research on Innovation R&D Strategy of Manufacturers Considering Following Product Encroachment under Differentiated Innovation Capability. *Chin. J. Manag. Sci.* **2021**, 1–12. [[CrossRef](#)]
23. Zhang, T.; Li, G.; Lai, K.K.; Leung, J.W.K. Information disclosure strategies for the intermediary and competitive sellers. *Eur. J. Oper. Res.* **2018**, *271*, 1156–1173. [[CrossRef](#)]
24. Shanguan, L.L.; Miao, Z.W.; Lan, Y.Q. Two-dimensional product differentiation and pricing strategies. *Syst. Eng. Pract.* **2021**, *41*, 93–112.
25. Jalali, H.; Van den Broeke, M.; Van Nieuwenhuyse, I. Platform and product design for markets with quality and feature sensitive customers. *Int. J. Prod. Econ.* **2022**, *244*, 108354. [[CrossRef](#)]
26. Tian, C.; Xiao, T.; Shang, J. Channel differentiation strategy in a dual-channel supply chain considering free riding behavior. *Eur. J. Oper. Res.* **2022**, *301*, 473–485. [[CrossRef](#)]
27. Lv, H. Who benefits when coupons are issued by a duopoly from an e-market? *Manag. Decis. Econ.* **2021**, *42*, 1656–1664. [[CrossRef](#)]
28. Feng, J.; Li, X.; Zhang, X.Q. Online Product Reviews-Triggered Dynamic Pricing: Theory and Evidence. *ISR* **2019**, *30*, 1107–1123. [[CrossRef](#)]
29. Liu, Y.M.; Yang, R.Y.; Fan, C. The Manufacturer's Strategy Choice of Online Customization Channel Considering Product Variety. *Soft Sci.* **2017**, *31*, 123–127+138.
30. Zhao, J.; Liu, J.; Mei, S.E. Optimization Strategies of Corporate Targeted Advertising Based on Dual Channels. *CIMS* **2019**, *25*, 2385–2394.
31. Shou, Y.Y.; Song, C.J. Optimization of R&D project partner selection for complex product systems. *Sci. Res. Manag.* **2014**, *35*, 144–149.

Earned Value Management Agent-Based Simulation Model

Manuel Castañón-Puga ^{1,*}, Ricardo Fernando Rosales-Cisneros ², Julio César Acosta-Prado ³,
Alfredo Tirado-Ramos ^{4,5}, Camilo Khatchikian ⁵ and Elías Aburto-Camaclanqui ⁶

¹ School of Chemistry and Engineering, Universidad Autónoma de Baja California, Tijuana 22424, Mexico

² Research Center of Complexity Studies, School of Accounting and Administration, Universidad Autónoma de Baja California, Tijuana 22424, Mexico

³ School of Business Science, Universidad del Pacífico, Lima 15072, Peru

⁴ Department of Epidemiology, Geisel School of Medicine at Dartmouth, Dartmouth College, Lebanon, NH 03756, USA

⁵ Department of Biomedical Data Science, Geisel School of Medicine at Dartmouth, Dartmouth College, Lebanon, NH 03756, USA

⁶ Research Center (CIUP), Universidad del Pacífico, Lima 15072, Peru

* Correspondence: puga@uabc.edu.mx

† Current address: School of Chemistry and Engineering, Universidad Autónoma de Baja California Tijuana Campus, Calzada Universidad 14418, Tijuana 22424, Baja California, Mexico.

Abstract: Agile project management (APM) can be defined as an iterative approach that promotes satisfying customer requirements, adjusts to change, and develops a working product in rapidly changing environments. Managers usually apply agile management as the project management approach in projects requiring extraordinary speed and flexibility in their processes. Earned value management (EVM) is a fundamental part of project management to establish practical measures. Often, managers use a task board to visually represent the work on a project and the path to completion. Still, managing an agile project can be a challenging endeavor. In this paper, we propose an agent-based model describing the management of tasks within a project using earned value assessment and a task board. Our model illustrates how EVM yields an efficient method to measure a project's performance by comparing actual progress against planned activities, thus facilitating the formulation of more accurate predicted estimations. As proof of concept, we leverage our implementation to calculate EVM performance indexes according to a performance measurement baseline (PMB) in a task board fashion.

Keywords: agile development; earned value management; task board; agent-based simulation

Citation: Castañón-Puga, M.; Rosales-Cisneros, R.F.; Acosta-Prado, J.C.; Tirado-Ramos, A.; Khatchikian, C.; Aburto-Camaclanqui, E. Earned Value Management Agent-Based Simulation Model. *Systems* **2023**, *11*, 86. <https://doi.org/10.3390/systems11020086>

Academic Editor: Philippe Mathieu, Juan M. Corchado, Alfonso González-Briones and Fernando De la Prieta

Received: 29 December 2022

Revised: 24 January 2023

Accepted: 26 January 2023

Published: 7 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Agile project management (APM) can be defined as an iterative approach that promotes satisfying customer requirements, adjusts to change, and develops a working product in rapidly changing environments [1]. Applying agile management as the project management approach requires extraordinary speed and flexibility in your processes and the formation of dedicated teams willing to adapt to changes, according to the Project Management Institute (PMI) [2]. Such a management approach is not only suitable for software development [3], but it has also been expanded to other environments, such as manufacturing, education, and health care, among others within the guide's scope [4].

In the manufacturing sector, there is much interest in understanding the determinants of effective agile project management to save time and energy in a context where customer requirements are broader, and new proposals for technological innovation are appearing [5,6]. For example, there is the proposal of a matrix that suggests agile practices based on the objectives and priority principles for complex project teams [5]. In addition, a scheme has been elaborated with the necessary actions to increase the probability of success in each of

the phases of agile projects (planning, implementation, and closure) in a cycle of continuous improvement [6].

The earned value management (EVM) [7] is considered a fundamental part of the project management body of knowledge (PMBOK) [2] to establish practical measures. Over the last four decades, project management professionals have used this method to measure performance and assess the status of a project [8]. Still, managing an agile project can be a challenging endeavor. Often, managers use a task board to visually represent the work on a project and the path to completion. The route includes pending, in-progress, and completed tasks performed by teams. For example, the “Kanban” methodology uses a task board to distribute assignments and activities as a fundamental part of a production process [9].

Implementing agile project management can be a complex, time-demanding undertaking, taking into consideration that different mechanisms influence the project’s performance. For example, cultural agency theory [10] proposes that operative play rules, individual traits, and cultural matters interact dynamically to produce emergent behaviors in the production system. From this theory, we could see EVM and a task board as the agency’s operative system. Such an approach could be a critical step for a comprehensive understanding of the agile process rules and development. This allows a constant evaluation of the intermediate results and allows adjustments if the users and the interested parties want them. This way, the entire project team, including stakeholders, continuously improves the product. This methodology allows for immediate product modifications as previously unknown requirements are discovered [1].

Therefore, we propose an agent-based model describing the management of tasks within a project using a task board. The model’s purpose is to illustrate how the participants in a project complete the tasks represented on the board. We consider the EVM approach to assess performance and control the work completion level compared to the set plan. In this study, we first explicitly identify the problem that motivated this work. Second, we describe the proposed model and briefly discuss the model’s benefits and limitations. Finally, we provide a set of conclusions and identify needs for future work and developments.

2. Problem Statement

According to the PMI [2], project management is the application of knowledge, skills, tools, and techniques to project activities to achieve the expected results. Generally speaking, traditional project management has been oriented to projects whose phases were programmable, with predictable endings; tasks, times, and deadlines were clearly established and defined with technical prescriptions. That is, the tasks that make them up were explicitly defined during the project planning process [11].

However, changes in technology, business, economics, and stakeholder expectations imply that project management considers a static component (pre-plannable) and a dynamic component (unpredictable and not initially programmable). Considering this dichotomy, organizations require flexibility to adopt different methodologies and techniques in project execution [12].

In addition, project management involves carrying out a set of functions performed by groups that interact reciprocally and configure an organizational system that must be appropriately coordinated. For the PMI, stakeholders are people and organizations that actively participate and whose interests may be affected due to the project execution [2]. According to the methodology of stakeholders, there are four main processes in project management: planning, design, execution and control, and closure. This study focuses on the execution and control phase, where the promoters and executors participate most in developing the activities planned through the task board [13].

Using this conceptual framework, it is possible to evaluate the elements involved in the planning and development of organizational projects with the help of models. Therefore, we propose developing an agent-based model to explore different scenarios that seek to manage the increasing complexity of the systems to be designed and implemented as an

alternative solution to specific problems in project management. The EVM technique is used to compute the performance and control the level of work achieved compared to the plan [14], addressing the following questions: How do employee conditions affect the performance of an agile project? Do the number of employees and the number of tasks each simultaneously affect cost performance? Does the likelihood of employees performing their tasks faster or slower cause convenient advances or inconvenient delays affecting cost performance? Under what circumstances do projects become so unpredictable that they could be considered complex?

3. Methodology

Social simulation has been gaining ground in the social sciences as a way of approaching the complexity of social systems. Computational social science has now incorporated data science into its arsenal of techniques but has also included alternative methods, such as agent-based modeling, from the outset. Agent-based modeling (ABM) is a method of computational modeling and simulation to study complex systems' organization and dynamics.

We consider that project management is complex for several reasons: first, because it is a process where humans make decisions (not as rational as one would expect); second, because there are structural constraints that condition their behavior; and finally, because social processes affect the culture of organizations. Consequently, we regard earned value management as a model that reduces the issue's complexity to create the illusion of simplicity due to focusing on optimizing performance and costs.

We based the methodology's sequence on the well-known social simulation approach in which the procedure selected and represented real-life targets in a simplified way through a model executed and outputs data [15]. In this work, we use an agent-based system to approach the EVM agency as an operating system (structure and imperatives for decisions, operative intentions, etc.) and simulate hypothetical scenarios from an exploratory and illustrative point of interest in cultural agency theory [10].

3.1. Modeling and Simulation Method

The following is a brief description of the adopted modeling and simulation easyABMS methodology [16]. In this process, all steps can go back to the previous step, so the analyst and modeler can generate multiple approaches til the objective. We finalize with results analysis, as seen in Figure 1.

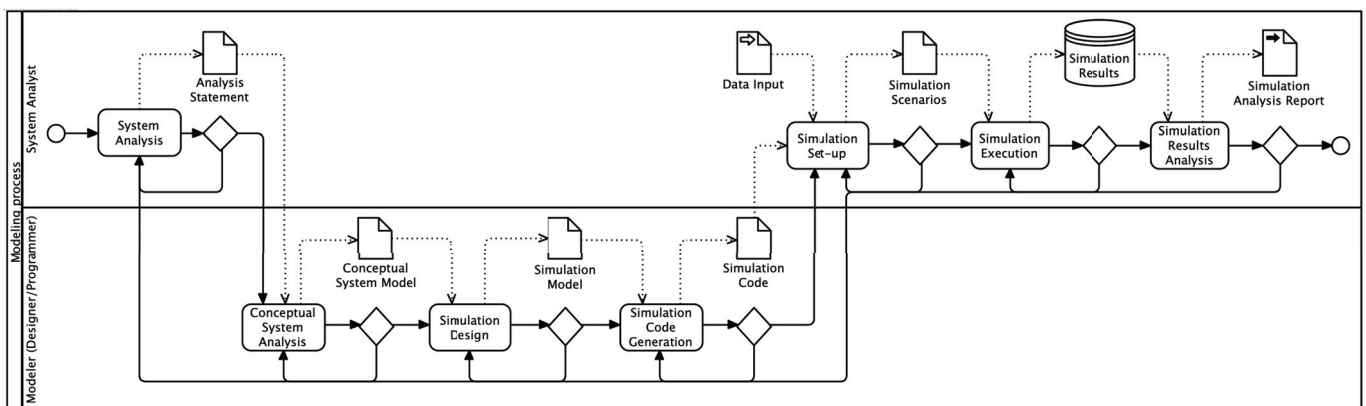


Figure 1. The adopted modeling and simulation process based on easyABMS methodology [16].

- System analysis. In this activity, we establish the aim of the model based on the research questions. The result is an analysis statement. In our case, it is a narrative document based on the ODD protocol that defines the purpose and details of the model we built.

- Conceptual modeling of the system. In this activity, we analyze the problem domain's language to make a first approximation. The result is a conceptual system model. We use the Unified Modeling Language (UML) to represent the abstractions produced in the analysis of the problem language.
- Simulation design. In this activity, we design the simulation. The result is a simulation model based on a specific framework or tool. We use the Netlogo tool as the technological basis for the design.
- Simulation Code Generation. In this activity, we write a computer executable code that implements the designed model in the selected tool. The result is a simulation code. The generated code is written in Logo for Netlogo and implements the simulator design.
- Simulation Setup. In this activity, we configure the experiment in the simulator. Using input data, we specify simulation scenarios. We used Netlogo's BehaviorSpace tool to experiment with a dataset based on a typical software project management template with 61 core tasks and a max of seven employees. This experimentation consisted of 2100 runs resulting from the combination of input variables and their possible valid values.
- Simulation execution. In this activity, we ran the experiment within the pre-set parameters. We obtained simulation results. The data obtained are the product of each "tick" (the discrete-time in Netlogo) and the states of all the input variables, agents, and earned value management metrics produced in each of the 2100 runs. The resulting data give us system state information in the entire parameter space.
- Simulation Results Analysis. In this activity, we analyze the results to contribute to the clarification of the proposed research questions. We use the resulting data to generate a simulation analysis report. We performed the following: (a) a t-Student test to compare dissimilarities in the results of simple scenario simulations between our prototype and tools suggested by PMI to analyze the EVM in hypothetical projects; (b) a sensitivity assessment to support the interpretation; (c) an explanation of simulation model outcomes and an active nonlinear test to examine the necessary considerations in the simulation structure and thereby begin to approach complexity.

3.2. Model Description

To formalize the proposed model, we followed the "S1: ODD Guidance and Checklists," proposed in [17], which provides guidance and checklists for writing "Overview, Design Concepts, Details" protocol (ODD) descriptions of agent-based or other simulation models. It is based on the ODD version published in earlier versions [18,19].

3.3. Model Validation

To validate the proposed model, firstly, we compared the results of simple scenario simulations between our prototype and tools suggested by PMI to analyze the EVM in hypothetical projects. For example, The Earned Value Management Calculator [20] or EVM Worksheet Package [21] could help compare results. Further, we applied a sensitivity assessment to support the interpretation and explanation of simulation model outcomes. Finally, we executed nonlinear active tests (ANTs) [22] to examine the necessary considerations in the simulation structure and thereby begin to approach complexity.

4. Results

Earned value management is founded on a set of metrics focused on evaluating the progress of a project from a cost and schedule standpoint. Figure A2 in the Appendix A.3.3 shows graphically how a project can be evaluated in execution time and how these metrics characterize its development. The cost performance index (CPI) and schedule performance index (SPI) metrics measure project performance. For example, the cost performance index (CPI) depends on comparing whether the actual cost (AC) corresponds to the estimated cost (EC). The earned value (EV) metric measures whether the project has economic gains

or losses. The model shows the behavior of these metrics during an artificial execution of a project (either using data obtained from a data file or artificially generated). We describe full EVM metrics in Table A4 in Appendix A.3.3.

The concept of EVM is introduced in the model, which is simulated through a spatial model of agents developed in the NetLogo programming environment [23]. A complete, detailed model description, following the ODD [17–19], is provided in Appendix A.

4.1. Netlogo Prototype

As a result of the agent-oriented analysis and design process, we produced an agent-oriented model in NetLogo based on our core code [24]. Figure 2 shows an EVM model NetLogo prototype screenshot. The NetLogo prototype used in this paper is available in [25] and can be downloaded directly from the repository online.

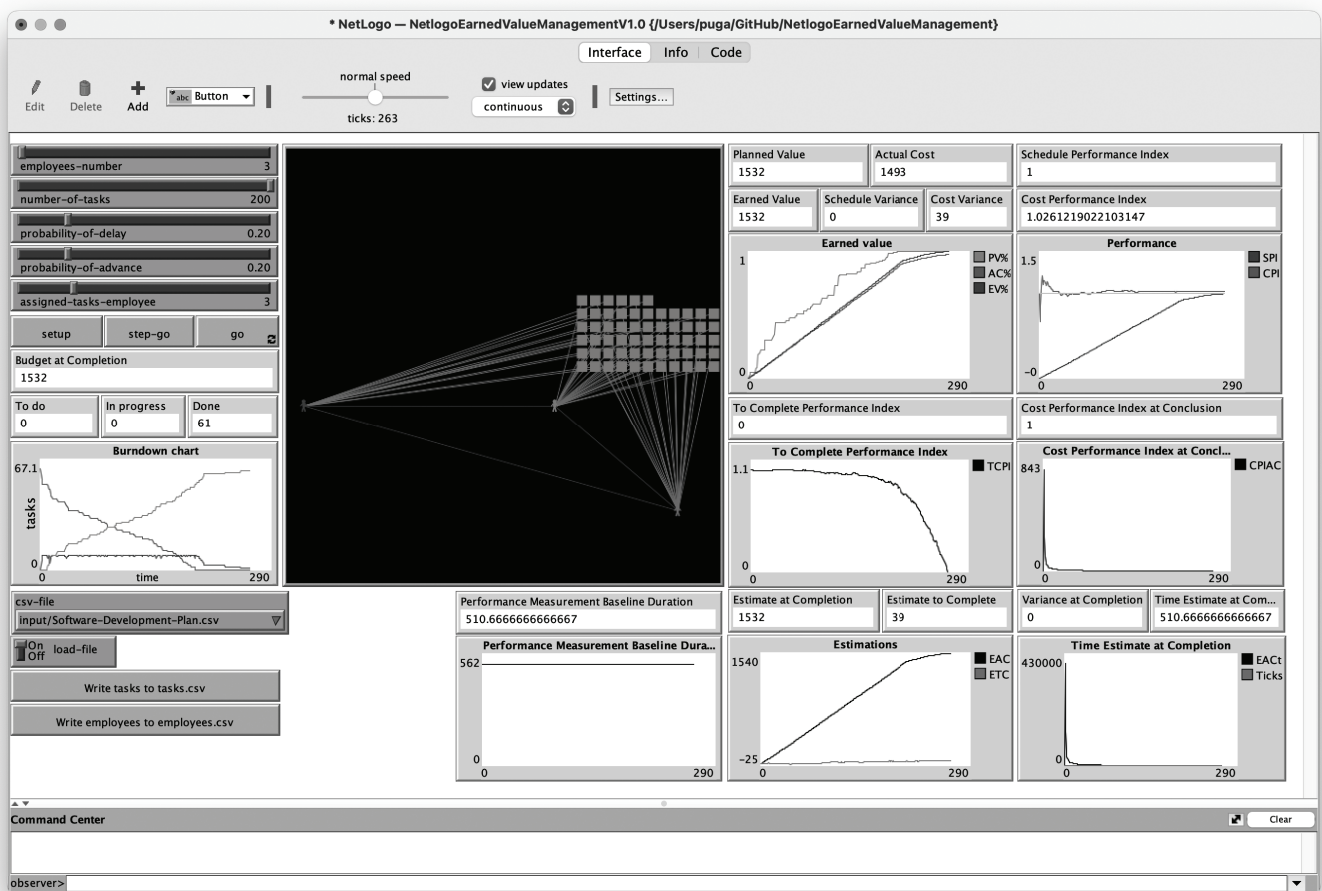


Figure 2. Earned value management (EVM) model in NetLogo.

First, the EVM model illustrates a set of tasks (backlog) in a task board (a Kanban task board style) at the top of the visual area of the simulator. The board has three columns, where each column denotes the status of the task: “To-do,” “In progress,” and “Done” tags. Then, we represent employees in the workspace at the bottom of the visual simulation area. A graphic link connects employees with assigned tasks. They take assignments from the “To-do” column to process the jobs (“In progress” cue) and transfer the finished task mark to the “Done” column. Finally, on the left are input controls to initialize different simulation scenarios, and on the right are additional output controls. The outputs show the results of the EVM in a dynamic way that reacts to the simulation process in real-time.

We designed the interface so the user can see how the variables behave in the form of a dashboard while the model simulates the initially configured scenario. Although the interface can display these inputs and outputs, the Netlogo tool can export a log file for better results processing. For example, Figure 3 plots the most significant inputs and outputs for EVM. These are the reproduction of the “Burndown,” “Earned Value,” and “Performance” charts shown in Figure 2 from a log file.

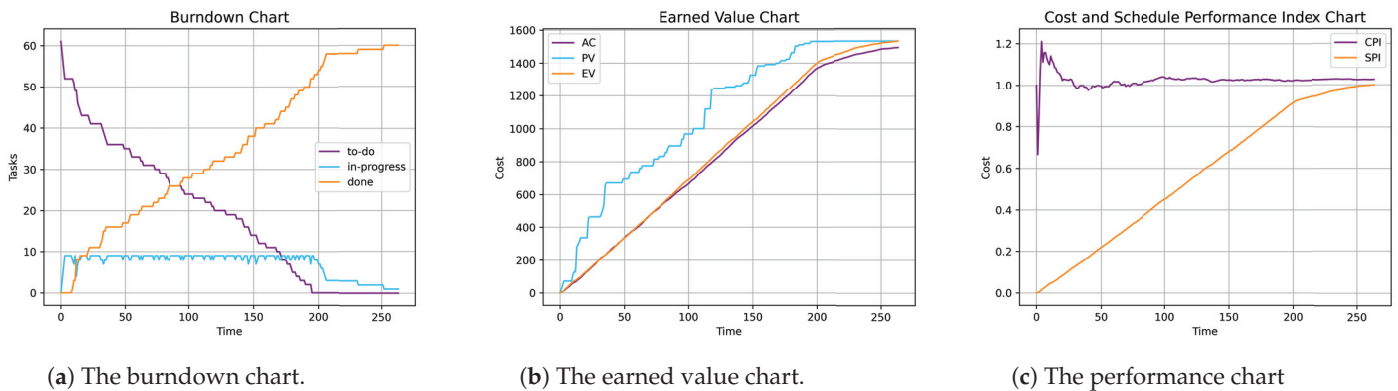


Figure 3. The plots are the most significant inputs and outputs for EVM. (a) The burndown chart, where tasks go through the to-do, in-progress, and done states during project execution. (b) The earned value chart compares the planned and actual costs. (c) The CPI and SPI chart depicts performance.

In Figure 3a, we show the burndown chart where tasks go through the to-do, in-progress, and done states during project execution. The prototype interface provides this standard visualization of task execution. In Figure 3b, we show the earned value chart that we used to compare the planned and actual costs. The prototype interface also offers this standard visualization of EVM. In Figure 3c, we show The CPI and SPI chart to depict performance. The interface shows the visualization of these metrics too. In this case, we are interested in showing the behavior of the CPI for the scope of this paper.

4.2. Model Validation

To validate the model, we tested with 2100 simulations. We established a fixed set of input tasks based on a typical planning template for a software development project. The template supplied 61 tasks with estimated costs and team members. Based on the information from this simple case study, we adjusted the values of the input variables in suitable ranges to calculate a proper sample of tests. This configuration helped us to observe the behavior of the cost performance index (CPI) and the project’s final cost under different conditions. Table 1 shows the input variables of the experiment and their value ranges.

Table 1. Validation settings. (See the Table A3 in the Appendix A to set up the variables description.)

Variable/Metric	Type	Values Range
number-of-tasks	input	61
employees-number	input	1–7
probability-of-delay	input	0.0–0.9
probability-of-advance	input	0.0–0.9
assigned-tasks-employee	input	1–3
step	output	1–n
CPI	output	0–n

The experiment produced much information, but the most relevant is the final state of the variables at the end of each simulation. We obtained a total of 2100 final results. Table 2 shows the statistical description of the data obtained during this process.

Firstly, we used the EVM Calculator (“EVM Calculator V2” MS Excel file), downloadable from the PMI website, to calculate the performance indexes and other EVM metrics using the same simulated data scenarios [6]. Appendix A.3.3 of Appendix A describes the EVM main variables and performance and estimations formulas. We planned an exploratory experiment focused on planned value, actual cost, and earned value, and the scheduled performance index (SPI) and cost performance index (CPI) EVM metrics to compare similarities. Table 2 shows a t-Student test result. Practically, the results are very identical.

Table 2. CPI t-Test: Two-Sample Assuming Unequal Variances.

	CPI-Netlogo Sample	CPI-EVM Calculator Tool Sample
Mean	2.75497723	2.754977232
Variance	14.5634324	14.56343242
Observations	2100	2100
Hypothesized mean difference	0	
df	160	
t stat	0	
P(T <= t) one-tail	0.5	
t critical one-tail	1.6544329	
P(T <= t) two-tail	1	
t critical two-tail	1.97490156	

Subsequently, we performed a sensitivity analysis to understand how the outputs change over the full range of possible inputs. We show the basic statistic dataset description in Table A5 in Appendix B, and we define the requirements verification in Table A6. In Table A7, we observe that 85% of “CPI” cases are within the range of 0.0961164439425309 to 3.5 (about 1880 of the 2100 tests).

In Figure 4, we depict the result of a detailed sensitivity analysis. We display a range of possible output values associated with each set of inputs. In our case, we analyze the possible combinations between the number of employees, the number of tasks each worker could perform simultaneously, the possibility of advancing the work, and the possibility of being delayed.

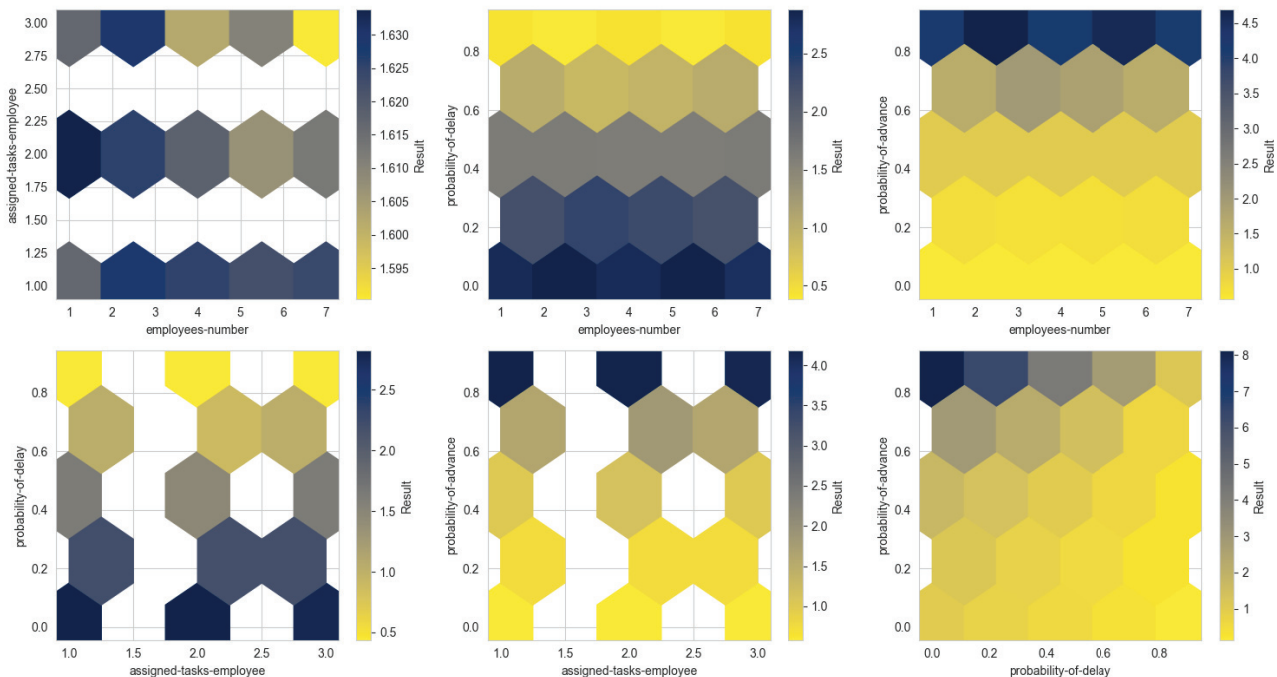


Figure 4. CPI sensitivity analysis results. The output plots.

Table 3 shows the results of the sensitivity analysis of the CPI concerning the number of employees and the tasks assigned to the employee. It shows that the number of employees and the number of tasks do not impact the cost performance index (CPI). The table shows high values in all cases without much variation.

Table 3. CPI sensitivity analysis results. The employees’ number versus the assigned tasks to an employee. The darker color in the table means a higher CPI value.

Assigned-Tasks-Employee Employees-Number	1	2	3
1	1.616634	1.633812	1.616834
2	1.633606	1.614470	1.628857
3	1.623663	1.637134	1.629288
4	1.625330	1.617751	1.602271
5	1.628319	1.573290	1.607234
6	1.616457	1.640380	1.613923
7	1.624135	1.612656	1.590358

Table 4 shows the results of the sensitivity analysis of the CPI concerning the number of employees and the probability of advancing. It shows that the number of employees and the advancement also affect the cost performance index (CPI). The table shows high CPI values when the probability is high and low values when the probability is low but does not vary much with the number of employees involved.

Table 4. CPI sensitivity analysis results. The employees’ number versus the probability of advance. The darker color in the table means a higher CPI value.

Probability-of-Advance Employees-Number	0.000000	0.100000	0.200000	0.300000	0.400000	0.500000	0.600000	0.700000	0.800000	0.900000
1	0.548008	0.609541	0.685730	0.783219	0.915103	1.106740	1.377564	1.846275	2.740608	5.611477
2	0.550267	0.607176	0.685457	0.787587	0.913403	1.104403	1.372748	1.817562	2.782771	5.635066
3	0.550096	0.612313	0.685667	0.787730	0.910915	1.101636	1.378627	1.841838	2.756855	5.674607
4	0.551275	0.609906	0.689134	0.788121	0.916298	1.103034	1.365377	1.839154	2.807393	5.481482
5	0.549912	0.610428	0.685860	0.779776	0.913223	1.111374	1.368596	1.828576	2.770266	5.411466
6	0.550164	0.610680	0.686776	0.785780	0.921015	1.104609	1.365208	1.859988	2.772758	5.578888
7	0.548133	0.611021	0.686966	0.785835	0.923714	1.109246	1.376516	1.819473	2.795025	5.434565

Table 5 shows the results of the sensitivity analysis of the CPI concerning the number of employees and the probability of delay. It shows that the number of employees and the delay also affect the cost performance index (CPI). The table shows high CPI values when the probability is low and low values when the probability is high but does not vary much with the number of employees involved.

Table 5. CPI sensitivity analysis results. The employees’ number versus the probability of delay in task execution. The darker color in the table means a higher CPI value.

Probability-of-Delay Employees-Number	0.000000	0.100000	0.200000	0.300000	0.400000	0.500000	0.600000	0.700000	0.800000	0.900000
1	2.923209	2.697309	2.356683	2.081919	1.788999	1.462714	1.165987	0.881181	0.575454	0.290812
2	2.986358	2.658718	2.394938	2.021872	1.749269	1.494834	1.172753	0.885583	0.597050	0.295065
3	2.993954	2.679616	2.397810	2.044512	1.760310	1.498649	1.168188	0.878099	0.584551	0.294594
4	2.953739	2.633249	2.347091	2.043536	1.743552	1.478904	1.176356	0.892230	0.587110	0.295405
5	2.950445	2.638388	2.290606	2.013881	1.761984	1.465932	1.158620	0.873110	0.583534	0.292974
6	3.003698	2.675100	2.340356	2.055888	1.762180	1.451223	1.187985	0.892449	0.576981	0.290005
7	2.896081	2.611585	2.321156	2.056459	1.793138	1.491311	1.171935	0.872740	0.581986	0.294104

Table 6 shows the results of the sensitivity analysis of the CPI concerning the number of tasks assigned to an employee simultaneously and the probability of performing tasks quickly. It shows that the number of tasks and the progress also affect the cost performance index (CPI). The table shows high CPI values when the probability is high and low values when the probability is low but does not vary much with the number of tasks involved.

Table 6. CPI sensitivity analysis results. The assigned tasks to employees versus the probability of advance in task execution. The darker color in the table means a higher CPI value.

Probability-of-Advance Assigned-Tasks-Employee	0.000000	0.100000	0.200000	0.300000	0.400000	0.500000	0.600000	0.700000	0.800000	0.900000
1	0.550079	0.610386	0.686651	0.785175	0.915242	1.107459	1.374933	1.840992	2.750297	5.618992
2	0.549219	0.610502	0.688131	0.786606	0.916836	1.103004	1.370400	1.834234	2.794441	5.531616
3	0.549782	0.609569	0.684757	0.784525	0.916638	1.107127	1.370940	1.833146	2.780553	5.489771

Table 7 shows the results of the sensitivity analysis of the CPI concerning the number of tasks assigned to an employee simultaneously and the probability of being late in performing the tasks. It shows that the number of tasks and the delay also affect the cost performance index (CPI). The table shows low CPI values when the probability is high and high values when the probability is low but does not vary much with the number of tasks involved.

Table 7. CPI sensitivity analysis results. The assigned tasks to employees versus the probability of delay in task execution. The darker color in the table means a higher CPI value.

Probability-of-Delay Assigned-Tasks-Employee	0.000000	0.100000	0.200000	0.300000	0.400000	0.500000	0.600000	0.700000	0.800000	0.900000
1	2.978426	2.663062	2.362711	2.046665	1.772522	1.479867	1.173713	0.888923	0.580200	0.294114
2	2.988602	2.658812	2.314510	2.069138	1.740356	1.477683	1.173824	0.882516	0.585297	0.294250
3	2.907608	2.646968	2.372195	2.020511	1.784022	1.475407	1.167531	0.875158	0.585931	0.291476

Table 8 shows the results of the sensitivity analysis of the CPI concerning the probability of being ahead of schedule and the probability of being late in performing the tasks. It shows that overtaking and delay affect the cost performance index (CPI). The table shows that, when the probability of advancing is high and the probability of delay is low, then performance increases. Conversely, when the probability of advance is low and the probability of delay is high, then performance drops. There is a strong relationship between these two input variables and the output variable CPI.

Table 8. CPI sensitivity analysis results. The probability of delay versus the probability of advance in task execution. The darker color in the table means a higher CPI value.

Probability-of-Advance Probability-of-Delay	0.000000	0.100000	0.200000	0.300000	0.400000	0.500000	0.600000	0.700000	0.800000	0.900000
0.000000	1.000000	1.108873	1.250670	1.424626	1.662969	2.030450	2.467776	3.331564	5.110659	10.194536
0.100000	0.901227	0.991782	1.126371	1.282113	1.493521	1.806010	2.257168	3.012464	4.621043	9.071108
0.200000	0.801303	0.891224	0.996478	1.148412	1.339701	1.606832	1.991850	2.699662	4.007647	8.014947
0.300000	0.697102	0.779057	0.873445	0.999431	1.166258	1.405789	1.750664	2.314024	3.465186	7.003424
0.400000	0.602836	0.666494	0.746477	0.857062	1.000851	1.202470	1.490557	1.995534	3.001171	6.092882
0.500000	0.497076	0.552568	0.622811	0.714094	0.830579	1.004919	1.256750	1.665710	2.559082	5.072936
0.600000	0.397853	0.445080	0.499436	0.571599	0.666491	0.806707	1.009625	1.332263	1.995027	3.992812
0.700000	0.301044	0.331156	0.373308	0.426633	0.499541	0.596892	0.754331	1.003200	1.497278	3.038604
0.800000	0.198708	0.224264	0.250335	0.287390	0.335720	0.399535	0.495143	0.672126	0.993766	1.981109
0.900000	0.099786	0.111024	0.125797	0.142994	0.166754	0.199028	0.247045	0.334692	0.500109	1.005570

Finally, we tested the model’s structure and robustness using the nonlinear search algorithm, designed to break the model’s implications actively (active nonlinear tests

(ANTs) [22]). BehaviorSearch is a software tool (included in the latest Netlogo versions) to help automate the exploration of agent-based models (ABMs) by using genetic algorithms and other heuristic techniques to search the parameter space [26].

We aim to explore the necessary reflections in the simulation structure and thereby begin to approach complexity. So, we configure the tool and search in the CPI parameter space to identify the max fitness of employees-number, assigned-tasks-employee, probability-of-delay, and probability-of-advance combinations using the 2100 tests' results dataset (we want to maximize the CPI-related space parameters that influence the project performance). In the same way, to compare, we configure the tool and search in the "step" parameter space to identify the min fitness of employees-number, assigned-tasks-employee, probability-of-delay, and probability-of-advance combinations (we want to minimize the "step"-related space parameters that influence the project duration). Table 9 shows an assortment of fitnesses in the search parameter space related to "CPI" in comparison with Table 10, which shows a similar fitness in the search parameter space related to "step" (project duration).

Table 9. "CPI" active nonlinear tests final bests fitness.

Search-Number	Evaluation	Employees-Number	Assigned-Tasks-Employee	Probability-of-Delay	Probability-of-Advance	Num-Replicates	Best-Fitness-so-Far
1	500	6	2	0.2	0.9	10	8.125806944
2	500	5	1	0.2	0.9	10	7.92122959
3	500	3	1	0	0.6	10	2.571417207
4	500	3	2	0	0.9	10	10.18728451
5	500	4	1	0	0.9	10	10.35474418
6	500	1	3	0.4	0.9	10	5.902861442
7	500	1	2	0	0.8	10	5.015753468
8	500	7	3	0.3	0.9	10	6.719011184
9	500	5	3	0	0.9	10	10.22730573
10	500	3	1	0.3	0.9	10	6.764397903

Table 10. "step" active nonlinear tests final bests fitness.

Search-Number	Evaluation	Employees-Number	Assigned-Tasks-Employee	Probability-of-Delay	Probability-of-Advance	Num-Replicates	Best-Fitness-so-Far
1	500	7	2	0	0.7	10	163
2	500	6	2	0.2	0.1	10	223.6
3	500	4	3	0	0.9	10	180
4	500	5	2	0	0.6	10	198
5	500	7	3	0	0.1	10	146
6	500	6	3	0	0.2	10	155
7	500	6	3	0	0.3	10	155
8	500	5	3	0	0.1	10	163
9	500	6	2	0.1	0.7	10	197.5
10	490	6	3	0	0.5	10	155

The model could describe the project duration linearly, but the CPI shows uncertain behavior. In the case of the search parameter space related to "CPI," the model is very predictable when the probability of delay and advance is close to 0. However, when close to 1, the sequence and time of execution could vary away from the estimation. We consider complexity to hide behind the tasks executed by agents that express a probability of delay or advance in an active project. In other words, the project execution could leave us in a different final stage, starting from the same initial project parameter values that are a feature of complex systems behavior.

5. Discussion

The objective of this research was to create an agent-based model that allows the exploration of different explanation alternatives to specific problems in agile project management through earned value management. Therefore, we presented a model of EVM where employees work on a task backlog in a characteristic project execution process to approach the agile development process. The to-do jobs are visually represented in a typical task board to show how the task path to completion happens. At this level of representation, the results show that the model behaves as expected: the model simulates the employees attending tasks, and the EVM metrics show the assessment.

Further to this first approach, studies related to the dynamism of project management, which seek to explain behavior and results using the fundamentals of complexity theory, are becoming more frequent [27,28]. In this context, project management gains importance among the complex sciences by studying the relevant variables involved [27,28]. Regarding the multidisciplinary character of project management, research in innovation and technology management that considers the different theoretical frameworks is perhaps the most influential emerging discipline [29,30].

So, to overcome the limitations of traditional project management, the cultural agency theory would allow the representation of the internal and external factors involved during the development of stakeholder scenarios [31]. This theory's holistic perspective considers the cultural, personality, and operational systems. In a business context, the cultural system integrates values and beliefs (knowledge management and market orientation); the personality system considers cognitive capabilities (goals, ideology, self-schema); and the operating system integrates structural components (operational performance and self-organization) [10].

Therefore, we could go beyond a simple system design where the EVM performance result could hide the causes linearly [32]. So, we could represent the EVM as an operational subsystem according to Yolles's cultural agency theory [33–35] in a complex system context. From this point of view, the EVM agency could establish the gameplay rules for the other agents in the system that constrains or motivates their behavior. Within these conditions, other stakeholder agents should negotiate and develop agreements to self-organize and accomplish their goals.

How do the agents' conditions affect the design of complex production systems? As a result of our experience modeling EVM and operating different scenario simulations, we observed that EVM, as an agency in a complex production system, concerns the operative game rules where other agencies should persist. In this circumstance, the play rules determine the other agents' behavior (for example, employees), execute assigned jobs, and earn value for the project following the production constraint. So, different initial conditions pre-determine the whole system's behavior; thus, making real-time corrections would help the project to succeed.

Beyond this embryonic project management representation, we consider that there are several advantages to using this prototype for more elaborated modeling:

1. We provide a simulation tool to explore the relationship between task planned and performance conditions and the effect in the EVM metrics observations. Additionally, the model shows a typical task board tool to visualize the job backlog processing as most managers used to. This experimentation could help EVM learners and managers explore scenarios to understand how the metrics perform in different conditions.
2. The model is inspired by agency theory, specifically by Yolles's cultural agency theory. Under this theoretical perspective, the model could have sense in the rationale of complexity. As the theory proposed, we can consider new features to add individual behavior and cultural factors.
3. We defined the model according to the ODD protocol. The ODD is a protocol recommended by the social simulation scientific community to overview the model and describe design concepts and implementation details to communicate agent-based models.
4. We programmed an agent-based model in a freely available tool. Netlogo is friendly for unskilled programmers and easily adaptable for new purposes.
5. The PMI considers the EVM a standard in project management.

However, we consider that the most significant weaknesses of this proposed model are as follows:

1. The tool has limitations to building high-performance simulations.
2. The implicit systematic EVM limitations to assess other aspects of agile development management.

3. It is limited to the execution and control processes of the tasks where the promoting and executing agents have direct participation.

Nevertheless, the current proposed model may only be able to answer some of the questions it could raise, and future expansion of the model could prove helpful.

6. Conclusions

The proposed model is a valuable tool for quantifying the operating system in project management. In particular, it makes it possible to quantify earned value management. Future research could propose a model that considers the sequentially of tasks, the organization of these tasks in work subteams, and the inclusion of the underlying systems of the cultural agency theory: the cultural system and the personality system [10]. In the cultural system, variables could be included at the organizational level (practices, corporate policies, and managerial leadership), and in the personality system, variables at the team level would be included (skills, coordination, cooperation, communication, cognition, leadership, and internal conditions) [36].

Author Contributions: Conceptualization, M.C.-P., R.F.R.-C. and J.C.A.-P.; methodology, M.C.-P.; software, M.C.-P.; validation, M.C.-P., R.F.R.-C. and J.C.A.-P.; formal analysis, M.C.-P.; investigation, M.C.-P., R.F.R.-C. and J.C.A.-P.; resources, M.C.-P. and R.F.R.-C.; data curation, M.C.-P.; writing—original draft preparation, M.C.-P., J.C.A.-P., C.K. and E.A.-C.; writing—review and editing, J.C.A.-P., E.A.-C. C.K. and A.T.-R.; supervision, A.T.-R.; project administration, M.C.-P.; funding acquisition, M.C.-P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the internal research project fund of the Autonomous University of Baja California. Project registry: 300/6/C/11/22.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We want to thank the Complex Systems Laboratory at the School of Chemistry and Engineering and Research Center of Complexity Studies at the School of Accounting and Administration, Universidad Autónoma de Baja California and the Biomedical Data Science Research Software Laboratory at the Geisel School of Medicine at Dartmouth, Dartmouth College for their support of this project.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AC	Actual Cost
ABM	Agent-Based Model
CPI	Cost Performance Index
EV	Earned Value
EVM	Earned Value Management
ODD	“Overview, Design Concepts, Details” protocol
PV	Planned Value
PM	Project Management
PMB	Performance Measurement Baseline
PMBOOK	Project Management Body of Knowledge
PMI	Project Management Institute
SPI	Scheduled Performance Index
WBS	Work Breakdown Structure

Appendix A. The Earned Value Management Model

In this appendix, we described the earned value management [7] model according to the ODD [17–19]. We followed “S1: ODD Guidance and Checklists” for guidance and checklists for writing ODD descriptions of simulation models, based on the ODD version published in [17].

Appendix A.1. Overview

Appendix A.1.1. Purpose and Patterns

This model illustrates how EVM provides an approach to measure a project’s performance. Our model’s instance performs a project execution and calculates the EVM performance indexes according to a performance measurement baseline (PMB), as detailed below.

Appendix A.1.2. Entities, State Variables, and Scales

Entities

We include the following entities in the model: agents representing employees (i.e., developers, architects, stakeholders, etc.), tasks, and the global environment representing the workspace (i.e., physical or virtual spaces).

The following entities are included in the model:

1. The employee-agent, representing the developers (i.e., team leaders, team members, architects, and stakeholders);
2. The task-agent, representing the tasks (i.e., the work breakdown structure and tasks);
3. The employee-task-link, representing the employee-task assignments (i.e., the tasks backlog);
4. The global environment, representing the task board and the workspace (i.e., the Kan-ban board).

State Variables

An *observer* is an individual that commands global variables and submodels. Therefore, *observer* state variables are global variables that may alter over time. In Table A1 we show the entities’ state variables.

Table A1. Entities’ state variables

Entity	Variable Name	Variable Type	Meaning
Task	status	Integer	The task status
	task-number	Integer	The task number
	task-description	String	The task description
	priority	Integer	The task priority
	planned-start	String	A planned task start date
	planned-finish	String	A planned task finish date
	planned-hours	Integer	Planned task execution hours
	complete-hours	Integer	Complete task execution planned hours
Employee	actual-hours	Integer	Real/actual task execution hours
	employee-number	Integer	The employee ID number
	status	Integer	The employee status
	role	String	The employee role

Scales

Our model’s temporal scale is set as hours because for project duration we often counted working hours. So, a tick in this agent-based model (ABM) means an hour. We set up the simulation time as long as the work breakdown structure (WBS) requires because the

term of most projects is different, and simulating according to the backlog retrieved from the WBS can adequately contain the usual operations of a short project-based organization.

In Table A2, we show the environmental scales.

Table A2. Scales.

Scale	Values	Meaning
Grid	16 × 32	The task board and color tags.
Grid	16 × 32	The workspace and employees.
Ticks	0–n	The working hours

Appendix A.1.3. Process Overview and Scheduling

First, we create a random task backlog according to the maximum number of jobs specified in the initial configuration. We also indicate how many workers will form the work team. Finally, we indicate how many tasks we will delay and how many will be advanced.

The workers then process the tasks. First, each worker chooses tasks from the backlog and moves the task to the in-progress column. Tasks can last in this state depending on the time specified in each task. We could delay some tasks or complete them early. Eventually, the tasks are tagged again with a done mark when the employee entirely performs them.

We continue processing the tasks in a loop until all jobs in the backlog have passed the done state on the board.

Figure A1 shows the process of executing the tasks.

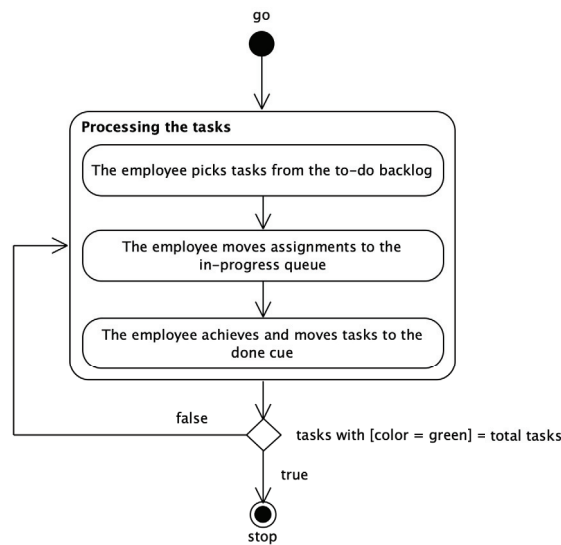


Figure A1. The employee processing the chosen tasks.

Each tick represents a unit of time in the schedule. Each task has an estimated time to complete and a real completed time.

In this model, tasks have no predecessors, and the hourly cost is the tick cost. So, the project’s total cost is equal to the total sum of the planned hours of the tasks or the total sum of ticks.

Appendix A.2. Design Concepts

Appendix A.2.1. Basic Principles

This model addresses a classic problem of project management (PM). This problem involves the risk of delay in execution and cost and schedule estimation failure. There is an extensive literature on earned value management to handle project behavior, mainly founded on cost levels and performance metrics. Our model executes a task board with

workers assigned to complete a task backlog, where workers may delay or advance task execution. We calculate performance using the *earned value management* approach, basing our model design on five fundamental ideas:

- A task backlog: a task backlog (to-do column) requires individuals to complete it.
- A task board: task states are portrayed on a task board to visualize the project's advancement.
- Players: players must take as many tasks as permitted from the "to-do" queue and deliver them to the "done" cue in the panel. While a player is working on an assignment, he must keep the assignment tag in the "in-progress" column.
- A cost and schedule: the task has a planned cost in hours and start–finish time, but the worker could delay or advance in completing the job, or environmental situations could increase and decrease the final cost.
- Performance metrics: the earned value management metrics estimate the project performance.

Appendix A.2.2. Emergence

The key outcomes of the model are earned value management impacts—mainly how suitable the entire system is; these outcomes emerge from how the task executions respond to delays and advance probabilities in tasks, backlog size, players number, and tasks assigned per person.

Appendix A.2.3. Adaptation

The project management behavior of employee agents is to re-estimate the task cost or schedule: the employee characterizes the decision to reduce or increase the actual hours (actual cost) in contrast with planned hours (planned cost) by the probability of affecting each task. Each decision (conscious or unconscious, rational or emotional) directly impacts the project performance (cost or schedule performance).

Appendix A.2.4. Objectives

The objective measure used by project managers to decide whether to take course-correcting action on a project is the cost–schedule performance ratio. Workers reduce their chances of failing to perform or estimate a task if they are motivated. However, the project manager can take analytical actions, such as increasing the number of workers, the number of assignments per person, etc. The project course will immediately reflect any manager's activity on the fly in the earned value management metrics observation.

Appendix A.2.5. Prediction

The project managers can observe project course predictions by cost and schedule to finish estimations beyond the cost–schedule performance ratio. For example, earned value management metrics figure the cost performance index at conclusion (CPIAC) or time estimate at completion (EACT) to help managers to have a future idea about the project.

Appendix A.2.6. Stochasticity

We used stochasticity in two ways. First, we initialize the model stochastically to establish the planned cost and duration task randomly. These initialization methods are stochastic so that the model can be assumed unsegregated at the start of a simulation and that each model run produces different results. Second, when an employee decides to delay or advance in task execution, its choice of the new cost or duration is stochastic. The latest actual cost of finish when the employee performs is stochastic because modeling the details of the decision is unnecessary for this model.

Appendix A.2.7. Collectives

Our model encompass two types of collective groups of tasks that affect the employees and are likewise powerfully affected by the individuals. Such groups are represented as model entities, with state variables and behaviors. These task and employee group entities have their state variables defined above at entities, state variables, and scales, naturally. Our model includes these groups due to employees having several cooperative behaviors, making decisions critical to the project's performance that depend on their collective choices. Tasks may clearly have diverse connections, establishing key constraints to the project's performance. We have found that it is much easier to model cooperative behavior and linkage conditions as *collective entity behaviors* than *individual entity behaviors*.

Appendix A.2.8. Observation

The model aims to study how potential management alternatives affect project behavior. One measure of simulated project management is the probability of failure within certain conditions. We can estimate this probability of failure as the fraction of replicate simulations in which employees never completed some task at the end. Arbitrary observation decisions are how many tasks or workers or how long are the delays that we execute. Here, we estimate the project performance as the fraction of 100 replicate simulations with a probability of high cost and schedule delays so high that the performance index is so low that the project never ends.

Appendix A.3. Details

Appendix A.3.1. Initialization

We initialize the state variable of each individual (planned-hours, probability-of-delay, probability-of-advance, etc.) from probability distributions that describe its variability. We randomly select the estimated scheduled hours from the following set of possible values: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, and 233. The values match the first ten numbers in the series of Fibonacci, which mimics an agile Fibonacci estimation (AFE) method. AFE refers to a way of quantifying the effort needed to complete a development task.

Appendix A.3.2. Input Data

In this model, we do not use input data files from external sources by default (tasks and assignments to employees). Instead, we generate observer-predetermined task sets with random estimates for each simulation. But the model has an example of loading data from a file. The data file could be a set of tasks from an existing or fictitious source in an excel file in CSV format. In Table A3 we show the initialization setup variables.

Table A3. Setup variables.

Input Variable	Data Type	Values
employees-number	Integer	0–100
number-of-tasks	Integer	1–n
probability-of-delay	Integer	0–1
probability-of-advance	Integer	0–1
assigned-tasks-employee	Integer	0–3

Appendix A.3.3. Submodels

Earned Value Management

In earned value management, unlike in traditional management, there are three data sources: planned value (PV), earned value (EV), and actual cost (AC). Figure A2 shows the graphic performance report and Table A4 shows the metrics description and calculations.

The PV is the budget (or planned) value of work scheduled, the EV is the “earned value” of the physical work completed, and the AC is the actual value of work achieved.

The tasks state determines PV, EV, and AC values and is the core of EVM performance indexes and estimations.

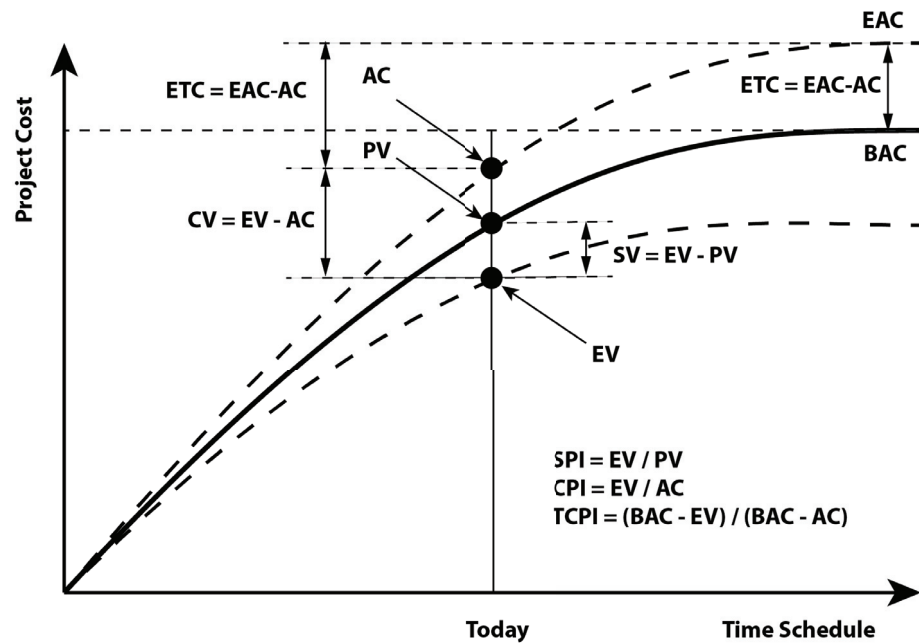


Figure A2. The earned value management (EVM) graphic performance report.

Table A4. Earned value management metrics description.

EVM Metric	Calculation and Description
Planned Value, PV	The budget (or planned) value of work scheduled
Earned Value, EV	The "earned value" of the physical work completed
Actual Cost (AC)	The actual value of work completed
Budget at Completion, BAC	$PV\% = PV / BAC$ $EV\% = EV / BAC$ $AC\% = AC / BAC$
Schedule Variance, SV	$SV = EV - PV$ $SV\% = SV / PV$
Cost Variance, CV	$CV = EV - AC$ $CV\% = CV / EV$
Schedule Performance Index, SPI	$SPI = EV / PV$
Cost Performance Index, CPI	$CPI = EV / AC$
To Complete Performance Index, TCPI	$TCPI = (BAC - EV) / (BAC - AC)$
Estimate at Completion, EAC	$EAC = BAC - SV$ $EAC = BAC / CPI$ $EAC = BAC / (CPI * SPI)$
Estimate to Complete, ETC	$EAC = AC + \text{new estimate of remaining work}$
Variance at Completion, VAC	$ETC = EAC - AC$ $VAC = BAC - EAC$ $VAC\% = VAC / BAC$
Cost Performance Index at Conclusion, CPIAC	$CPIAC = BAC / EAC$
Time Estimate at Completion, EACt	$EACt = (BAC / SPI) / (BAC / PMB \text{ Duration}) = PMB \text{ duration} / SPI$
Time Variance at Completion, VACT	$VACT = PMB \text{ duration} - EACt$
Time Schedule Performance Index at Conclusion, SPIACt	$VACT\% = VACT / PMB \text{ duration}$ $SPIACt = PMB \text{ duration} / EACt$

Appendix B. Sensitivity Assessment

Table A5. Data description.

Variable	Mean	SD	Median	MAD	Min	Max	n
employees.number	4	2.00047636061173	4	2.9652	1	7	2100
assigned.tasks.employee	2	0.81669105433311	2	1.4826	1	3	2100
probability.of.delay	0.45	0.287296544411313	0.45	0.37065	0	0.9	2100
probability.of.advance	0.45	0.287296544411313	0.45	0.37065	0	0.9	2100
step	1116.98571428571	1603.01591793252	581	471.4668	146	15,980	2100
AC	2469.54904761905	2817.20670248899	1532	1245.384	125	15,939	2100
PV	1532	0	1532	0	1532	1532	2100
EV	1532	0	1532	0	1532	1532	2100
SV	0	0	0	0	0	0	2100
SPI	1	0	1	0	1	1	2100
CV	-937.549047619048	2817.20670248899	0	1245.384	-14,407	1407	2100
CPI	1.61839999350462	1.86235517369687	1	0.814674377613205	0.0961164439425309	12.256	2100

Table A6. Requirements verification.

Requirement	Specification	Number of Traces Where Requirement Is True	Total Number of Traces	Percent of Cases Where the Requirement Is True out of Total Cases	Assessment
employees.number >= 1	Always True	2100	2100	1	Requirement Is Met in ALL cases
employees.number <= 7	Always True	2100	2100	1	Requirement is Met in ALL cases
assigned.tasks.employee >= 1	Always True	2100	2100	1	Requirement is Met in ALL cases
assigned.tasks.employee <= 3	Always True	2100	2100	1	Requirement is Met in ALL cases
probability.of.delay >= 0	Always True	2100	2100	1	Requirement is Met in ALL cases
probability.of.delay <1	Always True	2100	2100	1	Requirement is Met in ALL cases
probability.of.advance >= 0	Always True	2100	2100	1	Requirement is Met in ALL cases
probability.of.advance <1	Always True	2100	2100	1	Requirement is Met in ALL cases

Table A7. Percent of "CPI" Cases within Range 0.0961164439425309 to 3.5 = 89.5238095238095, n = 1880.

Condition	Number of Traces Where Condition Is True	Total Number of Traces	Likelihood That Condition Appears Alongside "CPI" within Range 0.0961164439425309 to 3.5	Likelihood That "CPI" within Range 0.0961164439425309 to 3.5 Contains the Condition	Sensitivity Assessment
employees.number >= 0	1880	2100	0.895238095238095	1	0.944723618090452
assigned.tasks.employee >= 0	1880	2100	0.895238095238095	1	0.944723618090452
probability.of.delay >= 0	1880	2100	0.895238095238095	1	0.944723618090452
probability.of.advance >= 0	1880	2100	0.895238095238095	1	0.944723618090452
employees.number >0	1880	2100	0.895238095238095	1	0.944723618090452
assigned.tasks.employee >0	1880	2100	0.895238095238095	1	0.944723618090452
probability.of.delay >0	1713	1890	0.906349206349206	0.911170212765957	0.908753315649867
probability.of.advance >0	1670	1890	0.883597883597884	0.888297872340426	0.885941644562334
employees.number == 0	0	0	NA	0	NA
assigned.tasks.employee == 0	0	0	NA	0	NA
probability.of.delay == 0	167	210	0.795238095238095	0.0888297872340426	0.159808612440191
probability.of.advance == 0	210	210	1	0.111702127659574	0.200956937799043
employees.number <0	0	0	NA	0	NA
assigned.tasks.employee <0	0	0	NA	0	NA
probability.of.delay <0	0	0	NA	0	NA
probability.of.advance <0	0	0	NA	0	NA
employees.number <= 0	0	0	NA	0	NA
assigned.tasks.employee <= 0	0	0	NA	0	NA
probability.of.delay <= 0	167	210	0.795238095238095	0.0888297872340426	0.159808612440191
probability.of.advance <= 0	210	210	1	0.111702127659574	0.200956937799043

Table A8. Sensitivity Assessment. Percent of “step” Cases within Range 146 to 2720 = 92.0476190476191, n = 1933.

Condition	Number of Traces Where Condition Is True	Total Number of Traces	Likelihood That Condition Appears Alongside “Step” within Range 146 to 2720	Likelihood That “Step” within Range 146 to 2720 Contains the Condition	Sensitivity Assessment
employees.number >= 0	1933	2100	0.92047619047619	1	0.958591619142078
assigned.tasks.employee >= 0	1933	2100	0.92047619047619	1	0.958591619142078
probability.of.delay >= 0	1933	2100	0.92047619047619	1	0.958591619142078
probability.of.advance >= 0	1933	2100	0.92047619047619	1	0.958591619142078
employees.number >0	1933	2100	0.92047619047619	1	0.958591619142078
assigned.tasks.employee >0	1933	2100	0.92047619047619	1	0.958591619142078
probability.of.delay >0	1723	1890	0.911640211640212	0.891360579410243	0.901386345801726
probability.of.advance >0	1741	1890	0.921164021164021	0.900672529746508	0.910803034266283
employees.number == 0	0	0	NA	0	NA
assigned.tasks.employee == 0	0	0	NA	0	NA
probability.of.delay == 0	210	210	1	0.108639420589757	0.195986934204386
probability.of.advance == 0	192	210	0.914285714285714	0.099327470253492	0.179188054129725
employees.number <0	0	0	NA	0	NA
assigned.tasks.employee <0	0	0	NA	0	NA
probability.of.delay <0	0	0	NA	0	NA
probability.of.advance <0	0	0	NA	0	NA
employees.number <= 0	0	0	NA	0	NA
assigned.tasks.employee <= 0	0	0	NA	0	NA
probability.of.delay <= 0	210	210	1	0.108639420589757	0.195986934204386
probability.of.advance <= 0	192	210	0.914285714285714	0.099327470253492	0.179188054129725

References

- Noteboom, C.; Ofori, M.; Sutrave, K.; El-Gayar, O. Agile Project Management: A Systematic Literature Review of Adoption Drivers and Critical Success Factors. In Proceedings of the Hawaii International Conference on System Sciences, Maui, HI, USA, 5–8 January 2021; pp. 6775–6783. [\[CrossRef\]](#)
- Institute, P.M. *The Standard for Project Management and a Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 17th ed.; Project Management Institute: Harrisburg, PA, USA, 2021.
- PMI. *Agile Practice Guide PMBOK*, 1st ed.; Project Management Institute, Inc.: Newtown Square, PA, USA, 2017; p. 130.
- Briggs, S. Agile Based Learning: What Is It and How Can It Change Education? 2014. *InformED*. 2019. Available online: <http://www.opencolleges.edu.au/informed/features/agile-based-learning-what-is-it-and-how-can-it-change-education/> (accessed on 31 August 2019).
- Brandl, F.J.; Roider, N.; Hehl, M.; Reinhart, G. Selecting practices in complex technical planning projects: A pathway for tailoring agile project management into the manufacturing industry. *CIRP J. Manuf. Sci. Technol.* **2021**, *33*, 293–305. [\[CrossRef\]](#)
- Medini, K. A framework for agility improvement projects in the post mass customisation era. *Int. J. Prod. Res.* **2022**, *2022*, 1–17. [\[CrossRef\]](#)
- The Standard for Earned Value Management*; Project Management Institute: Harrisburg, PA, USA, 2019.
- Fleming, Q.W.; Koppelman, J.M. *Earned Value Project Management*; Project Management Institute: Harrisburg, PA, USA, 2016, p. 231.
- Papatheocharous, E.; Andreou, A.S. Empirical evidence and state of practice of software agile teams. *J. Softw. Evol. Process* **2014**, *26*, 855–866. [\[CrossRef\]](#)
- U-tantada, S.; Yolles, M.; Shoosanuk, A.; Mujtaba, B.G. Strategic SMEs as Influential Drivers for the Market Economy: Modeling SMEs using Cultural Agency Theory. *Int. Rev. Res. Emerg. Mark. Glob. Econ.* **2019**, *5*, 1435–1452.
- Samset, K.; Volden, G.H. Front-end definition of projects: Ten paradoxes and some reflections regarding project management and project governance. *Int. J. Proj. Manag.* **2016**, *34*, 297–313. [\[CrossRef\]](#)
- Collyer, S.; Warren, C.M. Project management approaches for dynamic environments. *Int. J. Proj. Manag.* **2009**, *27*, 355–364. [\[CrossRef\]](#)
- Díez-Silva, H.M.; Pérez-Ezcurdia, M.A.; Gimena Ramos, F.N.; Montes-Guerra, M.I. Metodología de la Gestión Orientada a los Agentes del Proyecto. In Proceedings of the XVI Congreso Internacional de Ingeniería de Proyectos, Valencia, Spain, 11–13 July 2012.
- Hansen, E.G.; Schaltegger, S. Sustainability Balanced Scorecards and their Architectures: Irrelevant or Misunderstood? *J. Bus. Ethics* **2018**, *150*, 937–952. [\[CrossRef\]](#)
- Gilbert, N.; Troitzsch, K.G. *Simulation for the Social Scientist*, 2nd ed.; Open University Press: Glasgow, UK, 2005; p. 295.
- Garro, A.; Russo, W. easyABMS: A domain-expert oriented methodology for agent-based modeling and simulation. *Simul. Model. Pract. Theory* **2010**, *18*, 1453–1467. [\[CrossRef\]](#)
- Grimm, V.; Railsback, S.F.; Vincenot, C.E.; Berger, U.; Gallagher, C.; DeAngelis, D.L.; Edmonds, B.; Ge, J.; Giske, J.; Groeneveld, J.; et al. The ODD Protocol for Describing Agent-Based and Other Simulation Models: A Second Update to Improve Clarity, Replication, and Structural Realism. *J. Artif. Soc. Soc. Simul.* **2020**, *23*, 7. [\[CrossRef\]](#)

18. Grimm, V.; Berger, U.; Bastiansen, F.; Eliassen, S.; Ginot, V.; Giske, J.; Goss-Custard, J.; Grand, T.; Heinz, S.K.; Huse, G.; et al. A standard protocol for describing individual-based and agent-based models. *Ecol. Model.* **2006**, *198*, 115–126. [[CrossRef](#)]
19. Grimm, V.; Berger, U.; DeAngelis, D.L.; Polhill, J.G.; Giske, J.; Railsback, S.F. The ODD protocol: A review and first update. *Ecol. Model.* **2010**, *221*, 2760–2768. [[CrossRef](#)]
20. Jordan, A. The Earned Value Management Calculator. Available online: <https://www.projectmanagement.com/deliverables/284670/Earned-Value-Management--EVM--Calculator> (accessed on 27 December 2022).
21. Palacios, S. EVM Worksheet Package. Available online: <https://www.projectmanagement.com/deliverables/544342/evm-worksheet-package> (accessed on 10 December 2022).
22. Miller, J.H. Active Nonlinear Tests (ANTs) of Complex Simulation Models. *Manag. Sci.* **1998**, *44*, 820–830. [[CrossRef](#)]
23. Tissue, S.; Wilensky, U. NetLogo: A Simple Environment for Modeling Complexity. In Proceedings of the International Conference on Complex Systems, Boston, MA, USA, 16–21 May 2004; Volume 21, pp. 16–21.
24. Castañón-Puga, M.; Rosales-Cisneros, R.F.; Ahumada-Tello, E. NetLogo Task Board (1.0.0). Zenodo, 2022. Available online: <https://zenodo.org/record/7474614#Y-HSrK1ByUk> (accessed on 27 December 2022).
25. Castañón-Puga, M.; Rosales-Cisneros, R.F.; Acosta-Prado, J.C.; Tirado-Ramos, A.; Khatchikian, C.; Aburto-Camaclanqui, E. Netlogo Earned Value Management Model. Available online: <https://www.comses.net/codebases/4effb542-f69f-464d-9f96-69f19d36ae8a/> (accessed on 27 December 2022).
26. BehaviorSearch. Available online: <http://www.behaviorsearch.org> (accessed on 10 December 2022).
27. Singh, H.; Singh, A. Principles of complexity and chaos theory in project execution: A new approach to management. *Cost Eng.* **2002**, *44*, 23–33.
28. Stacey, R.D. The Science of Complexity: An Alternative Perspective for Strategic Change Processes. *Strateg. Manag. J.* **1995**, *16*, 477–495. [[CrossRef](#)]
29. Bredillet, C.N. PMI Research and Education Conference 2010. *Proj. Manag. J.* **2010**, *41*, 2–3. [[CrossRef](#)]
30. Ika, L.A. Project success as a topic in project management journals. *Proj. Manag. J.* **2009**, *40*, 6–19. [[CrossRef](#)]
31. U-tantada, S.; Yolles, M.; Shoosanuk, A.; Mujtaba, B.G. Diagnosing Viable SME Cultural Agency and Capabilities. In Proceedings of the 1st Humanities and Social Sciences Research Promotion Network National and International Conference, Chanthaburi, Thailand, 28–29 June 2018.
32. Suarez, E.D.; Castañón-Puga, M. Distributed Agency. *Int. J. Agent Technol. Syst.* **2013**, *5*, 32–52. [[CrossRef](#)]
33. Yolles, M. *Management Systems: A Viable Approach*; Financial Times Pitman: London, UK, 1999.
34. Yolles, M. *Organizations as Complex Systems: An Introduction to Knowledge Cybernetics*; Information Age Publishing: Creenwich, CO, USA, 2006; Volume 2.
35. Yolles, M. Governance through political bureaucracy: An agency approach. *Kybernetes* **2019**, *48*, 7–34. [[CrossRef](#)]
36. Tannenbaum, S.; Salas, E. *Teams That Work: The Seven Drivers of Team Effectiveness*; Oxford University Press: New York, NY, USA, 2021.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Deep Reinforcement Learning Reward Function Design for Autonomous Driving in Lane-Free Traffic

Athanasia Karalakou ^{1,†}, Dimitrios Troullinos ^{2,*}, Georgios Chalkiadakis ¹ and Markos Papageorgiou ²¹ School of Electrical and Computer Engineering, Technical University of Crete, 73100 Chania, Greece² School of Production Engineering and Management, Technical University of Crete, 73100 Chania, Greece

* Correspondence: dtroullinos@tuc.gr

† These authors contributed equally to this work.

Abstract: Lane-free traffic is a novel research domain, in which vehicles no longer adhere to the notion of lanes, and consider the whole lateral space within the road boundaries. This constitutes an entirely different problem domain for autonomous driving compared to lane-based traffic, as there is no leader vehicle or lane-changing operation. Therefore, the observations of the vehicles need to properly accommodate the lane-free environment without carrying over bias from lane-based approaches. The recent successes of deep reinforcement learning (DRL) for lane-based approaches, along with emerging work for lane-free traffic environments, render DRL for lane-free traffic an interesting endeavor to investigate. In this paper, we provide an extensive look at the DRL formulation, focusing on the reward function of a lane-free autonomous driving agent. Our main interest is designing an effective reward function, as the reward model is crucial in determining the overall efficiency of the resulting policy. Specifically, we construct different components of reward functions tied to the environment at various levels of information. Then, we combine and collate the aforementioned components, and focus on attaining a reward function that results in a policy that manages to both reduce the collisions among vehicles and address their requirement of maintaining a desired speed. Additionally, we employ two popular DRL algorithms—namely, deep Q-networks (enhanced with some commonly used extensions), and deep deterministic policy gradient (DDPG), which results in better policies. Our experiments provide a thorough investigative study on the effectiveness of different combinations among the various reward components we propose, and confirm that our DRL-employing autonomous vehicle is able to gradually learn effective policies in environments with varying levels of difficulty, especially when all of the proposed rewards components are properly combined.

Keywords: deep reinforcement learning; lane-free traffic; autonomous driving

Citation: Karalakou A.; Troullinos, D.; Chalkiadakis, G.; Papageorgiou M. Deep Reinforcement Learning Reward Function Design for Autonomous Driving in Lane-Free Traffic. *Systems* **2023**, *11*, 134. <https://doi.org/10.3390/systems11030134>

Academic Editors: Philippe Mathieu, Juan M. Corchado, Alfonso González-Briones and Fernando De la Prieta Pintado

Received: 30 January 2023
Revised: 25 February 2023
Accepted: 28 February 2023
Published: 2 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Applications of reinforcement learning (RL) in the field of autonomous driving have been gaining momentum in recent years [1] due to advancements in Deep RL [2,3], giving rise to novel techniques [4]. The fact that deep reinforcement learning (DRL) can handle high dimensional state and action spaces makes it suitable for controlling autonomous vehicles. Another important reason for this momentum is increasing interest in autonomous vehicles (AVs), as the current and projected technological advancements in the automotive industry can enable such methodologies in the real world [5,6]. As a result, novel traffic flow research endeavors have already emerged, such as TrafficFluid [7], which primarily targets traffic environments with 100% penetration of AVs (no human drivers). Trafficfluid examines traffic environments with two fundamental principles:

- (i) Lane-free vehicle movement, meaning that AVs under this paradigm do not consider lane-keeping, but are rather *free* to be located anywhere laterally. Lanes emerged to simplify human driving, and when only automated vehicles exist, they may no longer be required, given the observational capabilities of an AV compared to that of a human driver. The renunciation of the lane principle enables higher utilization of the available road capacity, and the lane-changing task is an operation that is no longer needed. AVs' lateral positions can be regulated gradually in order to overtake, and in this manner, an AV can also accommodate upstream traffic via the second principle examined under the lane-free paradigm—that is,
- (ii) Nudging, where vehicles may adjust their behavior so as to assist vehicles on the back that attempt to overtake. In lane-based traffic, this would involve a lane-change operation, and as such, nudging could not actually be considered in conventional traffic. In contrast, in lane-free settings, a vehicle can provide the necessary space for receding vehicles just by leaning towards an appropriate lateral direction.

In the context of lane-free driving, multiple vehicle-movement strategies have already been proposed [7–10]. To be more specific, existing studies have focused on optimal control methods that involve model predictive control [9]. In addition, the application of a movement strategy based on heuristic rules that incorporates the notion of “forces” [7], along with an extension on this endeavor [11], provides empirical insight on the benefits of lane-free traffic. Moreover, authors of [10] have designed a two dimensional cruise controller that specifically addresses lane-free traffic environments, and the authors of [12] incorporated these controllers in a quite challenging roundabout scenario, and provided an experimental microscopic study. Finally, ref. [8] investigated the utilization of the max-plus algorithm, and constructed a dynamic graph structure of the vehicles, considering communication among vehicles as well.

However, to the best of our knowledge, while there is an abundance of (deep) RL applications for conventional (lane-based) traffic environments [1,4,5], so far only two papers in the literature [13,14] tackle the problem of lane-free traffic with RL techniques. We believe that (deep) RL could be a valuable asset for managing this novel challenging domain, as it is capable of solving complex and multi-dimensional tasks with lower prior knowledge because of its ability to learn different levels of abstractions from data. Furthermore, (deep) RL provides policies that automatically adjust to the environment, and thus there is no need for either a centralized authority or explicit communication among vehicles. The efficiency of RL heavily relies on the design of the Markov decision process, and especially on the reward function guiding the agent's learning. The reward design is crucial and determines the overall efficiency of the resulting policy. This is evident in DRL studies related to AVs [5].

Against this background, in this work we tackle the problem of designing an RL agent that learns a vehicle-movement strategy in lane-free traffic environments. To this end, we establish a Markov decision process (MDP) for lane-free autonomous driving, given a single agent environment populated with other vehicles adopting a rule-based lane-free driving strategy [7], primarily focusing on the design of the reward function. Given the nature of the DRL algorithms, their ability to properly learn only with delayed rewards and obtain a (near-) optimal policy is uncertain, so we propose a set of different reward components, ranging from delayed rewards to more elaborate and therefore, more informative regarding the problem's objectives. The learning objectives in our environment are twofold, as they address: (i) safety, i.e., collision avoidance among vehicles; and (ii) desired speed, i.e., that our agent attempts to maintain a specific speed of choice.

In a nutshell, the main contributions of this work are the following:

- We constructed different components of reward functions specifically tied to a the lane-free traffic environment at various levels of information;
- We then combined and collated the aforementioned components, in order to attain an efficient reward function;

- We experimentally investigated the effectiveness of different combinations of the various reward components we propose, using the deep deterministic policy gradient algorithm;
- We confirmed experimentally that our DRL-employing autonomous vehicle is able to gradually learn effective policies;
- Finally, we conducted a comparative experimental study between several well-established deep Q-network variants, and the deep deterministic policy gradient algorithm.

This paper constitutes an extension of our work in [13], appearing in the Proceedings of the 20th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2022). Several contributions are distinct in this paper, compared to those already in [13], namely:

- We put forward a novel reward component that can guide the vehicle in a lateral placement appropriate for overtaking, along with a related experimental evaluation;
- We showcase experimentally that the inclusion of this new component in the reward function yields superior results, especially for high traffic densities;
- Our experimental evaluation was under more intense traffic conditions than those examined in [13], with various densities of nearby vehicles;
- As mentioned above, we conducted a comparative study between DDPG and DQN variants, the results of which point to DDPG exhibiting superior performance, and as such constituting a suitable choice for training an agent via DRL in this complex, continuous domain.

The rest of this paper is structured as follows: In Section 2, we discuss the relevant background and related work, and in Section 3 we present the proposed approach in detail. Then, in Section 4 we provide a detailed experimental evaluation. Finally, in Section 5 we summarize our study and discuss future work.

2. Background and Related Work

In this section, we discuss the theoretical background of this work and provide further information about related work.

2.1. Markov Decision Process

A Markov decision process (MDP) [15] is a mathematical framework that completely describes the environment in a reinforcement learning problem, and they are fundamental for decision-making problems. An MDP can be defined by the 5-tuple:

$$(S, A, P, R, \gamma)$$

where S describes the state space, A denotes the action space, P describes the transition model, R refers to the reward model, and γ is the discount factor.

2.2. Deep Q-Networks and Extensions

A deep Q-network (DQN) [2] adopts the Q-learning algorithm [16–18] for function approximation using neural networks, utilizes convolutional neural networks to obtain a graphical representation of the input state for an environment, and produces a vector of Q-values associated with each possible action. The concepts of target network and experience replay, formally introduced in [2], are the primary methods that enable the use of deep learning for approximating Q functions, resolving the issues of network stability. The Q-network is updated according to the following loss function:

$$L(\theta_t) = E_{(s_t, a_t, r_t, s_{t+1})} [(y_t^{DQN} - Q(s_t, a_t; \theta_t))^2] \quad (1)$$

where $y_t^{DQN} = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-)$ refers to the Q-network's target value at iteration t , and θ_t and θ^- are the network's parameters at iteration t and at a previous iteration, respectively—the latter being the target network's parameters.

The main idea of experience replay is to store the agent's experiences (the tuples (S_t, A_t, R_t, S_{t+1})) in a buffer. Then, in each training step, a batch of experiences is uniformly sampled from the buffer and fed to the network for training. Experience replay ensures that old experiences are not disregarded in later iterations, and removes the correlations in the data sequences, feeding the network with independent data. This is suitable for DQN, since it is an off-policy method.

A well-known pathology of Q-learning, and consequently of DQN, is overestimation occurring from the use of the max operator in the Bellman equation to compute Q-values. The double deep Q-networks (DDQN) [19] algorithm addresses this issue by decomposing the maximizing operation in the target into action selection and evaluation. The idea behind the DDQN is that of double Q-learning [20], which includes two Q-functions. One function selects the optimal action, and the other estimates the value function. As a consequence, double DQN can offer faster training with better rewards across many domains, as evident by the results in [20]. Specifically, DDQN uses (as in vanilla DQN) the target network as the second action-value function, but it selects the next maximizing action according to the current network, while utilizing the target Q-network to calculate its value. The update is similar to DQN's, but the target y_t^{DQN} is replaced with:

$$y_t^{DDQN} = r_t + \gamma Q(s_{t+1}, \arg \max_{a'} Q(s_{t+1}, a'; \theta_t), \theta^-) \quad (2)$$

Prioritized experience replay (PER) [21] is an additional DQN extension that focuses on prioritizing the experiences that contain more "important" information than others. Each experience is stored with an additional priority value, so that experiences with higher priority have a higher sampling probability and therefore, have the chance to remain longer in the buffer than others. As importance measure, the temporal difference (TD) error [18], can be used. It is expected that if the TD error is high (in absolute value), the agent can learn more from the corresponding experience because the agent behaved better or worse than expected.

The dueling network architecture (DNA) [22] decouples the value and advantage function [23], which leads to improved performance. In detail, it is constructed with two streams, one of them providing an estimate of the value function, and the other stream producing an estimate of the advantage function. There is a common feature learning module which is then separated to produce the two different mentioned outputs that are then combined appropriately in order to compute the state–action value function Q.

2.3. Deep Deterministic Policy Gradient

Deep deterministic policy gradient (DDPG) [24] is an off-policy, actor–critic algorithm that utilizes the deterministic policy gradient (DPG) [25] and the DQN architectures, and it overcomes the restriction of discrete actions.

It uses experience replay, just like a DQN, and target networks to compute the target y in the temporal difference error. To be more specific, target networks are two separate networks which are copies of the actor and critic network and are updated in a slow pace, in order to track the learned networks. The update for the critic is given by the standard DQN update by taking targets y_t^{DDPG} to be:

$$y_t^{DDPG} = r_t + \gamma Q(s_{t+1}, \mu(s_{t+1}; \theta^{\mu-}); \theta^{Q-}) \quad (3)$$

where $Q(s, a; \theta^{Q-})$ and $\mu(s; \theta^{\mu-})$ refer to the target networks for the critic and actor, respectively. The differentiating factor in the critic network architecture, with respect to DQN, is that the network contains the action vector as an input and has a single output approximating the corresponding Q-value. Moreover, the policy (actor network) in DDPG is updated using the sampled policy gradient, given a minibatch of transitions [25]:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a; \theta^Q)|_{s=s_i, a=\mu(s_i; \theta^\mu)} \nabla_{\theta^\mu} \mu(s; \theta^\mu)|_{s=s_i} \quad (4)$$

where $\mu(s; \theta^\mu)$ refers to the actor network with parameters θ^μ , and N is the size of the sampled minibatch from the buffer. DDPG also introduces the notion of soft target updates to improve upon learning stability, meaning that the network parameters of the actor (θ^μ) and critic (θ^Q) are updated in every iteration through a temperature parameter τ as: $\theta^- = \tau\theta + (1 - \tau)\theta^-$, with $\tau \ll 1$.

2.4. Related Work

Under the lane-free traffic paradigm, multiple vehicle-movement strategies [7–10] have already been proposed, with approaches stemming from control theory, optimal control, and multi-agent decision making. In more detail, the authors of [7,11] introduced and then enhanced a lane-free vehicle-movement strategy based on heuristic rules that involve the notion of “forces” being applied to vehicles, in the sense that vehicles “push” one another so as to overtake, or in general to react appropriately. Then, reference [9] introduced a policy for lane-free vehicles based on optimal control methods and utilizing a model predictive control paradigm, where each vehicle optimizes its behavior for a specified future horizon, considering the trajectories of nearby vehicles as well. Furthermore, the authors of [10] designed a two-dimensional cruise controller for lane-free traffic, with more emphasis on control theory. They provided a mathematical framework for the controller design considering a continuous-time dynamical system for the kinetic motion and control of vehicles. Then, the authors of [12] incorporated a variation of this controller that is appropriate for discrete-time dynamical systems. The authors apply this in a challenging lane-free roundabout scenario populated with vehicles following different routing schemes within the roundabout. Finally, the authors of [8] tackled the problem with the use of the max-plus algorithm. The authors constructed a dynamic graph structure of the vehicles, considering communication among vehicles as well, and provided an extension in [26] for a novel dynamic discretization procedure and an updated formulation of the problem.

To the best of our knowledge, besides the conference paper [13] we extend herein, there is only one work in the literature that has tackled lane-free traffic using RL: In [14], the authors introduced a novel vehicle-movement strategy for lane-free environments that also incorporates DDPG for a learning agent in a lane-free environment. This agent learns to maintain an appropriate gap with a single vehicle downstream¹ [14], and other operations are handled by different mechanisms based on the notion of repulsive forces and nudging, similarly to [7]. Notably, the overall implementation differed significantly with respect to our work, since the underlying endeavor was evidently quite distinct. The goal of this work was to investigate the use of RL for the task of lane-free driving, its potential, and limitations in this environment, by exclusively utilizing a relevant and well-studied DRL algorithm. Instead, the RL agent in [14] is part of a system composed of different controlling components. As such, in that case the learning task involves a specific longitudinal operation, where the agent also needs to learn to act in accordance with the other mechanisms.

Finally, there are several papers that report reward-function design for (deep or not, single- or multi-agent) RL techniques for lane-based traffic.

In the related literature, it is common to have a reward function form that contains multiple reward components (tied to different objectives), similarly to our approach. Regarding the total reward function that the agent receives, it is usually a simple weighted sum of the different reward components—whereas we employ a reciprocal form. In [27,28], the authors tackle a single-agent lane-based environment in a multi-lane highway using RL. These two approaches differ significantly in their MDP formulation and the RL algorithm of choice. However, they both contain a reward function comprised of several terms in an additive form. This aspect is apparent in multiagent RL applications as well. For instance, in [29], the authors examine a multiagent roundabout environment using the A3C method with parameter sharing among different learning agents. There, the agents likewise receive a reward as a sum of different reward terms tied to collision avoidance, maintaining an operational speed and terminal conditions. Finally, ref. [30] examines a signalized intersection

environment with multiple vehicles learning to operate using the Q-learning algorithm in order to derive an efficient policy. Again, we observe the summation of different terms that form the total reward value that each agent receives.

3. Our Approach

In this section, we first present in detail the lane-free traffic environment we consider, and then the various features of the MDP formulation—specifically, the state representation, and the action space, and the different components proposed for the reward function design.

3.1. The Lane-Free Traffic Environment

Regarding the training environment, we consider a ring-road traffic scenario populated with numerous automated vehicles applying the lane-free driving behavior, as outlined in [7]. Our agent is an additional vehicle that adopts the proposed MDP formulation (more details in the next subsections), learning a policy through observation of the environment. The agent's observational capabilities include the positions (x, y) and speeds (v_x, v_y) of nearby vehicles, and its own position and speed. Both the position and speed are observed as 2-dimensional vectors, consisting of the associated longitudinal (x axis) and lateral (y axis) values. All the surrounding vehicles share the same dimensions and movement dynamics. Each vehicle initially adopts a random desired speed v_d , within a specified range $([v_{d,min}, v_{d,max}])$. Our agent controls two (continuous) variables, namely, the longitudinal and lateral acceleration values (a_x, a_y) in m/s^2 , and determines the gas/break through a_x , and left/right steering through a_y .

Figure 1 depicts the traffic environment. A highway was used to simulate the scenario for the examined ring road, by having vehicles at the end-point of the highway being placed at the starting point appropriately. Vehicles' observations were adjusted accordingly so that they observed a ring-road; e.g., vehicles towards the end of the highway observed vehicles in front, located after the highway's starting point.

As mentioned, other vehicles follow the lane-free vehicle-movement strategy in [7], which does not involve learning; i.e., other agents follow deterministic behavior with respect to their own surroundings. In addition, we disable the notion of nudging for other vehicles, since when enabled, other vehicles move aside whenever we attempt an overtake maneuver, meaning that the agent would learn a very aggressive driving policy.



Figure 1. Indicative image from the lane-free traffic environment. The DRL agent is marked with green.

3.2. State Space

The state space describes the environment of the agent and must contain adequate information for choosing the appropriate action. Thus, the observations should contain information about both the state of the agent in the environment and the surrounding vehicles. More specifically, in regard to the state of the agent, it was deemed necessary to store its lateral position y , and both its longitudinal and lateral speed v_x, v_y . Considering that the surrounding vehicles comprise the agent's environment, we have to include them as well. In particular, in standard lane-based environments, the state space can be defined in a straightforward manner, as an agent can be trained by utilizing information about the front and back vehicles on its lane and accounting for two more vehicles in each adjacent lane, in case a lane-changing movement is required as well. On the contrary, in a lane-free environment, we do not have such lane structures, and as a consequence, the number of the surrounding vehicles in the two-dimensional space we consider varied.

However, our state needs to include information about only a predefined number of vehicles, since the MDP formulation cannot handle state vectors with varying sizes. As such, only the n closest neighboring vehicles are considered in the state space, and “placeholder” vehicles appearing far away may be included on the occasions wherein the number of vehicles is less than n .

We store information about the speed of the surrounding vehicles, both longitudinal and lateral. Additionally, intending to have a sufficient state representation, we include information regarding the distances of the agent from the cars within the aforementioned perimeter using a range of d meters. In this work, these above-mentioned distances are referred to as dx_j and dy_j and result from the distance of the agent’s center and that of each neighboring vehicle j , as shown in Figure 2. Moreover, the desired speed v_d is also included within the state space. Summarizing, the state information is a vector with the form:

$$s = [y, v_x, v_y, v_d, \cup_{j=1}^n [dx_j, dy_j, v_{x,j}, v_{y,j}]]^t \quad (5)$$

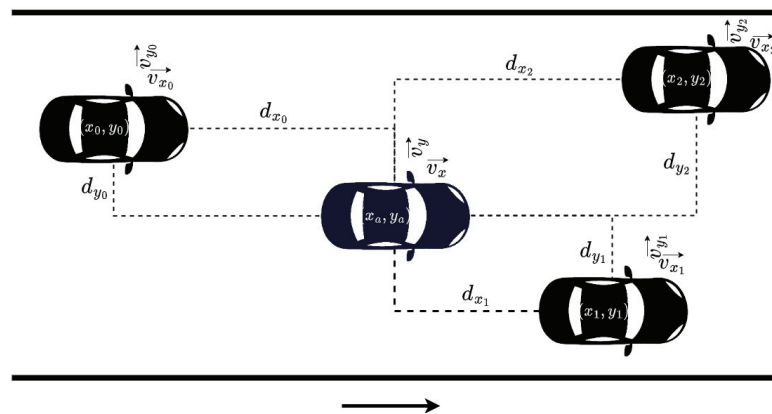


Figure 2. Observed information of surrounding vehicles.

3.3. Action Space

The primary objective of this work was to obtain an optimal policy that generates the appropriate high-level driving behavior for the agent to move efficiently in a lane-free environment. Hence, the suggested action space consisted of two principal actions: one concerning the car’s longitudinal movement by addressing braking and accelerating commands, and one relevant to the lateral movement through acceleration commands for acceleration towards the left or right.

Moreover, in the context of this work, we investigated both a continuous and a discrete action space. Specifically, DQN and its extensions generate state-action values for each action; thus, we had to use an appropriate set of discrete actions. A discrete action space consists of a finite set of distinct actions [18]. In detail, at each time-step t , the agent can perform one of nine possible actions:

- a_0 : zero acceleration in both axes;
- a_1 : longitudinal acceleration;
- a_2 : longitudinal deceleration;
- a_3 : lateral acceleration towards the left;
- a_4 : lateral acceleration towards the right;
- a_5 : combination of a_1 and a_3 ;
- a_6 : combination of a_2 and a_3 ;
- a_7 : combination of a_1 and a_4 ;
- a_8 : combination of a_2 and a_4 .

On the other hand, the DDPG method is developed for environments with continuous action spaces. Therefore, when DDPG is adopted, the action space is a vector $\mathbf{a} \in \mathbb{R}^2$ that controls the vehicle via specifying its longitudinal and lateral acceleration.

3.4. Reward Function Design

The design of the reward function is critical for the performance of (deep) RL algorithms. This is indeed pivotal for enabling DRL controlled autonomous vehicles, since the reward function should represent the desired driving behavior of our agent. It is worth noting that finding an appropriate reward function for this problem proved to be quite arduous due to the novel traffic environment. As also stated earlier, to the best of our knowledge, there is only limited amount of lane-free research using DRL, and related work in the literature is typically based on the existence of driving lanes (see Section 2.4), which constitutes a different problem altogether. For this reason, a reward function was constructed specifically for lane-free environments.

Several components of reward functions were investigated in order to explore their mechanisms of influence, and to find the most effective form that combines them. Before presenting the various components, we first determine the agent's objectives within the lane-free traffic environment.

The designed reward function should combine the two objectives of the problem at hand, that is, (a) maintaining the desired speed v_d and (b) avoiding collisions with other vehicles. All of the presented reward components attempt to tackle these two objectives. Some are more targeted only toward the end goal and do not provide the agent with information for intermediate states, (i.e., provide delayed rewards); others are more elaborate and informative, and consequently tend to better guide the agent towards the aforementioned goals. Naturally, the more informative rewards aid in the learning process for the baseline algorithms examined. We also observed a strong influence on the results (see Section 4.3). Of course, such informative rewards arguably add bias to the optimization procedure and provide more specific solutions compared to delayed rewards. This bias can potentially limit the agent's capability to explore the whole solution space and obtain a (near-) optimal policy for the problem at hand. Still, whether only delayed rewards are adequate depends on the algorithm of choice and its ability to harness the whole solution space for a specific problem.

3.4.1. Longitudinal Target

Regarding the desired speed objective, we utilize a linear function that focuses on maintaining the desired speed. In detail, the function is linear with respect to the current longitudinal speed v_x and calculates a reward based on the deviation from the desired speed v_d of the agent at that specific time step. To achieve this, the following mathematical formula is used:

$$c_x = \frac{|v_x - v_d|}{v_d} \quad (6)$$

It is evident that this function tends to be minimized at 0 whenever we approach the respective goal. As such, the form of the total reward r_x is a reciprocal function that contains a weighted form of c_x in the denominator. Our choice for a reciprocal form is consistent across the subsequent reward components and their combinations, as was selected after preliminary empirical investigation among different reward function forms. We found that a reciprocal form yields superior results, especially when multiple components are combined, compared to, e.g., a linear combination of reward.

$$r_x = \frac{\epsilon_r}{\epsilon_r + w_x \cdot c_x} \quad (7)$$

where r_x is the total "longitudinal" (i.e., x -movement-axis-specific) reward at any time-step t , and ϵ_r is a parameter that allows the reward to be maximized at 1 whenever c_x tends to 0. Moreover, w_x is a weighting coefficient that quantifies the normalized cost c_x , and its use serves to combine c_x with the other components that we subsequently introduce, i.e., balancing multiple objectives appropriately and combining them in a unified reward

function (as we showcase later in Section 3.4.6). We chose a small value for ϵ_r , specifically, $\epsilon_r = 0.1$, so as to make the minimum reward close to 0 when c_x is maximized.

3.4.2. Overtake Motivation Term

In our preliminary experiments, we determined that the agent tends to get stuck behind slower vehicles, as that is deemed a “safer” action. However, this behavior is not ideal, as it usually leads to a greater deviation from the desired speed. To address this particular problem, we include a function that motivates the agent to overtake its surrounding vehicles.

In detail, a positive (and constant) reward term r_o is attributed whenever the agent overtakes one of its neighboring vehicles. However, this reward is received only in cases where there are no collisions.

$$r_{x,o} = \begin{cases} r_x + r_o, r_o > 0 & \text{if agent does not collide \& overtakes a vehicle;} \\ r_x & \text{otherwise} \end{cases} \quad (8)$$

where r_x denotes the reward function regarding the desired speed objective, as described in Equation (7).

3.4.3. Collision Avoidance Term

Concerning the collision elimination objective, our first step was to incorporate collisions into the reward function. In this light, we examined numerous components, the first of which was a “simpler” reward, by incorporating the training objective directly into the reward, aiming to “punish” the agent whenever a collision occurs.

This is exclusively based on the collisions between our agent and its surrounding vehicles. Specifically, a negative constant reward value r_c is received whenever a collision occurs. Essentially, provided the reward r_x according to the longitudinal target, the reward $r_{x,c}$ that the agent receives is calculated as:

$$r_{x,c} = \begin{cases} r_x + r_c, r_c < 0 & \text{if agent is involved in a collision;} \\ r_x & \text{otherwise} \end{cases} \quad (9)$$

However, this imposes the issue of delayed (negative) rewards, which can inhibit learning. In our domain of interest especially, the agent can be in many situations where a collision is inevitable, even many time-steps before the collision actually occurs. This depends on the speed of our agent, along with the speed deviation and distance from the colliding vehicle.

3.4.4. Potential Fields

To tackle the problem of delayed rewards, we also employ an additional, more informative reward component, one that “quantifies” the danger of a collision between two vehicles. The use of ellipsoid fields has been already utilized for lane-free autonomous driving as a measurement of the probability of a collision with another vehicle [8,9]. Provided a pair of vehicles, the form of the ellipsoid functions evaluates the danger of collision, taking into account the longitudinal and lateral distances, along with the respective longitudinal and lateral speeds of the vehicles and their deviations.

Given our agent and a neighboring vehicle j , with longitudinal and lateral distances dx_j, dy_j , and longitudinal and lateral speed deviations $dv_{x,j}, dv_{y,j}$, the form of the ellipsoid functions is as follows:

$$c_{f,j} = E_c(dx_j, dy_j) + E_b(dx_j, dy_j, dv_{x,j}, dv_{y,j}). \quad (10)$$

where both $E_c(dx_j, dy_j)$ and $E_b(dx_j, dy_j, dv_{x,j}, dv_{y,j})$ have an ellipsoid function and capture a critical and broad region, respectively, and $c_{f,j}$ is a cost that quantifies the danger of collision between our agent and neighbor j .

The particular ellipsoid form was influenced by [31] and is the following:

$$E(d_x, d_y) = \frac{m_f}{\left(\left(\frac{|d_x|}{a} \right)^{p_x} + \left(\frac{|d_y|}{b} \right)^{p_y} + 1 \right)^{p_t}} \quad (11)$$

where d_x and d_y are longitudinal and lateral distances; and a and b are regulation parameters for the range of the field for the two dimensions, x and y , respectively. The exponents marked as p_x , p_y , and p_t shape the ellipse; and lastly, the parameter m_f defines the magnitude when the distances are close to 0.

Essentially, the critical region is based only on the distance between the two vehicles (agent and neighbor j), and the broad region stretches appropriately according to the speed deviations, so as to properly inform the system on the danger of a collision from a greater distance, and consequently, the agent has more time to respond appropriately. The interested reader may refer to [8] for more information on these functions.

Moreover, we also need to accumulate the corresponding values for all neighboring agents, i.e., $c_{f,all} = \sum_j c_{f,j}$ for each neighboring vehicle j within the state observation at a given time-step t . Finally, we want the associated cost to be upper-bounded, so we have:

$$c_f = \min\{c_{f,all}, 1\} \quad (12)$$

We know that each ellipsoid function is bounded within $[0, m_f]$, where m_f is a tuning parameter. As such, each cost $c_{f,j}$ is bounded within $[0, 2m_f]$. Therefore, m is set accordingly ($m_f = 0.5$), so as to normalize all $c_{f,j}$ values to $[0, 1]$. Thus, considering the form of the reward r_x corresponding to the longitudinal target of desired speed (Equation (7)), the total reward $r_{x,f}$ that incorporates the use of (potential) fields follows the reciprocal form as well, which contains the related cost terms:

$$r_{x,f} = \frac{\epsilon_r}{\epsilon_r + w_x \cdot c_x + w_f \cdot c_f} \quad (13)$$

Notice that $r_{x,f} = r_x$ whenever there is no captured danger with neighboring vehicles; i.e., the ellipsoid function for each neighbor j returns $c_{f,j} = 0$.

3.4.5. Incorporating a Lateral Movement Target with Construction of Overtaking Zones

During the experimental evaluation of the aforementioned methods, we noticed that even though a significant number of collisions were avoided, there were still some occurrences where the agent could not learn to react properly, specifically in highly populated environments. Therefore, in order to further reduce the number of collisions, we constructed a method that translates the notion of the longitudinal target (see Section 3.4.1) to the lateral movement. In particular, we examined a reciprocal form for the lateral component, by calculating the associated cost as the normalized deviation of the agent's lateral position y from its desired lateral position y_d :

$$c_y = \frac{|y - y_d|}{w_r} \quad (14)$$

where y is our agent's lateral position at the time, y_d describes the agent's desired lateral position, and w_r is the width of the road, so that the cost c_y is also bounded within $[0, 1]$. Considering that the vehicle always lies within the road boundaries, the deviation from the desired (lateral) position cannot exceed the width of the road.

As we discuss later in this Section, to acquire an appropriate desired lateral position y_d , we devised a method that identifies available lateral zones to move towards, evaluates them, and returns an appropriate lateral point for the vehicle.

Similarly to Equation (7), this cost function tends to be minimized at 0 whenever we approach the respective goal. As such, the form of the total reward $r_{x,y}$ has again a

reciprocal function that contains a weighted sum of c_y and c_x in the denominator to balance the longitudinal target as well.

$$r_{x,y} = \frac{\epsilon_r}{\epsilon_r + w_x \cdot c_x + w_y \cdot c_y} \tag{15}$$

where ϵ_r is a parameter that allows the reward to be maximized at 1 whenever the weighted sum of costs tends to 0. We chose a small value for ϵ_r , specifically, $\epsilon_r = 0.1$, so as to make the minimum reward be close to 0 when c_x, c_y are maximized.

As mentioned earlier, we constructed an algorithm that estimates an appropriate desired lateral position y_d for the agent to occupy. This serves to provide information for the agent so as to avoid collisions with vehicles downstream, especially if an overtaking maneuver is taking place (when vehicle(s) downstream drive slower). For this computation, the space downstream of the vehicle is partitioned (with respect to the y axis) into zones, as illustrated in Figure 3. These zones reflect potential regions that the vehicle may choose to drive towards. Naturally, the lateral space occupied by vehicles in front is discarded, as illustrated in Figure 3. For the remaining zones, we also dismiss ones that our vehicle does not actually fit, e.g., the zone in red in the figure. When more than one zone is available, we simply select the one closer to our agent, since it will result in a more gradual maneuver. The desired lateral position y_d will be the center point of the chosen zone (the potential choices for y_d are evident in the figure with dashed lines).

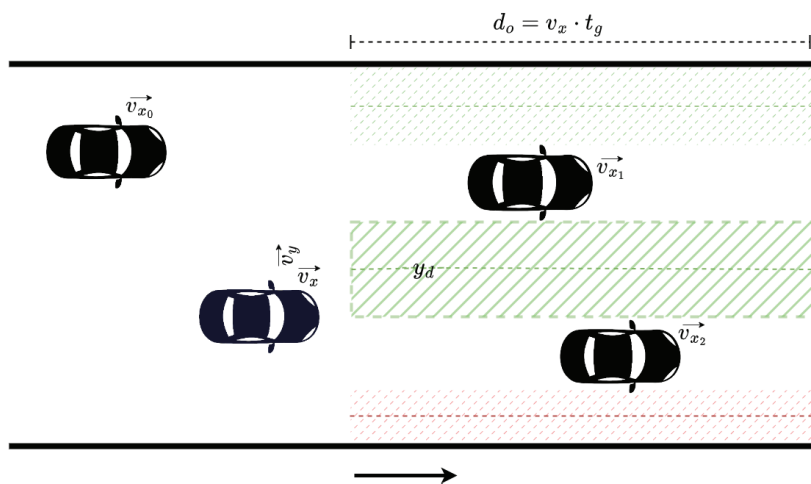


Figure 3. Zone-selection process.

The observation range is dynamic for this process, adapting to the longitudinal speed of our agent, and the range is selected through a timegap value t_g . This dictates a longitudinal distance d_o downstream of our agent through its longitudinal speed v_x —i.e., we scan until reaching the longitudinal distance d_o for vehicles, in order to determine y_d . This distance is calculated as: $d_o = v_x \cdot t_g$. If vehicles are observed within the specified distance d_o , the entire road is considered as a single zone; therefore, y_d is the center of the entire road’s width.

Note that we may also be unable to determine an available zone if vehicles downstream completely block any overtaking maneuver. In that case, it is evident that the methodology outlined above is not appropriate. When a zone cannot be determined due to heavy traffic, then the agent is in any case not able to overtake and is forced to remain behind the vehicles in front. Therefore, the desired speed v_d of the agent is adjusted according to the slowest vehicle blocking its way, and the desired lateral position y_d is set behind the fastest blocking vehicle in front. Consequently, the agent will be able to overtake sooner, and with the adjustment in the desired speed, it will not have any motivation to overtake unless it is actually feasible, i.e., without causing a collision.

3.4.6. Combining Components into a Single Reward Function

All of the aforementioned reward components can be combined into a single reward function of the form:

$$r_{x,y,f,c,\rho} = \begin{cases} \frac{\epsilon_r}{\epsilon_r + w_x \cdot c_x + w_y \cdot c_y + w_f \cdot c_f} + r_c & \text{if agent is involved in a collision} \\ \frac{\epsilon_r}{\epsilon_r + w_x \cdot c_x + w_y \cdot c_y + w_f \cdot c_f} + r_o & \text{if agent does not collide \& overtakes a vehicle;} \\ \frac{\epsilon_r}{\epsilon_r + w_x \cdot c_x + w_y \cdot c_y + w_f \cdot c_f} & \text{otherwise} \end{cases} \quad (16)$$

where the coefficients w_x, w_y, w_f and reward terms r_c, r_o are appropriately tuned in order to balance all the different components or choose to completely neglect certain components by simply setting the associated coefficient to 0. Thereby, we can directly examine all different combinations of components with this reward form by setting the coefficients accordingly.

As is evident in the following results, each reward component within the final reward function results in an additional improvement to the policy of the agent, without causing deterioration of the overall efficiency. The use of potential fields and then the lateral zones' reward components provided the most significant efficiency improvements, and the resulting policies containing these managed to tackle both objectives and with consistent performance, even in environments with varying levels of difficulty, as discussed in Section 4.3.

4. Experimental Evaluation

In this section, we present: our experimental results through a comparative study of the different reward functions that we propose; various parameter settings that aim to showcase trade-offs between the two objectives; and a comparison between the examined DRL algorithms, namely, DDPG and DQN.

4.1. RL Algorithms' Setup

First, we specify some technical aspects of the proposed implementation. Regarding the DQN algorithm and its extensions, we employed the Adam [32] optimization method to update the weight coefficients of the network at each learning step. The ϵ -greedy policy [18] was employed for action selection, in order to balance exploration and exploitation during training, with ϵ decreasing linearly from 1 (100% exploration) to 0.1 (10% exploration) over the first 200 episodes, and fixed to 0.1 thereafter. We utilized DQN and its extension with a deep neural network of 128 neurons in the first hidden layer and 64 in the second hidden layer using a rectified linear unit (ReLU) activation function. The network outputs 9 elements that correspond to the estimated Q-value of each available action.

In the setup for the DDPG algorithm, we also used Adam to train both the actor and the critic. Furthermore, we chose to use the Ornstein–Uhlenbeck process to add noise (as an exploration term) to the action output, as employed in the original paper [24]. The actor network employed in the DDPG implementation contained 256 neurons in the first hidden layer, 128 in the second hidden layer, and 2 in the output layer. Again, ReLU activation function was used for all hidden layers, and the output used the hyperbolic tangent (tanH) activation function, so as to provide a vector of continuous values within the range $[-1, 1]$. Similarly, the critic network contained 256 neurons in the first hidden layer, 128 in the second hidden layer, and 1 neuron in the output layer, including a ReLU activation function for all hidden layers and a linear activation unit in the output layer.

Training scenarios included a total of 625 episodes for all experiments. We have empirically examined different parameter tunings concerning the learning rate, discount factor, and the number of training episodes for both DQN and DDPG, with the purpose of finding the configurations that optimize the agent's behavior. We provide the values of the various parameters in Table 1. Finally, the obtained results were acquired using a system running Ubuntu 20.04 LTS with an AMD Ryzen 7 2700X CPU, an NVIDIA GeForce

RTX 2080 SUPER GPU, and 16 GB RAM. Each episode simulated 200 s on a ring-road with many vehicles. With the system configuration above, each episode required on average 30 s approximately. This execution time included the computational time for training the neural networks at every time-step, meaning that even real-time training would be feasible for a DRL lane-free agent.

Table 1. Hyper-parameters for RL algorithms.

Parameter	Value	Parameter	Value
Learning rate used by Adam α	0.001	Mini-Batch size	64
Discount factor (DQN)	0.98	Discount factor (DDPG)	0.98
Replay Memory size (DQN)	50,000	Replay Memory size (DDPG)	100,000
Soft update parameter (DDPG)	0.001	Number of episodes for training N	625

4.2. Simulation Setup

The proposed implementation heavily relies on neural network architectures, since all DRL methods incorporate them for function approximation. As such, in the context of this work, we utilized:

- TensorFlow [33]: An open-source platform used for numerical computation, machine learning, and artificial intelligence applications, which was developed by Google. It supports commonly used languages, such as Python and R, and also makes developing neural networks faster and easier.
- Keras [34]: A high level, open-source software library for implementing neural networks. In addition, Keras facilitates multiple backend neural network computations, while providing a Python frontend and being complementary to the TensorFlow library.

We trained and evaluated all methods on a lane-free extension of the Flow [35] simulation tool, as described in [8]. Moreover, to facilitate the experiments, we utilized the Keras-RL library [36]. The Keras-RL library implements some of the most widely used deep reinforcement learning algorithms in Python and seamlessly integrates with Tensorflow and Keras. However, technical adjustments and modifications were necessary to make this library compatible with our problem and environment, as it did not conform to a standard Gym environment setup.

The lane-free driving agent was examined in a highway environment with the specified parameter choices of Table 2, whereas in Table 3, we provide the parameter settings related to the MDP formulation and specifically the reward components. Regarding the lane-free environment, we examined a ring-road with a width of 10.2 m, which is equivalent to a conventional 3-lane highway. The road's length and vehicles' dimensions were selected in order to allow a more straightforward assessment of our methods. The choices for the weighting coefficients and related reward terms were selected after a meticulous experimental investigation.

Table 2. Simulation parameters.

Parameter	Value
Highway length	500 m
Highway width	10.2 m
Vehicles' length	3.5 m
Vehicles' width	1.8 m
Types of vehicles	2
Num. of vehicles	35
Agent's length	3.2 m
Agent's width	1.6 m
Episode Time Length	200 s
Time-Interval	0.25 s

Table 3. Parameter choices related to the MDP formulation.

Parameter	Value
timegap t_g	0.7
agent's desired speed v_d	20 m/s
other vehicles' desired speed	18 m/s to 22 m/s
num. of vehicles in state n	5
longitudinal observation distance d	80 m
w_y	0.65
w_f	1
w_x (for RFs with lateral target)	0.35
w_x (for RFs without lateral target)	0.65
k_r	0.5
r_c	-2.5
r_o	2
e_v	0.4

4.3. Results and Analysis

The effectiveness of all reward functions was evaluated based on three metrics. These were: the average reward value, the speed deviation from the desired speed (for each step, we measured the deviation of the current longitudinal speed from the desired one ($v_x - v_d$), in m/s), and of course, the average number of collisions. All results were averaged from 10 different runs.

We typically demonstrate in all figures the designed agent's average reward and speed deviation, and the average number of collisions for each episode. In the examined reward functions, the longitudinal target reward (Section 3.4.1) was always employed, whereas other components associated with the collision-avoidance objective were evaluated for many different combinations, in order to provide an ablation study, i.e., show how each component affects the agent. To be exact, for each of the tested reward functions, we employed Equation (16) while assigning the values of Table 3 to the corresponding weights when the equated components were used. Otherwise, we set them to 0. We refer the reader to Table 4 for a complete list of all different reward functions examined, including the associated equations stemming from the general reward function (Equation (16)) and the subsections relevant to their descriptions. The constant reward terms r_c, r_o were of course not always added, but were according to Equation (16).

In Section 4.3.1, we first demonstrate the performance of our reward functions that do not involve the lateral target component, namely, the "Fields RF", the "Collision Avoidance RF", the "Overtake and Avoid Collision RF", the "Fields and Avoid Collision RF", and the "Fields, Overtake and Avoid Collision RF" functions. Next, in Section 4.3.2, we introduce the concept of the zones component (with the use of lateral targets) to our experimental procedure, by comparing the "Fields, Overtake and Avoid Collision RF" to the "Fields, Zones, Overtake and Avoid Collision RF" and the "Zones, Overtake and Avoid Collision RF". In addition, to collate our two most efficient reward functions, we examine, in

Section 4.3.3, their behavior in more complex and demanding lane-free environments with higher traffic densities.

The evaluation described above was conducted using the DDPG algorithm. This was done since extensive empirical testing, along with the results of the comparative evaluation of DRL algorithms presented in Section 4.3.4, suggest that DDPG is a suitable DRL algorithm for this complex continuous domain (and indeed, exhibits the best overall performance when compared to DQN and its extensions).

Table 4. All reward functions.

Function Label	Equation	Component(s) Description
Collision Avoidance RF	$\frac{\epsilon_r}{\epsilon_r + w_x \cdot c_x} + r_c$	Section 3.4.3
Overtake and Avoid Collision RF	$\frac{\epsilon_r}{\epsilon_r + w_x \cdot c_x} + r_c + r_o$	Sections 3.4.2 and 3.4.3
Fields RF	$\frac{\epsilon_r}{\epsilon_r + w_x \cdot c_x + w_f \cdot c_f}$	Section 3.4.4
Fields and Avoid Collision RF	$\frac{\epsilon_r}{\epsilon_r + w_x \cdot c_x + w_f \cdot c_f} + r_c$	Sections 3.4.3 and 3.4.4
Fields, Overtake and Avoid Collision RF	$\frac{\epsilon_r}{\epsilon_r + w_x \cdot c_x + w_f \cdot c_f} + r_c + r_o$	Sections 3.4.2–3.4.4
Zones, Overtake and Avoid Collision RF	$\frac{\epsilon_r}{\epsilon_r + w_x \cdot c_x + w_y \cdot c_y} + r_c + r_o$	Sections 3.4.2, 3.4.3 and 3.4.5
Fields, Zones, Overtake and Avoid Collision RF	$\frac{\epsilon_r}{\epsilon_r + w_x \cdot c_x + w_y \cdot c_y + w_f \cdot c_f} + r_c + r_o$	Sections 3.4.2–3.4.5

4.3.1. Evaluation of the Reward Function Components

We refer to the reward associated with the collision avoidance term (Equation (9)) as “Collision Avoidance RF”, and the addition of the overtaking motivation (Equation (8)) as “Overtake and Avoid Collision RF”. Furthermore, the use of the fields (Equation (13)) for that objective are labeled as “Fields RF”, and “Fields and Avoid Collision RF” when combined with the collision avoidance term. Finally, the assembly of the collision avoidance term, the overtaking motivation, and the potential fields components in a single reward function is referred to as “Fields, Overtake and Avoid Collision RF”, whereas in our previous work [13] it was presented as the “All-Components RF”. All of the aforementioned functions demonstrate how the agent’s policy has improved over time.

As is evident in Figures 4–6, the “Collision Avoidance RF” managed to maintain a longitudinal speed close to the desired one. Still, it did not manage to decrease the number of collisions sufficiently. Moreover, we see that the addition of the overtaking component in “Overtake and Avoid Collision RF” achieved a longitudinal speed slightly closer to the desired one, though the collision number was still relatively high. On the contrary, according to the same figures, the “Fields RF” exhibited similar behavior to the previously mentioned reward functions, but with slight improvement in collision occurrences. Finally, both the “Fields and Avoid Collisions RF” and the “Fields, Overtake and Avoid Collision RF” performed slightly worse in terms of speed deviations. However, they obtained significantly better results in terms of collision avoidance (“Collision Avoidance RF”, “Fields RF” and “Overtake and Avoid Collision RF” performed 3.5-, 2.7- and 3.5-times worse with respect to collision avoidance when compared to “Fields, Overtake and Avoid Collision RF”), thereby balancing the two objectives much better. On closer inspection though, the “Fields, Overtake and Avoid Collision RF” managed to maintain a smaller speed deviation and fewer collisions, thus making it the reward function of choice for a more effective policy overall.

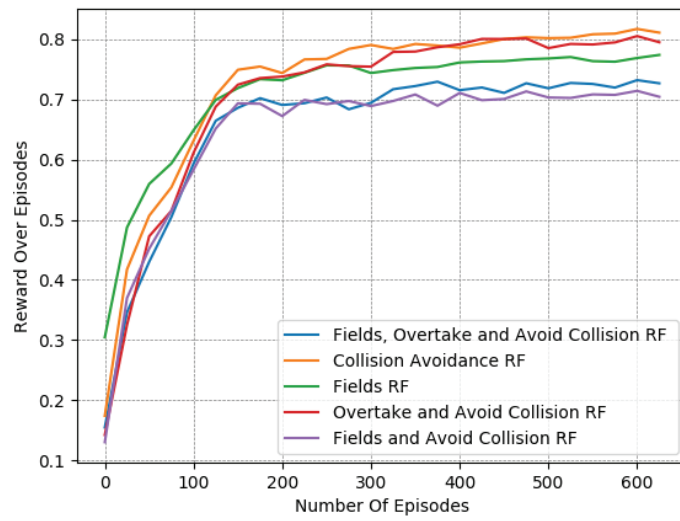


Figure 4. Reward over time for different reward functions.

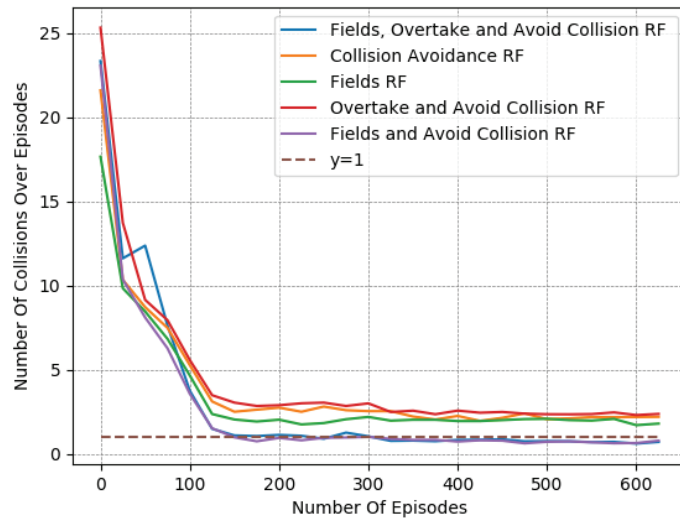


Figure 5. Collisions over time for different reward functions.

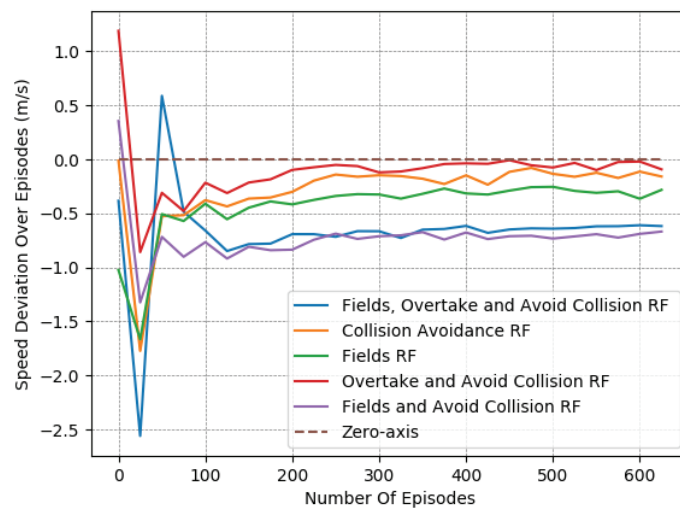


Figure 6. Speed deviation over time for different reward functions.

To further demonstrate this point, we present in Table 5 a detailed comparison between these five reward functions. The reported results were averaged from the last 50 episodes of each variant. The learned policy had converged in all cases, as shown in Figures 4–6.

Table 5. Comparing the different reward functions.

Reward Function	Collision Avoidance RF	Fields RF	Fields & Avoid Collision RF	Overtake & Avoid Collision RF	Fields, Overtake & Avoid Collision RF
Collisions	2.256	1.761	0.726	2.293	0.649
Speed Dev. (m/s)	−0.135	−0.323	−0.698	−0.058	−0.613

Evidently, higher rewards do not coincide with fewer collisions, meaning that the reward metric should not be taken at face value when we compare different reward functions. This is particularly noticeable in the case of the “Fields, Overtake and Avoid Collision RF” and the “Fields and Avoid Collisions RF”, where there is a reduced reward over episodes, but when observing each objective, they clearly exhibit the best performances. This was expected, since the examined reward functions have different forms. In Table 5, we can also observe the effect of the “Overtake” component. Its influence on the final policy is apparent only when combined with “Fields and Avoid Collisions RF”, i.e., forming the “Fields, Overtake and Avoid Collision RF”.

Policies resulting from different parameter tunings that give more priority to terms related to collision avoidance (r_c, w_f) do in fact further decrease collision occurrences, but we always observed a very simplistic behavior where the learned agent just followed the speed of a slower moving vehicle in front; i.e., it was too defensive and never attempted overtake. Such policies did not exhibit intelligent lateral movement, and therefore were of no particular interest given that we were training an agent to operate in lane-free environments. Therefore, these types of parameter tunings that mainly prioritized collision avoidance were neglected.

For the subsequent experiments, we mostly refrain from commenting on the average reward gained and mainly focus on the results regarding the two objectives of interest—namely, collision avoidance and maintaining a desired speed. Nevertheless, we still demonstrate them, so as to also present the general learning improvement over episodes across all experiments.

As discussed in the related conference paper [13], the most promising reward function form was at this point the “Fields, Overtake and Avoid Collision RF”. Here, we further investigate the influence of the additional component presented, namely, the lateral target component that makes use of the zone selection technique, as presented in Section 3.4.5. As we discuss below, the inclusion of the “Zones” component in the reward provided us with marginal improvement with respect to the collision-avoidance objective, at the expense of the desired speed task. However, its contribution regarding collision avoidance was much more evident when investigating intensified traffic conditions with more surrounding lane-free vehicles (see Section 4.3.3).

4.3.2. Evaluation of the Zone Selection Reward Component

In particular, we present in Figures 7–9 a detailed comparison between: the “Fields, Overtake and Avoid Collision RF” and the “Fields, Zones, Overtake and Avoid Collision RF” in Equation (16). Additionally, we further highlight the impacts of potential fields on the designed reward functions by including one more variant of the reward function for comparison. Accordingly, we present results for another variation titled “Zones, Overtake and Avoid Collision RF” that lacks the field’s related reward component.

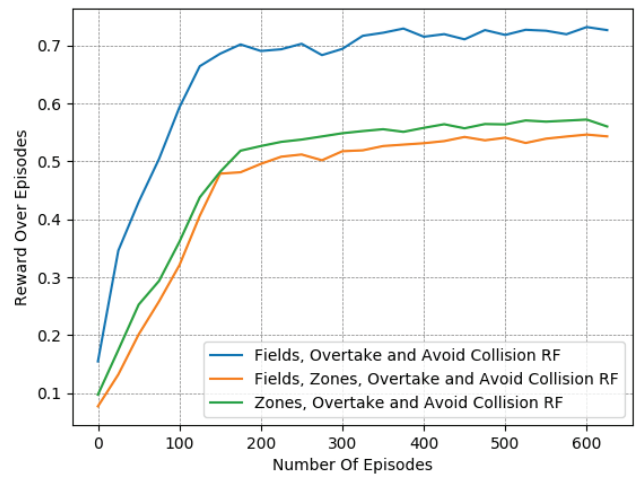


Figure 7. Reward over time for different reward functions: evaluation of the zone selection reward component.

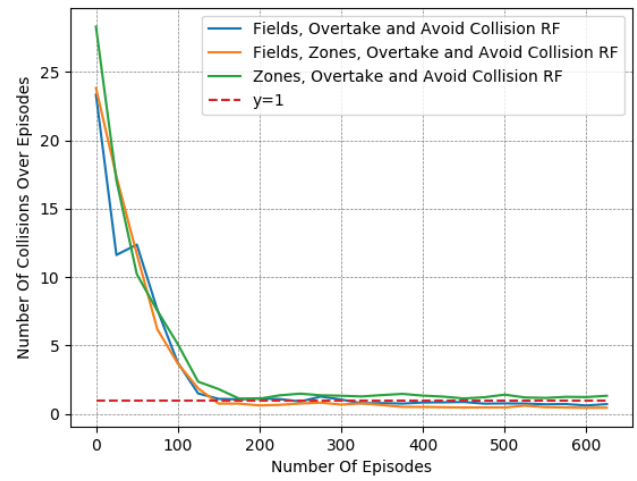


Figure 8. Collisions over time for different reward functions: evaluation of the zone selection reward component.

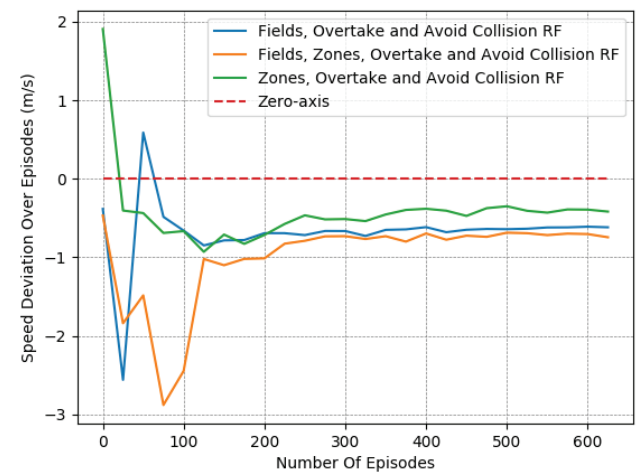


Figure 9. Speed Deviation over time for different reward functions: evaluation of the zone selection reward component.

In Table 6, we provide a closer look at the comparison between these 3 reward functions. The reported results were averaged from the last 50 episodes of each variant. The learned policy converged in all cases. Here, we observe that just the addition of the lateral target component improved the performance notably, as it managed to moderately mitigate collision occurrences, at a marginal expense to the desired speed objective. However, this deviation from the desired speed in the experiments is to be expected. Maintaining the desired speed throughout an episode is not realistic, since slower downstream traffic will, at least partially, slow down an agent. Still, the use of lateral zones is beneficial only when combined with the fields component; otherwise, we can see that the agent performs worse with respect to the collision-avoidance task, while obtaining quite similar speed deviations.

In general, the use of lateral zones provides important information to the agent that is combined with the overtaking task but can undermine safety. In preliminary work with different parameter tunings, we observed that the bias of this information caused notable performance regression regarding collisions. This occurred when the zones-related component was given more priority, especially in environments with higher vehicle densities. In practice, the selected parameter tuning should not allow for domination of the fields reward by the lateral zones' reward component.

Table 6. Comparing the different reward functions: the effect of the “Zones” reward component.

Reward Function	Fields, Zones, Overtake & Avoid Collision RF	Fields, Overtake & Avoid Collision RF	Zones, Overtake & Avoid Collision RF
Collisions	0.449	0.649	1.282
Speed Dev. (m/s)	−0.723	−0.613	−0.403

Throughout our experiments, it was obvious that the two objectives were countering each other. A vehicle operating at a slower speed is more conservative, whereas a vehicle wishing to maintain a higher speed than its neighbors needs to overtake in a safe manner, and consequently has to learn a more complex policy that performs such elaborate maneuvering. Specifically, we do note that the experiments with the smallest speed deviations were those with the highest numbers of collisions, and on the other hand, those that showcased small numbers of collisions deviated the most from the desired speed.

In addition, according to the results presented in Table 6, it is evident that “Fields, Overtake and Avoid Collision RF” and the “Fields, Zones, Overtake and Avoid Collision RF” result in quite similar policies in the training environment, despite the fact that the second one is much more informative.

4.3.3. Evaluation for Different Traffic Densities

Thus, to perform a more comprehensive and thorough comparison, we decided to test the 2 most promising reward functions in more complex and demanding lane-free environments. We chose to run both “Fields, Overtake and Avoid Collision RF” and the “Fields, Zones, Overtake and Avoid Collision RF”, using a set of different traffic densities. Specifically, in Figures 10–12 we illustrate the results of running the reciprocal RF using densities equal to 70, 90, and even 120 veh/km (vehicles per kilometer). Meanwhile, in Figures 13–15, we demonstrate the corresponding outcomes when running the “Fields, Overtake and Avoid Collision RF” for the same set of traffic densities.

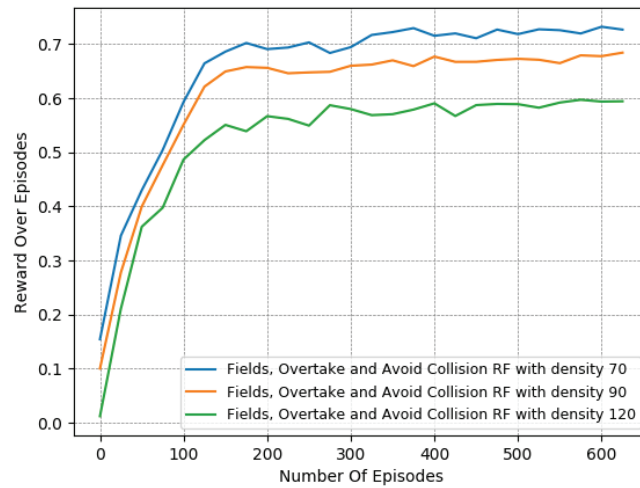


Figure 10. Reward over time for the Fields, Overtake and Avoid Collision RF for different traffic densities.

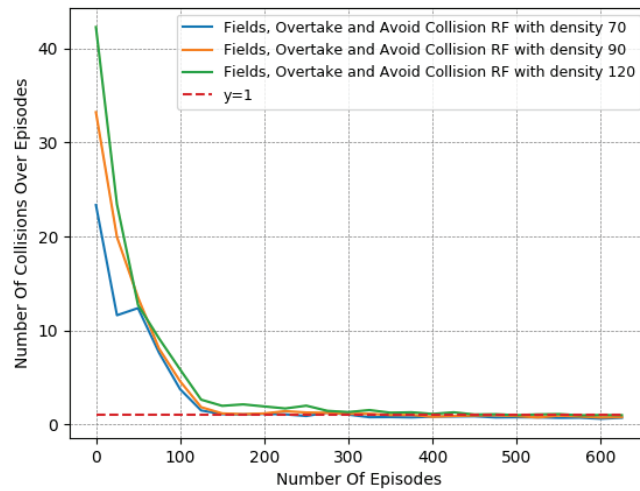


Figure 11. Collisions over time for the Fields, Overtake and Avoid Collision RF for different traffic densities.

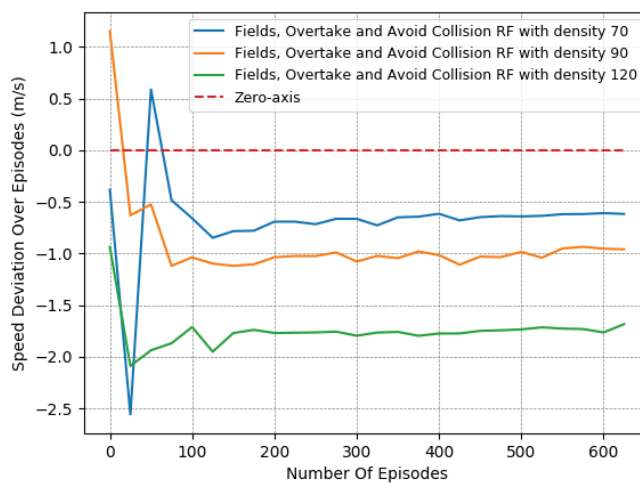


Figure 12. Speed Deviation over time for the Fields, Overtake and Avoid Collision RF for different traffic densities.

We observe that when using the “Fields, Overtake and Avoid Collision RF”, the agent tends to handle the surrounding traffic quite well. In detail, it is noticed that the number of collisions decreased dramatically with the passage of the episodes, and in all cases, at the end of the training, it approached or fell below one on average. At the same time, the collisions and the deviation in the agent’s speed from the desired one scaled according to the density of the surrounding vehicles. However, this behavior is to be expected, since in denser traffic environments, vehicles tend to operate at lower speeds and overtake less frequently, as the danger of collision is more present.

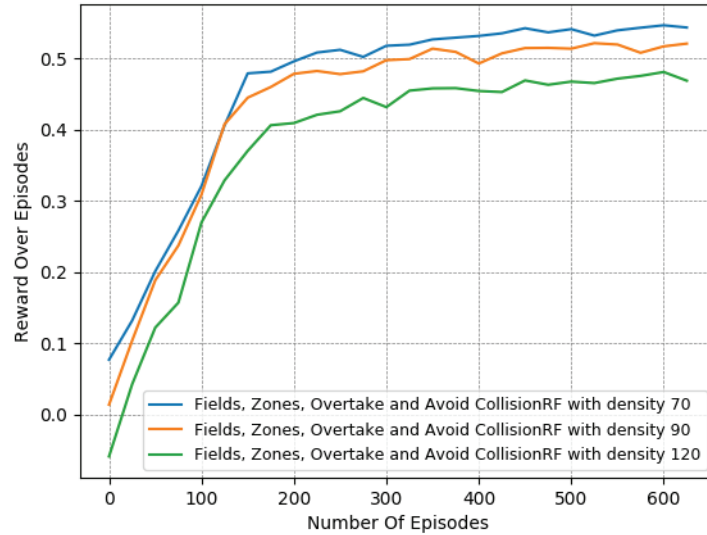


Figure 13. Reward over time for the Fields, Zones, Overtake and Avoid Collision RF for different densities.

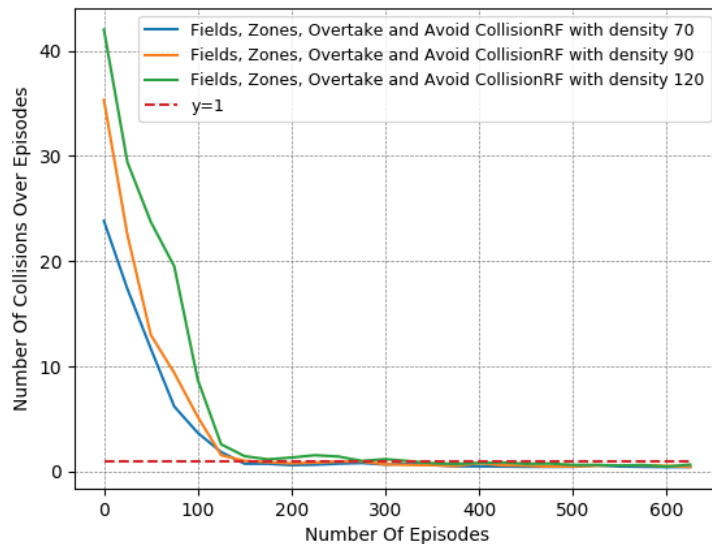


Figure 14. Collisions over time for the Fields, Zones, Overtake and Avoid Collision RF for different densities.

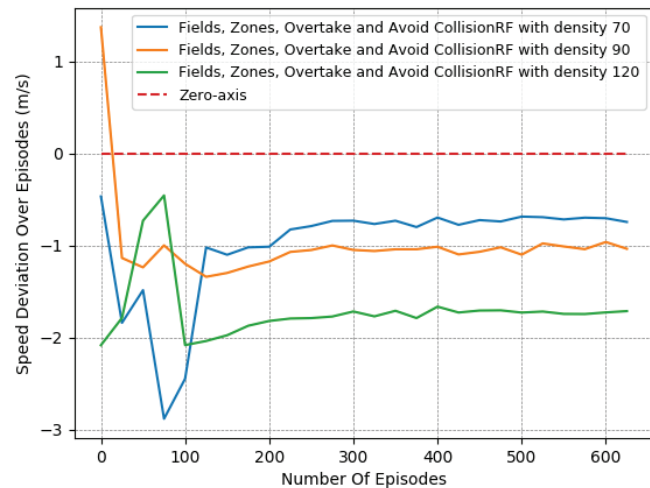


Figure 15. Speed deviation over time for the Fields, Zones, Overtake and Avoid Collision RF for different densities.

Similarly, according to the results presented in Figures 13–15, we note the impact of the “Zones” reward component Section 3.4.5 in our problem, as it managed to boost the agent’s performance, especially when compared, in denser traffic, to a reward function that incorporated the same other components, namely, the “Fields, Overtake, and Avoid Collision RF”. In particular, while the speed objective did not showcase any significant deviation between the two variants, the difference was quite noticeable in collision avoidance, where the increase was substantially mitigated, resulting in more robust agent policies with respect to the traffic densities. Again, we emphasize that this benefit of the lateral zones is evident only when combined with the other components, and especially with the fields-related reward. Without the use of fields, the other reward components cannot adequately tackle the collision-avoidance task, especially in demanding environments with heavy traffic.

In addition, a more direct comparison of the behavior of the two reward functions is found in Table 7. The numerical results presented confirm that both of the compared reward functions achieve consistent performance, regardless of the difficulty of the environment. Nevertheless, they also confirm the superiority of the “Fields, Zones, Overtake and Avoid Collision RF”, since even in environments with higher densities, the agent mitigated both of the training objectives simultaneously, and by the end, the number of collisions was much lower and close to 0.5.

Table 7. Evaluating the efficiency of the “Zones” reward component under different traffic densities.

Reward Function Density	Fields, Zones, Overtake & Avoid Collision RF			Fields, Overtake & Avoid Collision RF		
	70	90	120	70	90	120
Collisions	0.449	0.508	0.591	0.649	0.771	1.066
Speed Dev. (m/s)	−0.723	−0.998	−1.718	−0.613	−0.956	−1.715

4.3.4. Comparison of Different DRL Algorithms

Finally, we provide a set of experiments that compared different DRL algorithms in Figures 16–18. We employed the “Fields, Zones, Overtake and Avoid Collision RF”, using DQN, double DQN (DDQN), DDQN with dueling architectures (DNA), and DDQN with prioritized experience replay (PER), and compare their performances to that of DDPG.

It is apparent for all five methods that the learning process attempts to guide the agent to the expected behavior. However, DDPG clearly exhibited the best performance, as it is the only method that resulted in a number of collisions under 0.5 on average while managing to preserve a speed that was close to the desired one. Upon closer examination, by observing the averaged results extracted from the last episodes of each variant, as

presented in Table 8, DQN, DDQN, DNA, and PER resulted in smaller speed deviations, yet they caused significantly higher numbers of collisions. All DQN-related methods exhibited quite similar behavior in our lane-free environment, as visible in the related figures. Only the PER variant exhibited a notable deviation within the learning curves. Evidently, when utilizing PER memory, i.e., using the TD error to influence the probability of sampling, the agent results temporarily in a worse policy around training episodes $\approx [50-150]$. This was apparent for multiple random seeds. Still, the collision avoidance metric under PER is still 76% worse compared to the DDPG at the end of training.

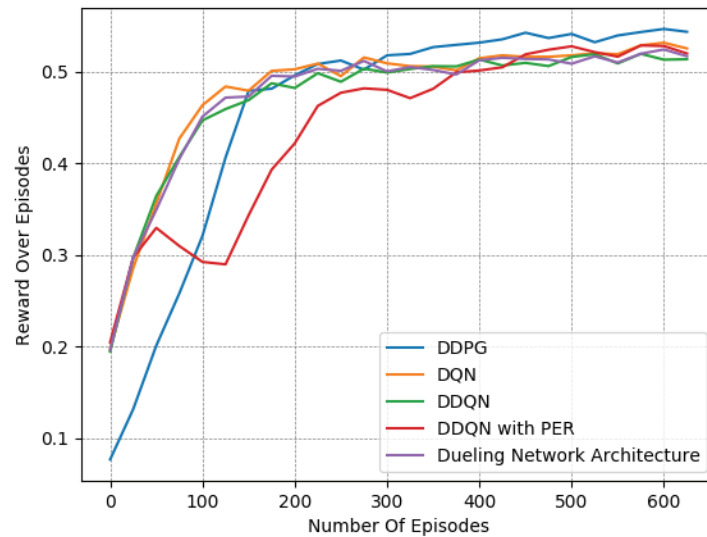


Figure 16. Reward over time for the Fields, Zones, Overtake and Avoid Collision RF with different DRL algorithms.

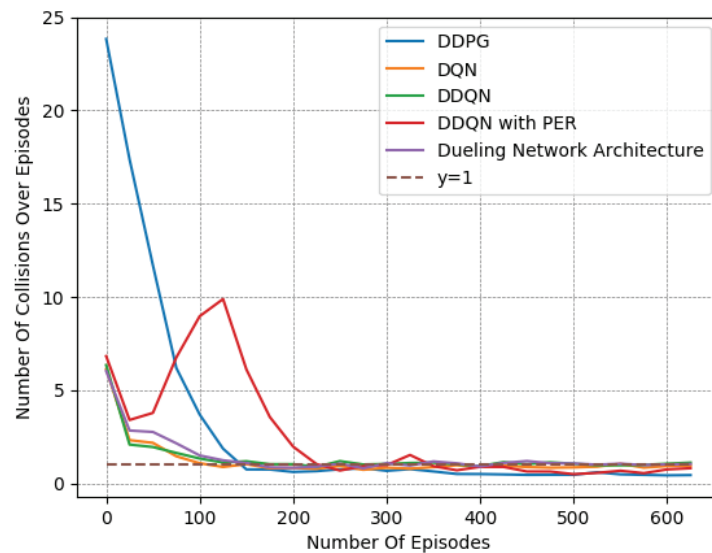


Figure 17. Collisions over time for the Fields, Zones, Overtake and Avoid Collision RF with different DRL algorithms.

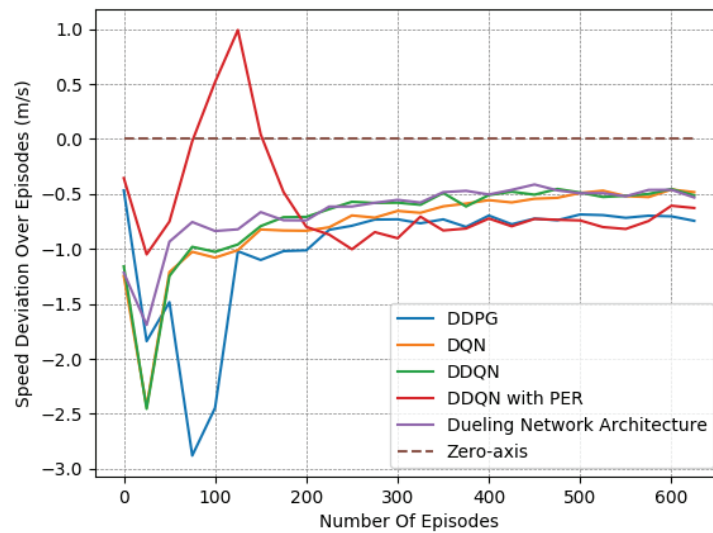


Figure 18. Speed deviation over time for the “Fields, Zones, Overtake and Avoid Collision RF” with different DRL algorithms.

Table 8. Comparing different DRL Algorithms with the “Fields, Zones, Overtake and Avoid Collision RF”.

DRL Algorithms	DDPG	DQN	DDQN	DNA	PER
Collisions	0.449	0.952	1.094	1.04	0.792
Speed Dev. (m/s)	−0.723	−0.471	−0.483	−0.496	−0.617

It is apparent that DDPG tackles the problem at hand more efficiently, as it is a method that was designed for continuous action spaces. Meanwhile, DQN requires discretizing the action space, which may not lead to the ideal solution. Moreover, we attribute the improved performance to the complexity of the reward function and training environment, as DDPG typically tends to outperform DQN.

Summarizing, as mentioned already, to the best of our knowledge, this is one of the earliest endeavors to introduce the concept of deep reinforcement learning to the lane-free environment. Thus, the main focus here was not to deploy a “perfect” policy that eliminates collisions, but to examine the limitations and potential of DRL in a novel and evidently quite challenging domain. In this approach, the MDP formulation places the agent in a populated traffic environment, where the agent directly controls its acceleration values. This constitutes a low-level operation that renders the task of learning a driving policy much more difficult, since it forces the agent to learn to act in the 2-dimensional space, where speed and position change according to the underlying dynamics, and more importantly, without any fall-back mechanism or underlying control structures that address safety or stability. That is in contrast to other related approaches that do manage to provide experimental results [14] with zero collisions and smaller speed deviations. However, there, the focus is quite different, since the RL agent acts in a hybrid environment alongside a rule-based approach (see Section 2.4).

We tackled a very important problem, that of reward function design, which is key for the construction of effective and efficient DRL algorithms for this domain. These DRL algorithms can then be extended considering realistic hard constraints and fallback mechanisms, which are necessary for a real-world deployment. The proper employment of such constraints and mechanisms is even more crucial for algorithms that rely on deep learning (and machine learning in general), where explainability endeavors are still not mature enough [37]. As such, for a more realistic scenario, one should also design and incorporate underlying mechanisms that explicitly address safety and comfort, where the RL agent will then learn to act in compliance with the regulatory control structures.

5. Conclusions and Future Work

The main objective of this paper was the extensive design of reward functions for deep RL methods for autonomous vehicles in lane-free traffic environments. We formulated this particular autonomous driving problem as an MDP and introduced a set of reward components at various levels of information, which we subsequently combined to formulate different reward functions in order to tackle two key objectives in this domain: collision avoidance and targeting a specific speed of interest. We then thoroughly tested those reward functions for environments with varying difficulties, using a set of both discrete and continuous deep reinforcement learning algorithms.²

We performed a quantitative comparison of all the proposed reward variants and different DRL methods, in order to evaluate their respective performances, and provided insights for the employment of RL in the lane-free traffic domain. Our experiments verify that, given the appropriate reward function design, DRL can indeed be used for the effective training of autonomous vehicles operating in a lane-free traffic environment.

In essence, our work introduces the concept of deep reinforcement learning for lane-free traffic and opens up several avenues for future work in this domain. To begin, we aim to extend this work to even more complex lane-free traffic settings, such as on-ramp traffic environments under the lane-free paradigm. There, vehicles entering from the on-ramp need to appropriately merge onto the highway, and vehicles located in the main highway can potentially accommodate the merging operation. The incorporation of a learning-based approach in this setting, where strategic decision-making is important, constitutes an interesting future endeavor.

Moreover, different RL algorithms can be employed as learning methods for the problem at hand, such as PPO [38], which is a continuous, on-policy algorithm that has been shown empirically to provide better convergence and a better performance rate than most DRL algorithms. An additional potential technique to be examined is that of NAF [39], which can be described as dqn for continuous control tasks, and according to the authors outperformed DDPG on the majority of tasks [39].

Furthermore, one could consider adopting other noteworthy advancements from the DRL literature [3], such as the utilization of a different parameterizations of the state-action value function, similar to the one suggested by the authors. Another interesting endeavor would be to utilize DRL techniques that explicitly tackle multi-objective problems. Specifically, in [40], the authors proposed a method that could be effective for challenging problems, such as lane-free traffic, whose objectives can not be easily expressed using a scalar reward function, due to the complexity of the environment.

The proposed MDP formulation can also be paired with other methods that do not necessarily involve learning, such as Monte Carlo tree search (MCTS) [41]. MCTS could potentially be an alternative solution based on planning to the problem of autonomous driving in a lane-free traffic environment, using the proposed reward functions. We expect that delayed rewards will provide better policies in terms of the overall performance, at the expense of the required computational effort. A different future consideration is to address the multi-agent aspect of this problem, considering a lane-free environment consisting of multiple vehicles that learn a policy simultaneously using the proposed learning behaviors [4].

Finally, we believe that the incorporation of “safety modules” that regulate the agent’s behavior can result in a better balance between the collision avoidance and maintaining desired speed objectives. In our view, there are two candidate methodologies to achieve this: One is the incorporation of novel safe RL techniques that consider a set of “safe” states in the MDP formulation in which the agent is allowed to be, and utilizing optimization techniques to guarantee a safe policy [42,43]. Alternatively, one can consider adopting (and adapting) the responsibility-sensitive safety [44] model, which proposes a specific set of rules for autonomous vehicles that ensures safety. Regardless of methodology chosen, adding such safety modules to this novel deep RL for lane-free driving paradigm is essential for its eventual employment in real-world lane-free traffic.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, A.K., D.T. and G.C.; funding acquisition, M.P.; investigation, A.K. and D.T.; methodology, A.K., D.T., G.C. and M.P.; software, A.K. and D.T.; supervision, G.C. and M.P.; writing—original draft, A.K., D.T. and G.C.; writing—review and editing, A.K., D.T., G.C. and M.P. All authors reviewed the results and approved the final version of the manuscript.

Funding: The research leading to these results has received funding from the European Research Council under the European Union’s Horizon 2020 Research and Innovation programme/ERC Grant Agreement No. 833915, project TrafficFluid.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The associated code was developed for the TrafficFluid project [7] and cannot be shared as of now.

Conflicts of Interest: The authors declare no conflict of interest.

Notes

- ¹ Downstream vehicles refer to the vehicles located in front of the agent.
- ² Videos showcasing a trained agent with “Fields, Overtake and Avoid Collision RF” marked as “All-Components RF” can be found at: <https://bit.ly/3O0LjJW>.

References

1. Aradi, S. Survey of Deep Reinforcement Learning for Motion Planning of Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 740–759. [CrossRef]
2. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602.
3. Badia, A.P.; Piot, B.; Kapturowski, S.; Sprechmann, P.; Vitvitskyi, A.; Guo, Z.D.; Blundell, C. Agent57: Outperforming the Atari Human Benchmark. In Proceedings of the 37th International Conference on Machine Learning, Virtual Event, 13–18 July 2020; Volume 119, pp. 507–517.
4. Di, X.; Shi, R. A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to AI-guided driving policy learning. *Transp. Res. Part C Emerg. Technol.* **2021**, *125*, 103008. [CrossRef]
5. Kiran, B.R.; Sobh, I.; Talpaert, V.; Mannion, P.; Sallab, A.A.A.; Yogamani, S.; Pérez, P. Deep Reinforcement Learning for Autonomous Driving: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 4909–4926. [CrossRef]
6. Kendall, A.; Hawke, J.; Janz, D.; Mazur, P.; Reda, D.; Allen, J.M.; Lam, V.D.; Bewley, A.; Shah, A. Learning to Drive in a Day. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 8248–8254.
7. Papageorgiou, M.; Mountakis, K.S.; Karafyllis, I.; Papamichail, I.; Wang, Y. Lane-Free Artificial-Fluid Concept for Vehicular Traffic. *Proc. IEEE* **2021**, *109*, 114–121. [CrossRef]
8. Troullinos, D.; Chalkiadakis, G.; Papamichail, I.; Papageorgiou, M. Collaborative Multiagent Decision Making for Lane-Free Autonomous Driving. In Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS ’21), Virtual, 3–7 May 2021; International Foundation for Autonomous Agents and Multiagent Systems: Richland, SC, USA, 2021; pp. 1335–1343.
9. Yanumula, V.K.; Typaldos, P.; Troullinos, D.; Malekzadeh, M.; Papamichail, I.; Papageorgiou, M. Optimal Path Planning for Connected and Automated Vehicles in Lane-free Traffic. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; pp. 3545–3552.
10. Karafyllis, I.; Theodosis, D.; Papageorgiou, M. Lyapunov-Based Two-Dimensional Cruise Control of Autonomous Vehicles on Lane-Free Roads. In Proceedings of the 60th IEEE Conference on Decision and Control (CDC2021), Austin, TX, USA, 14–17 December 2021; pp. 2683–2689.
11. Malekzadeh, M.; Manolis, D.; Papamichail, I.; Papageorgiou, M. Empirical Investigation of Properties of Lane-free Automated Vehicle Traffic. In Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 8–12 October 2022; pp. 2393–2400.
12. Naderi, M.; Papageorgiou, M.; Karafyllis, I.; Papamichail, I. Automated vehicle driving on large lane-free roundabouts. In Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 8–12 October 2022; pp. 1528–1535.
13. Karalakou, A.; Troullinos, D.; Chalkiadakis, G.; Papageorgiou, M. Deep RL reward function design for lane-free autonomous driving. In Proceedings of the 20th International Conference on Practical Applications of Agents and Multi-Agent Systems, L’Aquila, Italy, 13–15 July 2022.

14. Berahman, M.; Rostmai-Shahrbabaki, M.; Bogenberger, K. Driving Strategy for Vehicles in Lane-Free Traffic Environment Based on Deep Deterministic Policy Gradient and Artificial Forces. *IFAC-PapersOnLine* **2022**, *55*, 14–21. [[CrossRef](#)]
15. Bellman, R. A Markovian Decision Process. *J. Math. Mech.* **1957**, *6*, 679–684. [[CrossRef](#)]
16. Watkins, C.J.C.H. Learning from Delayed Rewards. Ph.D. Thesis, King’s College, Cambridge, UK, 1989.
17. Watkins, C.J.C.H.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [[CrossRef](#)]
18. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; The MIT Press: Cambridge, MA, USA, 2018.
19. van Hasselt, H.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-Learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30, pp. 2094–2100.
20. Hasselt, H. Double Q-learning. In *Advances in Neural Information Processing Systems*; Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., Culotta, A., Eds.; Curran Associates, Inc.: Vancouver, BC, Canada, 2010; Volume 23.
21. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized Experience Replay. In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, 2–4 May 2016.
22. Wang, Z.; Schaul, T.; Hessel, M.; Hasselt, H.; Lanctot, M.; Freitas, N. Dueling Network Architectures for Deep Reinforcement Learning. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; Balcan, M.F., Weinberger, K.Q., Eds.; PMLR: New York, NY, USA, 2016; Volume 48, pp. 1995–2003.
23. Baird, L.C., III. *Advantage Updating*; Technical Report WL-TR-93-1146; Wright Lab: Washington, DC, USA, 1993.
24. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, 2–4 May 2016;
25. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic Policy Gradient Algorithms. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; Xing, E.P., Jebara, T., Eds.; PMLR: Beijing, China, 2014; Volume 32, pp. 387–395.
26. Troullinos, D.; Chalkiadakis, G.; Samoladas, V.; Papageorgiou, M. Max-Sum with Quadrees for Decentralized Coordination in Continuous Domains. In Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, Vienna, Austria, 23–29 July 2022; pp. 518–526.
27. Bai, Z.; Shangguan, W.; Cai, B.; Chai, L. Deep Reinforcement Learning Based High-level Driving Behavior Decision-making Model in Heterogeneous Traffic. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 8600–8605.
28. Aradi, S.; Becsi, T.; Gaspar, P. Policy Gradient Based Reinforcement Learning Approach for Autonomous Highway Driving. In Proceedings of the 2018 IEEE Conference on Control Technology and Applications (CCTA), Copenhagen, Denmark, 21–24 August 2018; pp. 670–675.
29. Bacchiani, G.; Molinari, D.; Patander, M. Microscopic Traffic Simulation by Cooperative Multi-Agent Deep Reinforcement Learning. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS ’19), Montreal QC, Canada, 13–17 May 2019; International Foundation for Autonomous Agents and Multiagent Systems: Richland, SC, USA, 2019; pp. 1547–1555.
30. Kalantari, R.; Motro, M.; Ghosh, J.; Bhat, C. A distributed, collective intelligence framework for collision-free navigation through busy intersections. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 1378–1383.
31. Typaldos, P.; Papageorgiou, M.; Papamichail, I. Optimization-based path-planning for connected and non-connected automated vehicles. *Transp. Res. Part C Emerg. Technol.* **2022**, *134*, 103487. [[CrossRef](#)]
32. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv*, **2014** arXiv:1412.6980.
33. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
34. Keras. 2015. Available online: <https://keras.io> (accessed on 15 February 2022).
35. Wu, C.; Kreidieh, A.R.; Parvate, K.; Vinitzky, E.; Bayen, A.M. Flow: A Modular Learning Framework for Mixed Autonomy Traffic. *IEEE Trans. Robot.* **2021**, *38*, 1270–1286. [[CrossRef](#)]
36. Plappert, M. keras-rl. 2016. Available online: <https://github.com/keras-rl/keras-rl> (accessed on 15 February 2022).
37. Burkart, N.; Huber, M.F. A Survey on the Explainability of Supervised Machine Learning. *J. Artif. Int. Res.* **2021**, *70*, 245–317. [[CrossRef](#)]
38. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
39. Gu, S.; Lillicrap, T.; Sutskever, I.; Levine, S. Continuous Deep Q-Learning with Model-based Acceleration. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; Balcan, M.F., Weinberger, K.Q., Eds.; PMLR: New York, NY, USA, 2016; Volume 48, pp. 2829–2838.
40. Li, C.; Czarnecki, K. Urban Driving with Multi-Objective Deep Reinforcement Learning. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS ’19), Montreal QC, Canada, 13–17 May 2019; International Foundation for Autonomous Agents and Multiagent Systems: Richland, SC, USA, 2019; pp. 359–367.

41. Coulom, R. Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. In Proceedings of the Computers and Games, Turin, Italy, 29–31 May 2006; pp. 72–83.
42. Baheri, A.; Nagesh Rao, S.; Tseng, H.E.; Kolmanovsky, I.; Girard, A.; Filev, D. Deep Reinforcement Learning with Enhanced Safety for Autonomous Highway Driving. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1550–1555.
43. Peng, Z.; Li, Q.; Liu, C.; Zhou, B. Safe Driving via Expert Guided Policy Optimization. In Proceedings of the 5th Conference on Robot Learning, London, UK, 8–11 November 2021; Faust, A., Hsu, D., Neumann, G., Eds.; Volume 164, pp. 1554–1563.
44. Shalev-Shwartz, S.; Shammah, S.; Shashua, A. On a Formal Model of Safe and Scalable Self-driving Cars. *arXiv* **2017**, arXiv:1708.06374.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Engineering IoT-Based Open MAS for Large-Scale V2G/G2V †

Nikolaos I. Spanoudakis ^{1,*‡}, Charilaos Akasiadis ^{2,§}, Georgios Iatrakis ^{3‡} and Georgios Chalkiadakis ^{3,‡}

¹ School of Production Engineering & Management, Technical University of Crete, 73100 Chania, Greece

² Institute of Informatics & Telecommunications, National Centre for Scientific Research ‘Demokritos’, 15341 Athens, Greece; cakasiadis@iit.demokritos.gr

³ School of Electrical & Computer Engineering, Technical University of Crete, 73100 Chania, Greece; giatrakis@tuc.gr (G.I.); gchalkiadakis@tuc.gr (G.C.)

* Correspondence: nispanoudakis@tuc.gr; Tel.: +30-28210-37744

† This paper is an extended version of our paper published in PAAMS2022—The 21st International Conference on Practical Applications of Agents and Multi-Agent Systems, L’Aquila, Italy, 13–15 July 2022.

‡ Current address: University Campus, Technical University of Crete, 73100 Chania, Greece.

§ Current address: Patriarchou Grigoriou & Neapoleos 27, 15341 Aghia Paraskevi, Greece.

Abstract: In this paper, we aimed to demonstrate how to engineer Internet of Things (IoT)-based open multiagent systems (MASs). Specifically, we put forward an IoT/MAS architectural framework, along with a case study within the important and challenging-to-engineer vehicle-to-grid (V2G) and grid-to-vehicle (G2V) energy transfer problem domain. The proposed solution addresses the important non-functional requirement of scalability. To this end, we employed an open multiagent systems architecture, arranging agents as modular microservices that were interconnected via a multi-protocol Internet of Things platform. Our approach allows agents to view, offer, interconnect, and re-use their various strategies, mechanisms, or other algorithms as modular smart grid services, thus enabling their seamless integration into our MAS architecture, and enabling the solution of the challenging V2G/G2V problem. At the same time, our IoT-based implementation offers both direct applicability in real-world settings and advanced analytics capabilities via enabling digital twin models for smart grid ecosystems. We have described our MAS/IoT-based architecture in detail; validated its applicability via simulation experiments involving large numbers of heterogeneous agents, operating and interacting towards effective V2G/G2V; and studied the performance of various electric vehicle charging scheduling and V2G/G2V-incentivising electricity pricing algorithms. To engineer our solution, we used ASEME, a state-of-the-art methodology for multiagent systems using the Internet of Things. Our solution can be employed for the implementation of real-world prototypes to deliver large-scale V2G/G2V services, as well as for the testing of various schemes in simulation mode.

Keywords: internet of things (IoT); open multiagent systems; smart grid; engineering multiagent systems (EMASs); digital twin

Citation: Spanoudakis, N.I.; Akasiadis, C.; Iatrakis, G.; Chalkiadakis, G. Engineering IoT-Based Open MAS for Large-Scale V2G/G2V. *Systems* **2023**, *11*, 157. <https://doi.org/10.3390/systems11030157>

Academic Editors: Philippe Mathieu, Juan M. Corchado, Alfonso González-Briones and Fernando De la Prieta Pintado

Received: 4 February 2023

Revised: 10 March 2023

Accepted: 14 March 2023

Published: 19 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The smart grid [1] constitutes an important emerging application domain for artificial intelligence and multiagent systems (MAS). In the smart grid, energy and information both flow over electricity distribution and transmission networks in all possible directions. As such, buildings, as well as electric vehicles (EVs), become active energy consumers and/or producers, and the need for their effective integration into the system arises. Not only is the smart grid an electricity network with diverse consumers and producers, it is also a dynamic marketplace where heterogeneous devices appear and need to connect and interoperate [2,3]. To date, several smart grid-related business models and information system architectures have been proposed, but they do not always adhere to particular standards [4]. This is not surprising, given the fact that energy markets can differ in scale, i.e., they can be global, regional, or isolated; that they may be regulated or owned by a

public authorities or the private sector; that they can involve renewable energy sources or non-renewable sources; and finally that they can allow for dynamic pricing based on demand and offers.

As such, energy markets can be naturally viewed as open multiagent systems [5–7]: Participants are agents that can freely enter or exit the system at any time and who are (proactively) setting and pursuing their own (presumably diverse) agendas, goals, and business models. They are able to compete, adapt, or react to their ever-changing, dynamic environment [8]. Moreover, they are socially able: they can negotiate, argue, and partner with others in coalitions [9].

Currently, the Internet of Things (IoT) offers a networking layer that interconnects distributed resources, such as charging controllers, power meters, various sensors and actuators, and processing and decision-support services [3,10]. Given this state of affairs, heterogeneous resources are rendered interoperable by the IoT, since they are henceforth able to exchange information and also reconfigure the parameters that are crucial for their operation [11]. Thus, the actual deployment of such approaches is now possible.

An IoT-enabled smart grid digital twin can represent the running states of the multitude of underlying interconnected physical devices (e.g., smart meters, controllers, energy storage devices), and has the ability to continuously collect the respective sensor measurements. The actual per-device grid state can thus be made available to the operators in real-time [12,13]. This monitoring capability can be further expanded with predictive maintenance techniques, allowing for the detection of malfunctions even before they occur [14]. At the same time, having access to historical per-device measurements allows the post-hoc analysis and/or training of machine-learning models [13]. To this end, agents can enable the digital twins of (cyber-) physical objects by being able to represent all (physical) assets of the smart grid domain. Agents can also represent classes of producers/consumers or even prosumers (i.e., entities that can be both consumers or producers). Electric vehicles, in particular, act as consumers while charging, but can also be producers if they provide energy from their batteries back to the Grid.

However, existing smart grid approaches do not provide functional open prototypes offering features such as the above, nor do they adequately exploit existing engineering MAS research paradigms. This is especially true for our particular domain of focus in this paper, the vehicle-to-grid (V2G)/grid-to-vehicle (G2V) problem. Regardless of this, there is a real need for diverse agents representing stakeholders in an open system to be equipped with predefined protocols which they can use in order to interact [7]. Importantly, stakeholders also need to be able to enrich such protocols with their own goals and/or algorithms.

The objective of our work in this paper was to fulfil such requirements in the V2G/G2V domain, contributing a novel IoT-based open MAS architecture designed to meet the aforementioned objectives. The innovative aspects of our work are, on the one hand, the fact that by employing our architecture, according to their particular goals, the various stakeholders are able to develop new agents, or to re-use existing ones, as they see fit, to cover their needs. Moreover, on the other hand, by employing an IoT platform that supports multiple application-layer protocols, we ensure that new, diverse agents can connect to the system to offer their services and to exchange energy, given pricing mechanisms that are possibly dynamic—i.e., designed to adapt and fluctuate so as to promote system stability and reliability in a game-theoretic manner [15].

In particular, our system employs SYNAISTHISI, a research-oriented IoT platform deployed in docker containers, which allows agents to connect and communicate using the Message Queuing Telemetry Transport (MQTT) publish/subscribe protocol [16], among others. We demonstrate the validity of the approach via simulation experiments involving three different charging scheduling algorithms and two dynamic pricing mechanisms proposed in the recent V2G/G2V research literature.

In this paper we extend upon a recent study presented at the 20th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2022) [17] in

three ways. First, we emphasize the engineering aspect of the development of this system. We describe the application of the agent systems engineering methodology (ASEME) [18] and provide insights on the agents' architecture, as well as the interaction protocols that they use. The ASEME methodology was selected as it has been used in the past for IoT-based MAS development [19] and is also referred to in surveys of this application domain [20–22]. Second, we show explicitly how agents enable the digital twins of vehicles and other stakeholders in the smart grid domain. Third, we conduct and document more experiments, and compare the performances of two different dynamic pricing mechanisms.

The rest of this paper is structured as follows. Section 2 presents the necessary background and discusses related work. Section 3 then puts forward our V2G/G2V-specific MAS-based architecture and offers a detailed description of the roles of the various agents, and a presentation of their interactions. Following that, in Section 4 we discuss our system's development process in detail. In particular, we present the IoT communications infrastructure; the ASEME-based statechart description of the inter-agent protocols and intra-agent control models dictating the agents' interactions and behaviour respectively; and the various V2G/G2V agent strategies currently incorporated in our framework. In Section 5 we conduct a thorough experimental evaluation of our architectural framework, verifying its applicability within realistic use-case scenarios of interest. In Section 6 we discuss the benefits arising for various stakeholders from its potential adoption and real-world integration, also focusing on the digital twin for the smart grid. Finally, Section 7 concludes this paper, and outlines directions for future work.

2. Background

In this section, we provide the necessary background for our work in this study. This includes the concept of a smart grid and the V2G/G2V problem in particular, for which we provide an overview of previous works and the motivations for our own. We then discuss the simulators, the IoT technology, and the SYNIASTHISI platform that we employed. Subsequently, we focus on the method used for engineering IoT-enabled MASs.

2.1. Smart Grids and the V2G/G2V Problem

As EVs further penetrate energy markets globally, electricity demand patterns are subject to change at levels that might become disruptive to the stability and the reliability of the current electricity grids [23]. A way to mitigate this risk is by introducing "smart charging", or grid-to-vehicle (G2V) capabilities, according to which the charging of EVs can be delayed and can take place at later time intervals than immediately after connecting to a charger [24], seeking, e.g., those with more renewable production, with less demand from other EVs, or with better pricing. In the opposite direction to that of G2V, vehicle-to-Grid (V2G) approaches can benefit from the capability of EV batteries to store energy, and thus coordinate their discharging to support situations of energy supply shortage [25].

Since the smart grid consists of multiple individual and economically minded entities, it is natural to model it as a MAS [26]. MASs provide a number of benefits in contrast to their centralized counterparts, such as faster computation times and scalability, since processing is performed in a decentralized fashion, and private data are not required to be shared. To date, many simulation tools and prototypes have been proposed that put forward V2G capabilities. Such approaches may either analyze low-level technical details regarding the operation of EVs, or are integrated into environments that include the respective individual stakeholders. We now proceed to briefly review several representative such systems.

The study presented in [27] introduced EVLibSim, a Java-based simulator of the operation of charging stations. This tool offers a user interface (UI) that can be used to manage charging stations. Its capabilities include the creation, modification, and monitoring of charging stations given the application of particular scheduling algorithms. In addition to being used by domain experts to test potential scenarios of use, it focuses on charging stations only, without incorporating different types of stakeholders.

The work presented in [28] involved describing a MAS that supports the decision-making of EV drivers in regard to locating charging stations and charging opportunities in the city of Valencia. This system incorporates multimodal information from various sources, such as traffic monitoring systems, social networks, and pricing, in order to optimize the placement of charging stations. Such approaches are valuable during the design of charging infrastructure, but do not help in deciding what will happen next, after the infrastructure is deployed and becomes operational.

The survey presented in [8] involved a large-scale literature analysis of the MAS-based control of smart grids, providing information regarding the related technologies and standards, and the application of intelligent agents commercial projects. The authors in [29] proposed coalition formation techniques for EVs, providing services related to V2G and demand-side management. An MAS architecture was designed, and the implementation of simulations was performed, using the Java Agent Development Framework (JADE [30]¹). In that study, different kinds of intelligent agents were considered, i.e., EVs, aggregators that form EV coalitions, and a transmission system operator that acts as a mediator and regulator. Coalitions were formed with the objective of reaching minimum energy requirements for participating in the regulation market. However, this approach did not allow for more sophisticated selection processes, making it difficult to scale, and the presented evaluation involved only five EVs.

Another approach based on the use of JADE for the coordination of EV battery charging is that of [31]. In that study, individual EV driver preferences were taken into account, such as their willingness for V2G participation and the vehicle's charging availability. It was shown via experiments that the proposed MAS managed to satisfy EV owners' charging preferences individually, even in emergency conditions.

2.2. Frameworks and IoT-Based Real-World Trials

In recent years, great progress has been achieved with respect to the delivery of real-world trials that offer V2G/G2V and which might incorporate simulators as well. To begin, XBOS-V [32] is a software-based open platform that can be utilized for controlling the charging of EVs connected to small buildings. The implementation of the standardized communication method for V2G, ISO 15118², provides the connection specifications for chargers and EVs. Relevant approaches are the open charge point interface (OCPI), the open charge point protocol (OCPP), and the open smart charging protocol (OSCP) [33]. OpenV2G [34] provides the required modules to implement the V2G public key infrastructure, e.g., to guarantee security for the EV and charging station connections, and also to allow for simulations to be executed. Another approach, the grid-integrated electric mobility model (GEM) [35], simulates both electricity and mobility aspects. However, this approach only allows for analysis to be conducted at a higher level, without referring to charging station recommenders, for example, and other particular stakeholders. Another tool that can be used to manage the charging of batteries is ACN-Sim [36], which can be utilized by individual end-users but not by large-scale grid operators.

Regarding the IoT domain, SYNAISTHISI is a research-oriented platform composed of open-source frameworks that can host dockerized services and can also act as a translator between various application-layer protocols [16]. Service dockerization allows scalable deployments to occur independently of the underlying operating systems of the hosts. Furthermore, the platform's support for multiple protocols enables the orchestration of heterogeneous agents and services that may follow different implementation paths. Furthermore, the platform offers user authentication and authorization and restricts access to private and sensitive channels for the exchange of information, and also supports semantic annotations of exchanged information and available services.

2.3. Engineering MASs

The fields of agent-oriented software engineering and multiagent systems engineering have produced a wealth of abstractions, methods, and techniques for developing MASs. A

survey of this field is outside of the scope of this paper; however, the motivation for our choice of method was based on surveys in the field and in the selected platform (IoT).

In the development of our system, we followed the approach laid out by ASEME, the Agent Systems Engineering MEthodology [18]. ASEME can be naturally used in the design and modeling of IoT-based MAS systems, as well as in ambient intelligence applications [19–22]. It builds on statecharts and, more broadly, the unified modeling language (UML [37]) in order to perform system analysis and design models. It is agent-architecture- and agent-mental-model-independent, allowing the designer to select the architecture type and the mental attributes of the agent, thus supporting heterogeneous agent architectures. Moreover, ASEME puts forward a modular agent design approach and uses the so-called intra-agent and inter-agent control concepts. The former is implemented to coordinate the different modules that implement the agent's capabilities, thus determining its behavior, whereas the latter allows for the control of the society of agents by defining the protocols that govern its coordination.

Importantly, in agent communication, there typically exist predefined message sequences that can be applied in several situations that share the same communication pattern regardless of the application domain [30]. Such message sequences are defined as protocols.

ASEME uses two relatively common abstractions for modeling agents: capability and functionality. Busetta et al. [38] view capability as “a cluster of plans, beliefs, events and scoping rules over them”. Braubach et al. [39] extended this idea and proposed that capabilities can contain sub-capabilities and have at most one parent capability. They defined the agent concept as an extension of the capability concept, aggregating capabilities. In the Prometheus methodology [40], each functionality identified in the analysis phase ends up being mapped to a capability in the design phase. In the agent modeling language (AML) [41], capability is a concept used to model an abstraction of a behavior in terms of its inputs, outputs, pre-conditions, and post-conditions. The behavior is the software component, and its capabilities are the signatures of the methods that the behavior realizes, accompanied by the method's pre-conditions and post-conditions. This approach is similar to that of service-oriented architectures, and thus considers the agent as an aggregation of services.

In ASEME, the agent coordinates its capabilities in the intra-agent control model. Capabilities are themselves decomposed to simple activities. For instance, one capability of a personal assistant agent is the ability to locate an appropriate charging station for its user's car. This task can be decomposed to specific activities, e.g., one activity is finding out which charging stations are in the user's vicinity, which charging sockets are available at each of them, and their free slots. Another activity is ranking the available slots according to the user's preferences. After identifying the activities associated with a capability, the next step is to connect them to a specific functionality, i.e., a specific method, algorithm, technology, or technique. This is an important managerial task as each activity can be connected to different functionalities. For example, the decision-making activity of ranking the available charging slots may be connected to an argumentation theory, implying an argumentation-based method, or to a utility function, implying a multi-criteria decision analysis method.

The inter-agent control model defines the capability of an agent to participate as a role in a specific protocol. ASEME allows the seamless integration of the inter-agent control model in the intra-agent control model as they follow the same formalism—i.e., statecharts [42]. The statecharts formalism does not exhibit the limitations (limited scalability, explosion of states) posed by other formalisms such as Petri-nets [43]. Therefore, in this study, we used the statechart formalism to define our open protocols and design patterns. Finally, ASEME automatically generates portions of the agent code or provides guidelines for the programmers to transform their design models into implementation models.

3. System Architecture

In this section, we provide an overview of the architecture of our system. First, we provide an overview of the application domain and identify the stakeholders. Then, we perform an analysis of the problem domain and identify the agents. Finally, we focus on defining their interactions. We use the ASEME System Actors Goals (SAG) model to depict the agents and their goals.

3.1. Overview of the Application Domain

We are interested in the vehicle-to-grid (V2G) and grid-to-vehicle (G2V) energy transfer problem domain. The first activity proposed in the ASEME methodology is to identify the stakeholders and their goals. After studying the related literature, we focused on the following smart grid stakeholder types [44–46]:

- Electric vehicle (EV) owners, who are usually also the drivers of EVs. To effectively use their EVs, they need to book a place for charging them at appropriate stations, and they pay for such a service. They might even be interested in charging them at a lower price if the charging station could discharge their EV batteries to contribute to the network when prices are high, i.e., acting as prosumers. Here, we consider owners of both battery electric vehicles (BEV) and plug-in hybrid electric vehicles (PHEV).
- Charging station owners buy energy from producers to charge electric vehicles. In some cases they can utilize (partially) charged EVs by employing them as energy producers when network prices are high and then recharge the EVs later at a better rate.
- Electricity producers are typical (possibly renewable) energy producers. They sell power to the network at rates that are usually based on supply and demand. To compute the latter, they depend on electricity imbalance indicators, which are usually monitored by the global network operator.
- Electricity consumers are typical households, industries, and other buildings and their corresponding infrastructure.
- Station recommender service providers represent groups of stations and act as mediators between EVs and charging stations. EV owners depend on them to find stations that suit their schedule and preferences, and stations use them to reach out to customers. The represented groups of stations may belong at the same firm or may operate in the same region.
- Electricity imbalance providers can be network operators or government agencies that monitor the grid balance and calculate/predict the periods of electricity shortage and surplus.
- Mechanism designers are intermediate trusted third parties responsible for calculating dynamic prices and managing the various payments between the stakeholders listed above.

3.2. The Agent-Based Approach

Since these diverse stakeholders have their own goals and business needs, they can all be represented by software *agents* in a system aiming to automate their function, and this system can be a digital twin, allowing for their simulation and study. Agents are a suitable paradigm, as they have the following capabilities [47]:

- They are autonomous, meaning they can operate without the direct control of humans, and with at least some control over their own actions, their internal state, and resource consumption;
- social, They are able to interact with other agents—including humans—and can choose their collaborators;
- They are reactive, perceiving and responding in a timely fashion to changes in the environment, according to their goals; and
- They are proactive, exhibiting goal-directed behavior by taking the initiative, being purposeful, and not simply acting in response to changes in the environment.

Thus, autonomous agents can represent stakeholders without direct intervention, with agents being able to locate their best collaborators and take initiative in constant pursuit of their goals. This can be achieved by adopting different software implementations that put forward heterogeneous strategies, for example, some charging stations may strive to charge EVs as soon as they are connected, whereas others might behave in a different manner and choose to manage demand congestion [48]. Note, however, that modeling the low-level power grid details, such as feeders or distribution-line constraints, lies beyond the scope of our work.

Special attention needs to be paid to the station recommender stakeholder, whose main goal is to bring together charging stations and EV owners. Such tasks are typically undertaken by “middle” agents. The “middle” agent can have different roles in a multiagent system (MAS) [49], such as that of a matchmaker (who brings service providers in contact with service requesters, who then communicate to make the transaction), a broker or facilitator (who facilitates the transaction), or a mediator (a combination of the previous two, who brokers the transaction but also brings the buyer in contact with the seller). In our case, we use the latter approach, since the charging station agent may need to negotiate the charging details directly with the EV agent.

Taking these into account, we present the specific cooperation protocols and the high-level architecture that can be used to deliver a MAS V2G/G2V framework. Our approach can be used to investigate and evaluate different implementations and strategies that agents may incorporate in a real-world setting. The protocols we put forward are open, as they can be easily extended and tailored to capture a plethora of real-world cases. In contrast to previous studies, detailed descriptions and semantic schemes for each service are provided, which enable the functionalities that agents would request in such settings.

By incorporating our framework, algorithms of the designers’ choosing can be tested and compared, e.g., methods that generate recommendations for charging and that calculate charging schedules for large numbers of EVs, or alternative pricing schemes that might induce different effects in real-world use cases. The applicability of our approach is illustrated below by implementing a functional prototype that adopts the proposed architecture, and by using it to execute simulations of use-cases to demonstrate its overall functionality. An important feature of our implementation is that agents come as modular components, and thus can be easily augmented or even replaced with approaches that nevertheless follow the protocols that we have defined.

Our architecture assumes that agents exist within a microgrid infrastructure that can be linked to other segments of the smart grid via distribution and transmission networks. A microgrid can import power when its local generation falls short, and can export surplus energy to the grid for added producer profits, according to any energy market regulations in effect [2]. Figure 1 provides an overview of the agents and their interactions.

Specifically, the types of agents considered in the system include multiple (a) electric vehicle agents (EV), (b) charging station agents (CS), (c) electricity producing agents (EP), and (d) electricity-consuming agents (EC). We also assume the existence of a regulatory service, which may be for-profit private service, consisting of the following agents: a service aggregator or (i) a station recommender (SR), (ii) an electricity imbalance (EI) for load monitoring, and (iii) a mechanism design (MD) for the generation of electricity prices.

Figure 2 depicts these agents as actors with goals in the form of the System Actors Goals (SAG) model of ASEME. The SAG model for the requirements analysis phase includes a graph with actors and their goals. The goal of one actor (the owner of the goal) may depend on another actor (collaborator) to be realized. In the figure, we can see the actors represented as cyan circles and the goals as yellow rounded rectangles. The dependencies are shown as arrows directed from the owner to the goal and from the goal to the collaborator(s). In the analysis phase, the particular goals are analyzed and represented by capabilities.

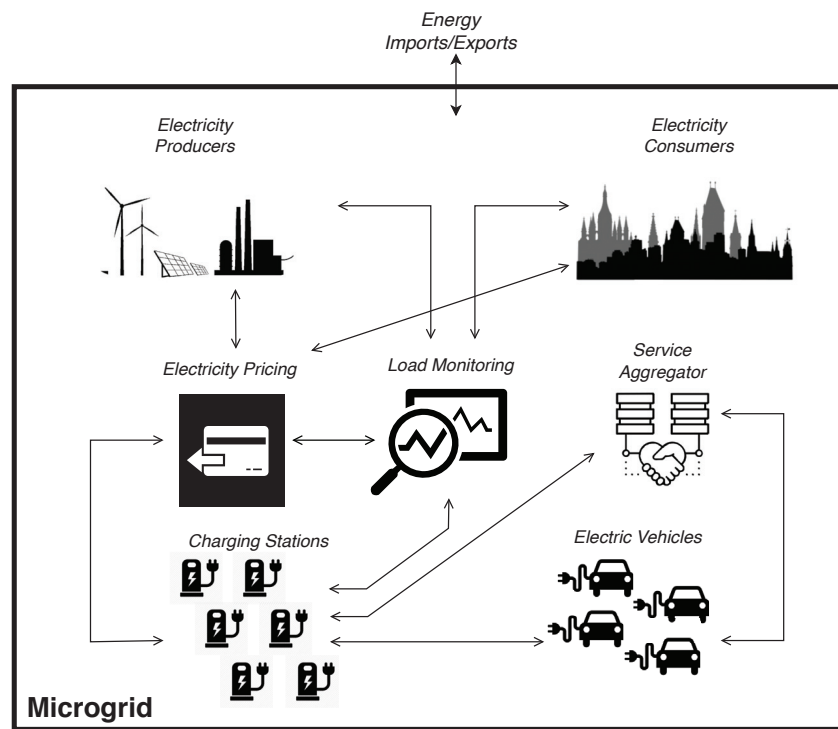


Figure 1. High-level overview of the V2G/G2V stakeholders and their interactions.

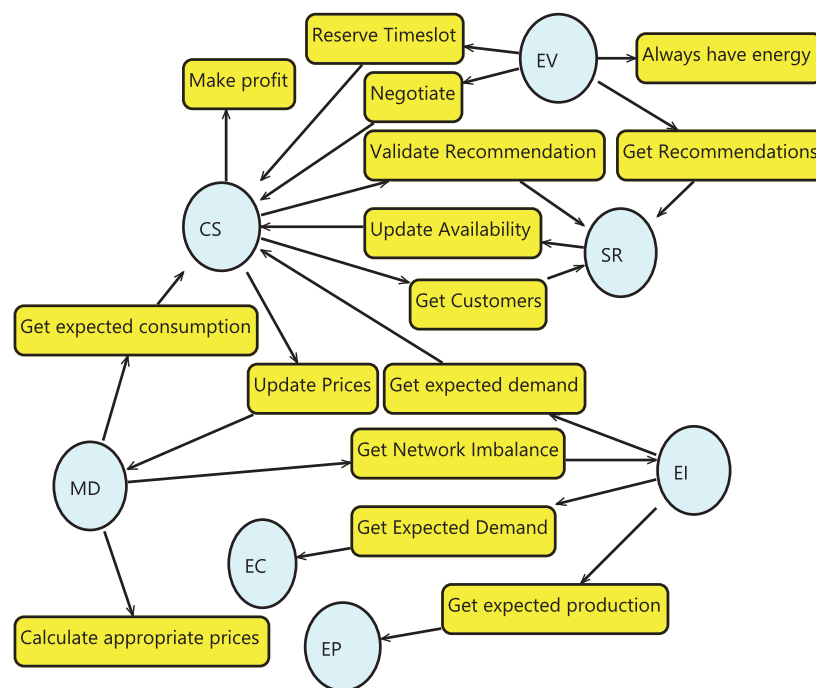


Figure 2. The System Actors Goals model of ASEME.

EV agents typically optimize utility functions that are set by the owner of each vehicle. Examples of utility functions include ones that guarantee that the EV will constantly have enough charge in the battery to perform its next trip, or to achieve this with the minimum cost possible, etc. The EV agent can monitor driving behavior and extract the underlying models to predict probable future activities and corresponding needs.

EVs may also communicate with a charging station to schedule a charging session or seek profit by participating in V2G and engage in negotiations with charging stations.

Such an agent might also consist of submodules, such as components for driver preference elicitation that monitor the typical habits and behavior of the driver, and possibly even forecast future preferences, or interoperable user interfaces, attainable either via a mobile device or the dashboard of the vehicle, which can be utilized by humans to operate respective procedures and monitor their conduct, for example, payments and negotiations, or simply to browse and select recommendations.

Furthermore, EV agents can adopt alternative tactics to automatically decide upon a charging schedule based on the defined preferences of each driver, such as cost reductions, faster access, preferred charging networks, location-based choices, and so on. An *EV agent* typically communicates with a *SR agent* with the aim of acquiring recommendations and with *CS agents* to reserve charging slots.

Subsequently, the *CS Agents* regulate the physical access points (e.g., connectors, parking spaces) through which EVs can connect to the grid, and may also earn revenue by charging their batteries. They communicate with EV agents regarding existing charging agreements and modify certain parameters to accommodate the charging of extra vehicles, in order to improve the utilization of the station infrastructure and generate increased profits. A *CS Agent* may encompass a charging scheduling module responsible for scheduling charging/discharging activities over a predetermined timeframe, a negotiation decision-making module for negotiating, a pricing module that calculates costs and payments, and a preference elicitation module that monitors the usage of charging slots and adjusts prices based on the station owner's needs. A *CS Agent* communicates with *SR*, *MD*, *EI*, and *EV* agents.

The *SR agent* notifies EVs with recommendations regarding a subset of the available CS and charging slots that match the most with their preferences, according to, e.g., the charging duration and the driving distance. This agent can be also augmented to take into account various grid constraints, for example, to help avoid herding effects. It consists of a recommendations engine module that generates charging station recommendations, an EV repository module that stores information about the past EV behavior in order to utilize it for future recommendations, and a charging station repository of all the CSs that have registered with the service. It communicates with the *CS* and *EV* agents.

The *EI agent* aggregates data from the *EP*, *CS*, and *EC* agents regarding their future anticipated energy consumption/production profiles, and calculates the periods during which electricity is in a state of shortage or surplus. In turn, it communicates the levels of energy imbalance with every interested party, in order for them to plan and optimize consumption and production. It includes a constraint extraction module that may incorporate different measures and methods relevant in such a scenario, e.g., monitoring any technical or practical limitations of V2G applicability, power flows, etc. It also calculates electricity imbalances over a planning horizon. Additional repositories may contain the stations, producers, and consumers that participate in the scheme.

The *MD agent* corresponds to an intermediate trusted third-party entity, which undertakes to calculate dynamic prices and to manage the payments of the various contributor types. Its goal is to assign appropriate and perhaps even personalized rates for energy consumption and production by *CS*, *EC*, and *EP* agents. It can put forward pricing mechanisms in order to incentivize agents to be truthful regarding their statements of expected values, as well as their actual behavior.

Finally, the *EP* and *EC* agents forecast and periodically report their expected production and consumption levels, respectively, accompanied by confidence values for these forecasts. *EP* and *EC* agents also exchange information with the *EI* and *MD* agents. User interfaces can be considered important submodules, since they are required by every agent type in order to provide monitoring capabilities to the operators if fully autonomous operation is enabled, or to allow human intervention in other operation modes (e.g., semi-automatic or manual).

3.3. Agent Interactions

Figure 3 illustrates the agent types and the protocols that they use to enable cooperation. Note that the high-level goals shown in Figure 2 have been elaborated to reflect specific interactions, which are labeled with an identifier so that we can easily refer to them. Briefly, the *cooperation protocols* dictating agent interactions are as follows:

- CP1** Charging Recommendation: Initiated by an EV for the scheduling of a charging session. The EV submits its preference and current location to the SR and receives a list of recommended CSs, along with the available time slots.
- CP2** Charging Station Reservation: Following CP1, the EV uses CP2 to reserve the selected charging slot at the respective CS.
- CP3** Negotiation: An optional protocol, which may be initiated after CP2, whenever either the CS or the EV, for whatever reason, needs to reschedule a charging session that has been reserved.
- CP4** Charging Station Registration: This interaction is used to register new CSs into the system. According to it, the CS informs the MD, EI, and SR agents about the required specifications.
- CP5** Authenticate Recommendation: After CP2, the CS asks the SR for validation that in fact the SR was the one that proposed the particular matchmaking between the EV and the CS.
- CP6** Electricity Prices: This follows CP7 and involves the MD calculating updated prices and submit the new values to every CS.
- CP7** Electricity Imbalance: This immediately follows CP10 or CP8. In the case that the expected production or consumption levels change, the EI must broadcast the updated values to the MD and every CS.
- CP8** Charging Station Update Schedule: After CP5, the CS makes a reservation of the requested time slot and notifies the EI and the MD accordingly.
- CP9** Producer Consumer Registration: Registers new producers and consumers. New stakeholders must inform EI and MD about their types.
- CP10** Update Expected Production/Consumption: This is initiated periodically (e.g., at the beginning of each day). In this step, every producer and consumer agent informs the EI and MD agents regarding the coming day's expected production and consumption levels.
- CP11** Update Energy Profile Confidence: This is initiated periodically (e.g., at the beginning of each day). In this step, every producer and consumer agent informs the EI and MD agents regarding the confidence that accompanies the forecasts described in CP10.
- CP12** Update Station Availability: Following CP2, this interaction is used by the CSs to update their information for the SR regarding available charging slots after new reservations.

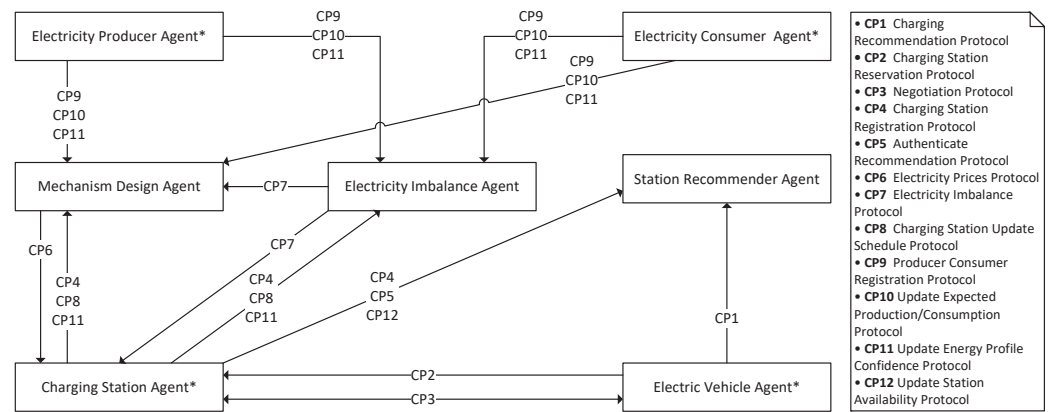


Figure 3. The proposed architecture. (*) denotes agent types with multiple instances. Arrows start from the agent that initiates the interaction and point to the receiver agents.

4. System Development

In the following, we first discuss the communications and deployment infrastructure. This enables the reader to understand the terminology behind the agent communication protocol definitions that are subsequently presented, using the language of statecharts. Then, we discuss the development of the agent models. We give an example in each case. Finally, we briefly describe the implemented strategies for the mechanism design agent and for the charging schedulers.

4.1. Communication Using the IoT Platform

Here, we describe the IoT platform that is used for agent communications and the incorporated cooperation protocols. Our implementation is based on the SYNAISTHISI platform [16], but any other solution that would offer the desirable features that we analyze below could be incorporated as well. We chose this particular platform for a number of reasons. First, it is offered with a non-commercial license and is mainly oriented towards research. By design, end-users are allowed to onboard new software services of their own and can be combined into more complex application designs. Furthermore, from a technical perspective, the platform supports widely used application-layer protocols (MQTT³, HTTP/REST⁴, etc.), along with translations of messages from one protocol to another. Importantly, the platform is deployed in docker containers, making it portable, interoperable with other software, as well as being scalable for large-scale deployments. Finally, to secure real-world deployments, user authentication and authorization processes are integrated in order to prevent unauthorized access to private information, such as locations, schedules, or other personal data that might be required to be shared for the purposes of V2G/G2V operations.

According to the modular service design paradigm, the interconnection among services is performed via an exchange of messages, for which we utilize the MQTT publish/-subscribe application-layer protocol in our design. A service can subscribe to topics to receive messages or publish to topics that other services have subscribed to in order to send information and commands. To access or transmit data through specific topics, however, the service owner must possess the necessary privileges, which can be managed through the platform’s graphical user interface (GUI). The same applies to the deployment and the execution monitoring of deployed services.

Mobile assets such as EVs that need to exchange messages are required to have wireless internet connections. For immobile objects, such as charging stations, supervisory control and data acquisition (SCADA) systems, etc., appropriate connectors can be interfaced with the platform, using either wireless or wired internet connections. The platform’s *broker* is responsible for notifying subscribers when a message is published.

4.2. Agent Interaction Protocols

The MAS cooperation protocols are defined here using the ASEME inter-agent control (EAC) model. To illustrate the process, we present the charging recommendation protocol (denoted as CP1 in Figure 3) defining the relevant interaction between an electric vehicle agent (EV) and a station recommender agent (SR) in Figure 4. The protocol is defined as a statechart (following the semantics of Harel [42] and the graphical model syntax of the ASEME statechart editor [19,50]) with the AND-state *CP1_ChargingRecommendation* as the root.

AND-states (depicted with a light blue color in the figure) contain OR states, and being in an AND-state entails being in all its OR-states simultaneously, implying that the latter are executed in parallel. OR-states (with yellow-colored labels) contain other sub-states, only one of which can be entered at any time. Basic states (shown in green) are where agent activities are executed. START-states show the beginning of the execution (black dots) and END-states show where execution ends (black dots within a circle). Transitions among states occur (i) when the activity of the source state is finalized and there is no event on the arrow, or (ii) when the event on the arrow takes place.

Returning to Figure 4, the two OR-states in the AND-state represent the participating agents—i.e., EV and SR. According to this protocol, the EV first enters the *SendRecommendationRequest* state and the SR enters the *ReceiveRecommendationRequest* state. The EV prepares its request by filling in the preferences and location data structures and publishes it to the broker. Via the latter action, the *publish* (“EV+/RequestChargingRecommendations”, [preferences, location]) event takes place and the EV transitions to the *ReceiveRecommendations* state to wait for the SR’s results. The ‘+’ sign is replaced by the agent’s ID, as each agent has its own topic for publishing requests.

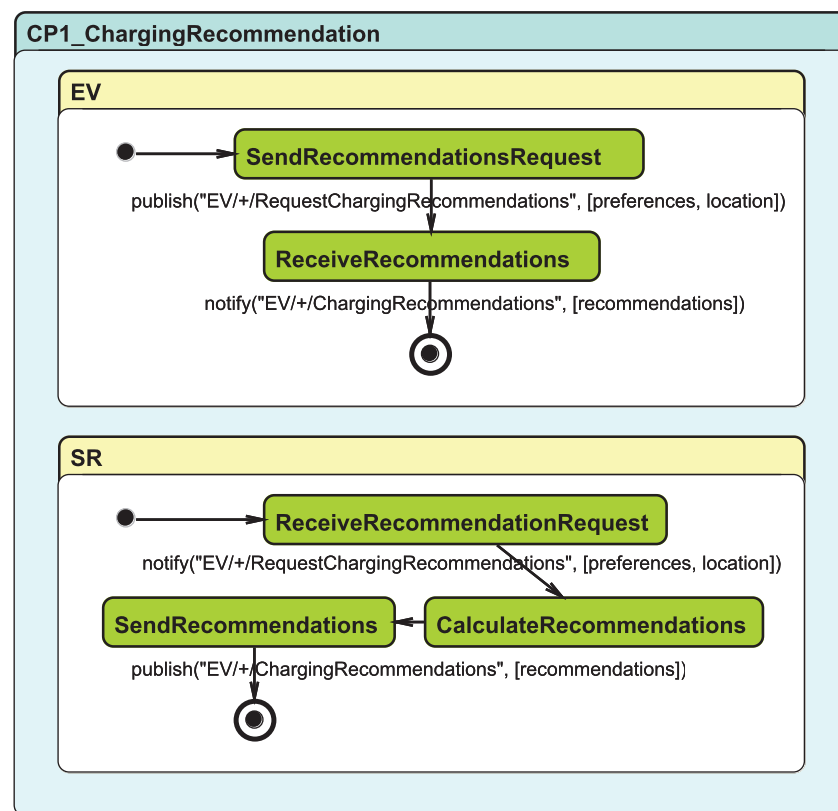


Figure 4. The charging recommendation protocol (CP1).

The broker receives the event and generates the relevant notification for the SR. The SR receives the event and enters the *CalculateRecommendations* state to find the best options for the EV. As soon as it completes this process, it enters the *SendRecommendations* state,

where it sends its reply by publishing a message to the appropriate topic. The EV is notified and the protocol is finished for both agents.

All the protocols shown in Figure 3 are defined using the ASEME EAC model. This allows the development of the protocols independently of the agents' development. The protocols are reusable modules and the agent developers can use them in order to ensure the agent's flawless participation in the open MAS.

4.3. Agent Model

The agent's are modeled using the intra-agent control (IAC) model in ASEME, which is represented by a statechart as well. Thus, it is very easy for an agent to integrate into its model the capability of participating in an interaction protocol. The orthogonal component of the assumed role in the EAC model is inserted as in the IAC model.

To illustrate this process, we depict the IAC model for the EV agent in Figure 5 (we do not provide the transition expressions so as not to visually clutter the diagram). Note that to simplify representation, we show the protocol roles that the agent realizes as basic states. These can be expanded to the relevant OR-states in the respective protocols. For example, the *CP1_ChargingRecommendation:EV* BASIC-state must be replaced by the *EV* OR-state of Figure 4⁵.

At the beginning of its operation, the agent enters the *Init* state and is initialized. Then, the *Negotiate* and *Reserve* orthogonal components follow (Figure 5). Arriving at the *Reserve* component, it then transitions to the *DecideNextAction* basic state. There, the agent decides the charging preferences and the desired location.

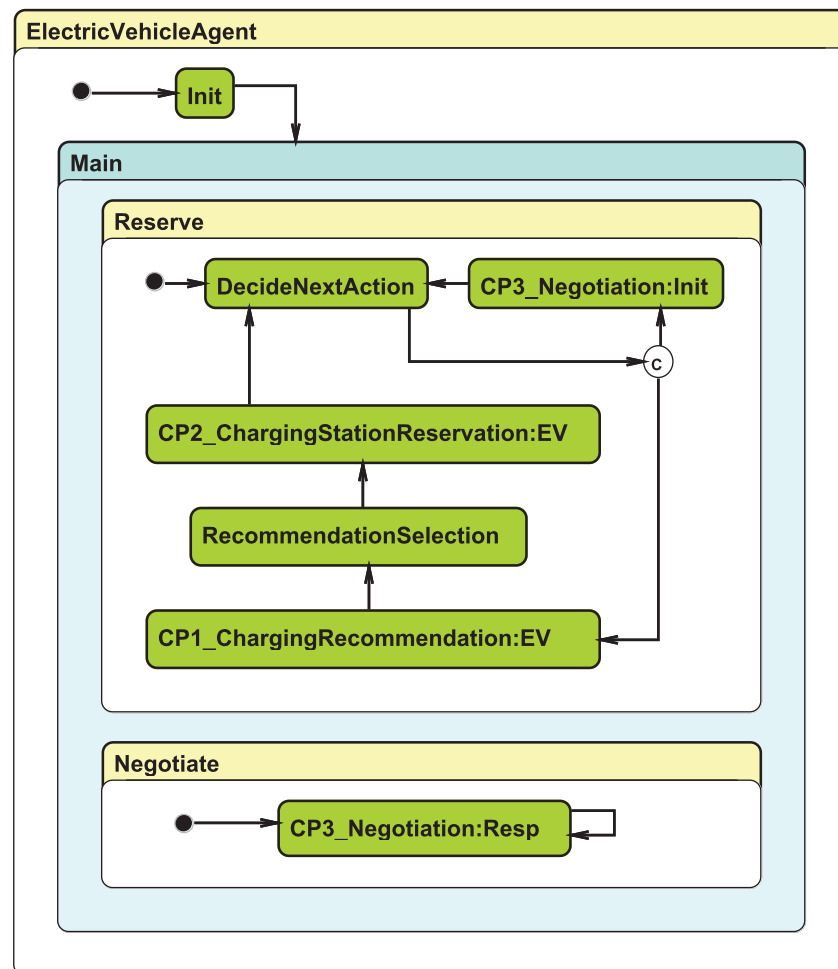


Figure 5. The intra-agent model of the electric vehicle agent.

Then, the *EV agent* has three different choices to make: (a) it monitors and predicts the battery state and driver preferences so as to autonomously decide if and how the charging will be arranged; (b) it receives required information from predefined datasets⁶; and (c) it may prompt the user via a GUI for charging preferences and manual protocol initiation. These three possibilities correspond to different implementations of the *DecideNextAction* state activity.

Whenever the agent decides to arrange a forthcoming charging process, it enters the *CP1_ChargingRecommendation:EV* state, then the *RecommendationSelection* state (to select the best offer), and finally the *CP2_ChargingStationReservation:EV* to reserve the selected slot. Then, it returns to the *DecideNextAction* state, from which it will have to transit in order to make a new reservation or to negotiate a change in an existing arrangement using the *CP3_Negotiation:Init* state.

As the negotiation protocol (CP3) can be initiated by both parties (EV or CS), the roles it defines are that of the initiator (*Init*) and that of the responder (*Resp*). As the reader can see, the EV can act either as an initiator (entering the *CP3_Negotiation:Init* state) or as responder (entering the *CP3_Negotiation:Resp* state). The latter merely sends a proposal, and if the CS agent replies, the rest of the negotiation process will be taken care of by the *NegotiationProtocol:resp* state (using the *responder* role of the respective protocol) at the *Negotiate* component. This is always executed in parallel to the *Reserve* component, as a CS agent could itself initiate a negotiation at any time.

4.4. Scenario Demonstration

Herein we present one system use-case scenario. In this scenario, an *EV* makes a reservation at a *CS* after receiving recommendations from the *SR*. In more detail, the agent interactions required for an *EV* to reserve a charging slot are depicted in the UML sequence diagram shown in Figure 6, which has been slightly simplified for ease of exposition. Note that these interactions require the execution of several protocols (in particular, CP1, CP2, CP5, CP6, CP7, CP8, and CP12), which are already provided in our implementation.

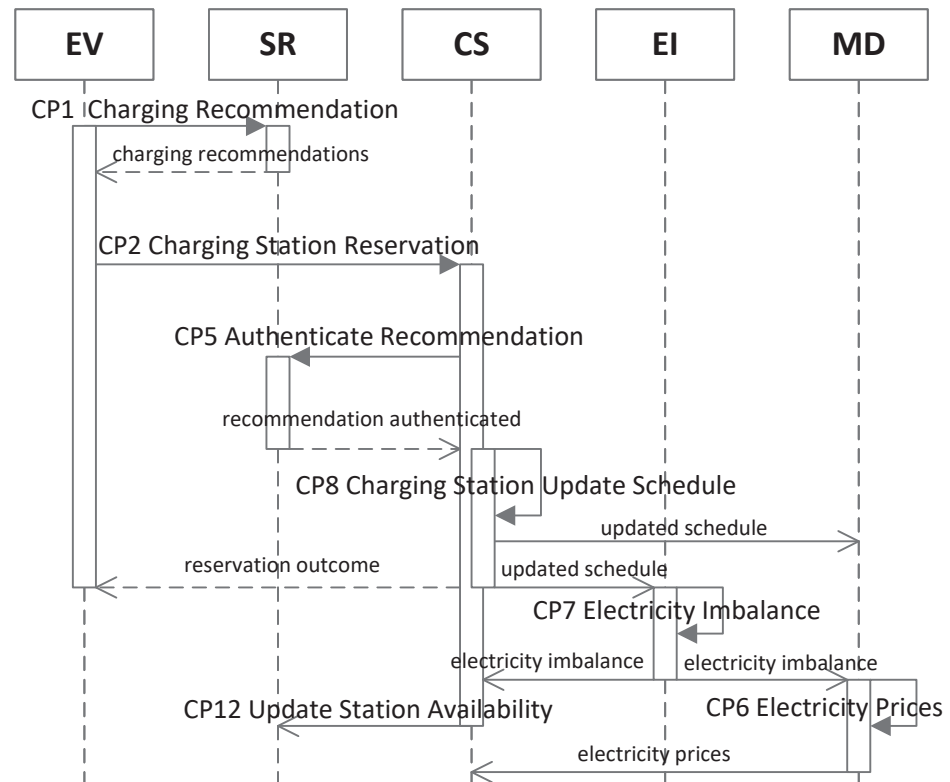


Figure 6. Agent interactions involved in reserving a charging slot.

The execution of this scenario is initiated when the EV agent sends a *CP1 Charging Recommendation* message, with its preferences and its location, to the SR. Then, the SR responds with the *charging recommendations*, which contain a list of the best-matched charging stations that are available for charging and that match the EV's preferences. The EV agent selects one charging station from the recommendations list and informs the CS of its selection by sending a tuple with the recommendation that was chosen and information about its battery state and the desired preferences, using *CP2 Charging Station Reservation*.

Once the CS agent has this information, it submits a *CP5 Authenticate Recommendation* to the SR in order to validate that the recommendation the EV selected is genuine. The SR checks the recommendations that it provided and responds accordingly to the CS agent; if the recommendation is valid, it notifies the CS agent that the *recommendation has been authenticated*.

Then, the CS calculates its new energy needs by performing a *CP8 Charging Station Update Schedule*, and sends the *updated schedule* to the EI and MD. Simultaneously, a *reservation outcome* is sent from the CS to the EV agent with the reservation information, the charging schedule, and the buy and sell prices for each time interval of the charging session.

The new reservation induces changes in the CS energy needs; thus, it sends a *Charging Station Update Schedule* to the EI and MD with the new consumption and production information. Then, the EI and MD respond with a *schedule update outcome* regarding their ability to record that CS's change.

The SR responds to the CS with an acknowledgment *availability update outcome*. Afterwards, the EI calculates the *CP7 Electricity Imbalance* for the time intervals that changed, and broadcasts an *electricity imbalance* with the updates to all CSs and the MD.

In turn, the CS informs the SR with a *CP12 Update Station Availability* about its new availability for charging slots. Finally, the MD executes the *CP6 Electricity Prices* protocol to calculate prices by taking into account the updated imbalance, and announces them to all CSs.

4.5. Implemented Agent Strategies

To validate the applicability of our framework, it was necessary to test the incorporation of different agent strategies and compare their effects on the ecosystem's behavior via simulations. For this purpose, we implemented two pricing algorithms that could have been used by an MD agent in the real world in order to test if and how they affected the stability of the grid, i.e., whether they led to more balanced production and consumption. We also implemented three charging scheduling approaches that were able to determine when and how much energy was exchanged between the CS and EV agents.

4.5.1. Electricity Price Calculation Algorithms Implemented by the Mechanism Design Agent

(A) NRG-Coin pricing algorithm: A mechanism inspired by [51], which aims to incentivize the participants to balance supply and demand. For its implementation, we used the parameter values as presented in [52]. In more detail, let the aggregate supply and demand at each time interval t be S_t , and D_t and the individual agent i 's desirable amounts of energy for selling and buying be s_t^i and d_t^i . The closer D_t and S_t are, the better prices are offered for buying and selling. The price for selling energy is:

$$P_t^{sell}(s_t^i, S_t, D_t) = (0.1 \cdot s_t^i) + \frac{0.2 \cdot s_t^i}{e^{(\frac{S_t - D_t}{D_t})^2}}$$

and the price for buying energy is:

$$P_t^{buy}(d_t^i, S_t, D_t) = \frac{(0.65 \cdot D_t) \cdot d_t^i}{D_t + S_t}$$

(B) Adaptive pricing algorithm: This is a pricing mechanism proposed by [53]. According to this mechanism, one estimates the evaluation of energy with respect to the cost induced by the EV agents by calculating an $\hat{\alpha}$ value:

$$\hat{\alpha} = \frac{\sum_{t=2}^N \frac{P_1^{buy} - P_t^{buy}}{2 \cdot (d_1^{i'} - d_t^{i'})}}{N - 1}$$

where N is the number of time intervals in the planning horizon, and $d_t^{i'}$ is the demand of EV agent i during the interval t . The mechanism can adjust prices to motivate agents to charge their EVs when there is an energy surplus on the grid. Buying prices for the intervals $t \in \{1, \dots, T\}$ are given by:

$$\hat{P}_1^{buy} - 2 \cdot \hat{\alpha} \cdot (S_1 - D_1) = \dots = \hat{P}_T^{buy} - 2 \cdot \hat{\alpha} \cdot (S_T - D_T)$$

Note that *adaptive pricing* does not determine prices for *selling* energy back to the grid—i.e., it does *not* support V2G activities.

4.5.2. Charging Scheduling Approaches

We now shift our focus to agent (EV) charging scheduling strategies, i.e., strategies tofordeciding the time intervals at which to charge the vehicles' batteries⁷. Specifically, the different charging scheduling methods that we tested in our simulations were the following:

(A) First slot: According to this method, EVs choose to charge their batteries immediately after they connect to a charger, regardless of how cheap or expensive electricity is at that particular time instant.

(B) Lowest Prices: In this case, EVs attempt to reduce overall costs by taking into account prices during the whole period that they are connected to a charger and end up selecting the intervals for which energy prices are the lowest.

(C) V2G : According to this approach, EVs are allowed to discharge their batteries and provide energy back to the grid considering high price time intervals, and select to charge it back at intervals with lower prices within the period of their connection to a charger. For this purpose and inspired by [26,57], we used linear programming to minimize an objective function representing charging costs in the presence of constraints regarding the EV preferences and charging specifications.

The cost function that is minimized is:

$$\min \sum_t^T C_t^{G2V} + C_t^{deg} - I_t^{V2G} \quad (1)$$

subject to:

$$C_t^{G2V} = d_t^{G2V} * P_{max}^{G2V} * p_t^{buy} * dt \quad (2)$$

$$C_t^{deg} = f^{deg} * d_t^{V2G} * P_{max}^{V2G} * dt \quad (3)$$

$$I_t^{V2G} = d_t^{V2G} * P_{max}^{V2G} * p_t^{sell} * dt \quad (4)$$

$$d_t^{G2V} + d_t^{V2G} \leq 1, \quad d_t^{G2V}, d_t^{V2G} \in [0, 1] \quad (5)$$

$$\sum_t^T d_t^{G2V} * P_{max}^{G2V} * dt - d_t^{V2G} * P_{max}^{V2G} * dt = E_{need} \quad (6)$$

$$\sum_t^k d_t^{G2V} * P_{max}^{G2V} * dt - d_t^{V2G} * P_{max}^{V2G} * dt + E_{init} \leq c_{max}, k \in [1, T] \quad (7)$$

$$\sum_t^k (d_t^{G2V} * P_{max}^{G2V} * dt - d_t^{V2G} * P_{max}^{V2G} * dt) + E_{init} \geq c_{min}, k \in [1, T] \quad (8)$$

where t is the charging interval of the charging period, C_t^{G2V} is the cost of charging, C_t^{deg} is the battery degradation cost, and I_t^{V2G} is the profit earned by selling energy to the grid. d_t^{G2V} and d_t^{V2G} are decision variables for G2V and V2G in our optimization problem and they can take values between zero and one, and intermediate values are assigned when it is optimal to charge or discharge at a fraction of the max charging (P_{max}^{G2V}) or discharging (P_{max}^{V2G}) power; p_t^{buy} and p_t^{sell} are the buying and selling prices of energy. f^{deg} is a degradation factor, based on the method presented in [58], which is used to evaluate the degradation cost C_t^{deg} , and dt is the duration of each time interval.

The constraints in expressions (5)–(8) must be satisfied during the EV scheduling optimization process. In (5), it is guaranteed that an EV will charge, discharge, or stay idle in each time interval. It can charge and then discharge in the same interval but this is unusual, because there must be a selling price greater than the buying price within the same interval. Constraint (6) states that at the end of the charging session the EV battery must be charged at the desired capacity E_{need} that the owner has set as a target. In constraints (7) and (8), we limit the allowable range of the battery charging state to be between the minimum (c_{min}) and the maximum (c_{max}) capacity by adding the net energy that has been received up to the end of each time interval, plus the initial amount of energy already stored in the battery.

5. Experimental Evaluation

In this section, we present four use-cases that showcase the real-world applicability of our solution. Through these use-cases, we evaluate and compare the implemented strategies that we discussed previously. The programming language that we chose to use was Python, and the datasets we utilized originated from a collection of real data from a number of publicly available online resources⁸; The simulation time horizon for each use-case was ten days The experiments were executed on a PC with an AMD Ryzen 5 1500X @ 3.5 GHz processor and 8 GB of RAM.

Overall, agents were implemented as different programs that were deployed in independent docker containers. Such containers could either be hosted on cloud infrastructure executed locally on the stakeholder's premises. Furthermore, to investigate the effects of different strategies, we set up simulations to test and evaluate particular desired algorithms. This was made possible via additional orchestrator scripts that utilized the API of the IoT platform to register, deploy, and configure services in batches. Moreover, the agent actions and final outcomes in the simulations were logged so that we could perform a post-hoc analysis of the results. For the purposes of the simulations, we also define the duration of a simulated hour in actual time, which in our experiments was set to two seconds. In an actual system deployment, the required data would be obtained in real-time via sensor measurements or user input forms. In our simulations, however, this information was retrieved from the datasets indicated above.

Our first use-case served the purpose of comparing the various EV charging scheduling methods, using the two pricing mechanisms that we described above. As explained above, (i) the *first slot* charging scheduling method involves charging the EV during the first slot when it is connected to a charger; (ii) the *lowest prices* method involves charging the EV during intervals with the lowest consumption price; and (iii) the *V2G* charging scheduling method allows an EV to also sell back to the grid some of its energy and recharge later, as long as it is ensured that it is profitable to do so (given the price difference between discharge and recharge intervals). Figure 7 depicts the average cumulative EV costs

for the entire planning horizon when the MD agent implemented the *NRG-Coin* pricing mechanism; whereas Figure 8 depicts these costs in the case in which the *adaptive* pricing mechanism was adopted. It is clear that, regardless of the pricing mechanism in use, the *first slot* method resulted in the highest costs for the EV. This was expected since, in this case, the EV agent chooses to charge their vehicle immediately, without taking into account the energy price. At the same time, by adopting the *lowest prices* method, the total cost of EV charging dropped by about 33% by the end of the time horizon for both pricing mechanisms examined. Regardless of the difference in the magnitude of the prices for the two mechanisms, for (*adaptive* responses to higher electricity prices), the drop in costs was relatively the same, and it was accrued via the better utilization of cheaply produced energy (e.g., from renewable sources). The difference in the absolute price values between the two mechanisms was not so important in our simulations since the two runs were independent from one another, and were subject to change according to the parametrization of the pricing functions that would have been adopted by real-world businesses. What is of interest is the relative difference between the prices of the time intervals for each mechanism, which, as we show in our third use-case below, ultimately affects the decisions of the EV agents in a similar manner. Finally, by allowing V2G operations, the charging costs dropped even more, being 15% lower than those of the *lowest prices* method, and 43% lower than those of the *first slot* method. This was not tested in the case of the *adaptive* pricing mechanism, since this mechanism was originally designed for smart charging and does not support V2G.

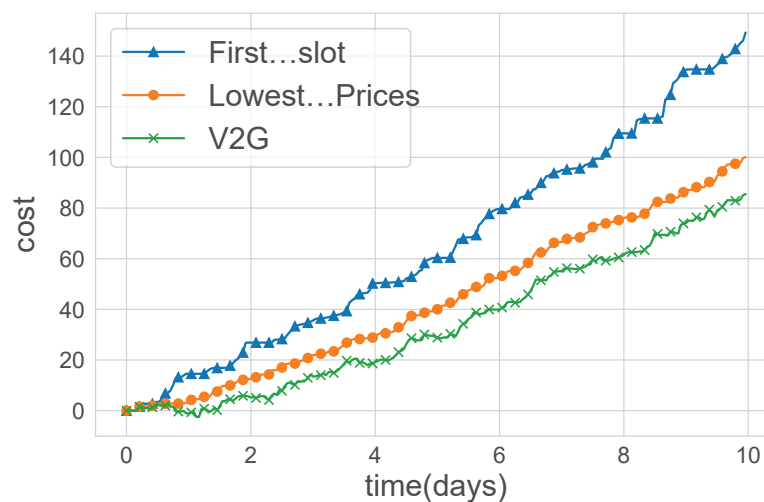


Figure 7. Average cumulative cost per EV for different charging scheduling methods (NRG-Coin pricing).

We then studied the impact of the charging scheduling methods on the aggregate energy imbalance. Our baseline was a grid imbalance without EV demand. We calculated the sum of the absolute imbalance values among the intervals, the sum of only the positive imbalance intervals (i.e., the total exported or “wasted” energy), and the sum of only the negative intervals (i.e., the total energy imports).

As seen in Table 1 we observed significant and differing impacts of different EV charging strategies on the energy imbalance. The employment of the *first slot* method by the EVs clearly affected the system negatively: the total imbalance was increased by 7%, and the imported energy increased by 104.2%, meaning that more than double the energy had to be produced to meet demand. At the same time, however, the EVs absorbed energy that would otherwise be wasted; thus, the corresponding amount dropped by 21.8%.

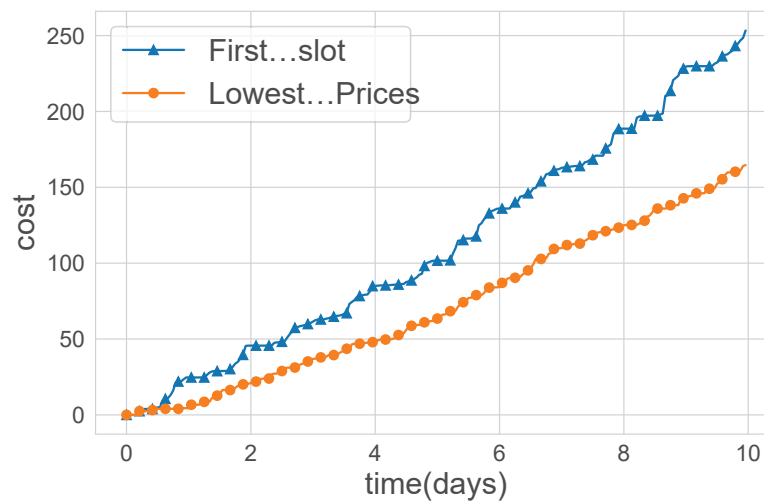


Figure 8. Average cumulative cost per EV for different charging scheduling methods (adaptive pricing).

In contrast, the use of the *lowest prices* method demonstrated a tangible positive effect on the system: there was a drop of 31.44% in the energy imbalance, whereas the amount of wasted energy was reduced by 45.6%. However, “imported” energy increased as well, albeit by a much smaller value, i.e., by 16.4%. This is because the (one hundred) EVs did introduce a significant demand that had to be met, whereas their charging strategy did not take the potential high energy prices into full account, nor did they contribute energy to tackle grid shortages.

Table 1. Energy differences in charging scheduling methods compared to the “no EV” baseline. The MAPE of the original imbalance curve was 63.9%.

Method	Imbalance	Wasted	Imported	MAPE
First Slot	+7.0%	−21.8%	+104.2%	−12.4%
Lowest Prices	−31.4%	−45.6%	+16.4%	−44.5%
V2G	−37.3%	−49.1%	+2.5%	−55.7%

An even more positive impact on the system imbalance was obtained when using the more “intelligent” V2G charging strategy. The EVs now optimized their charging/discharging plans, taking energy prices into full account and also contributing energy back to the grid. As a result, there was an even greater reduction in the grid imbalance of 37.3%. Moreover, the wasted energy was now reduced by 49.1% and there was only a slight increase of 2.5% in “imported” energy. Furthermore, this method demonstrated a reduction in the *mean absolute percentage error (MAPE)* that was significantly larger than those of the previous two methods. MAPE measures the difference between the induced imbalance and a totally flat curve with a value of zero, which resembles perfect matching between supply and demand. This is clear when plotting the imbalance across the time horizon for each method, as we have shown in Figure 9. Indeed, it is clearly visible there that the V2G strategy resulted in much lower induced peaks in the imbalance curve than those induced by *First Slot* or *Lowest Prices* methods.

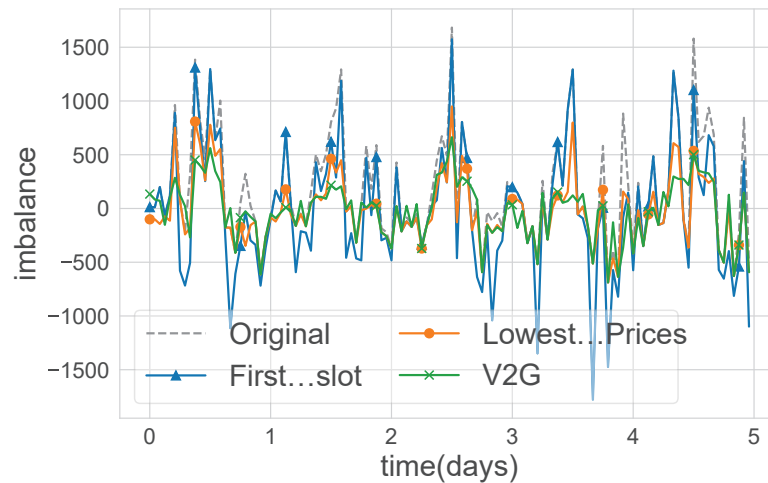


Figure 9. Tackling the energy imbalance using different charging scheduling methods.

In the second use-case, we measured the total cumulative costs of EVs when increasing the duration of their connection to chargers up to 12 h compared to the original data (i.e., without performing any charging rescheduling), following the three different charging scheduling methods as in the first use-case.

The results shown in Figures 10 and 11 demonstrate that by increasing the duration of connection, the *lowest prices* and *V2G* methods managed to gradually reduce the battery charging costs. This occurred since the longer an EV is connected to a charger, the greater the probability that it will be able to find the most advantageous intervals at which to buy energy from the grid—and also to sell it back to the grid in the case of V2G. As anticipated, again, the *V2G* method led to lower charging costs than the other two methods, and the difference (mirroring *V2G*'s advantages) increased as the duration of the connection to a charger became longer.

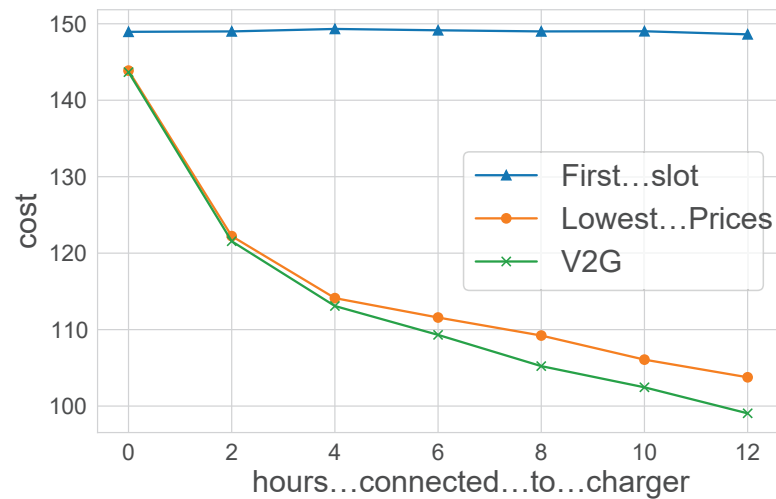


Figure 10. Cost comparison of varying time periods for which EVs were connected to chargers (NRG-Coin pricing).

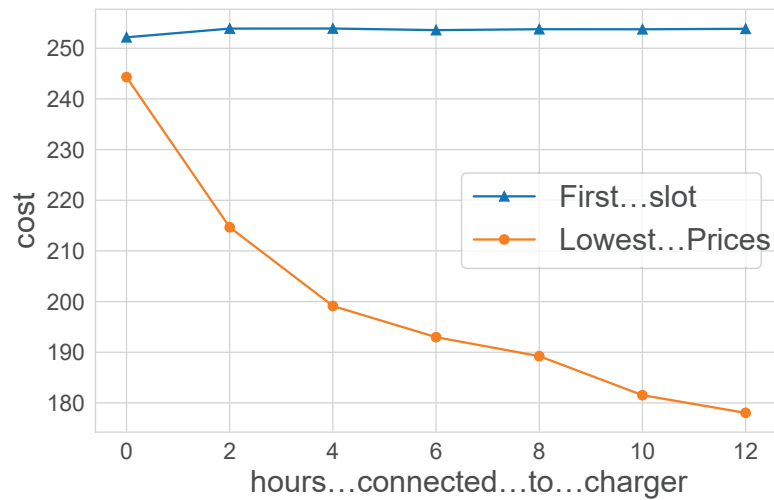


Figure 11. Cost comparison of varying time periods for which EVs were connected to chargers (adaptive pricing).

In our third use-case, we compare the performance of different pricing strategies for the MD agent. Specifically, we tested the *NRG-Coin* pricing and the *adaptive pricing* methods described in Section 4.5. Both algorithms aim to balance supply and demand by setting higher consumption prices during intervals in which there is a negative imbalance and lower consumption prices for intervals in which the imbalance is positive. In this use case, we assumed that EVs were charged following the *lowest prices* scheduling approach⁹. Assuming that EV agents were rational and aimed to reduce their expenses, the application of the two pricing algorithms resulted in demand being shifted to utilize the generated energy more effectively, thus leading to smaller peaks in the imbalance curve. Figure 12 shows that the algorithms had similar effects on the stability of the grid. In Table 2, we can observe similar behavior, namely, reducing the wasted energy, with this mechanism slightly outperforming the *NRG-Coin* pricing mechanism in terms of imported energy and MAPE reductions.

Table 2. Pricing Algorithms: energy differences compared to the “no EVs” baseline.

Method	Imbalance	Wasted	Imported	MAPE
NRG-Coin	−31.4%	−45.6%	+16.4%	−44.5%
Adaptive Pricing	−31.3%	−45.6%	+17.1%	−42.7%

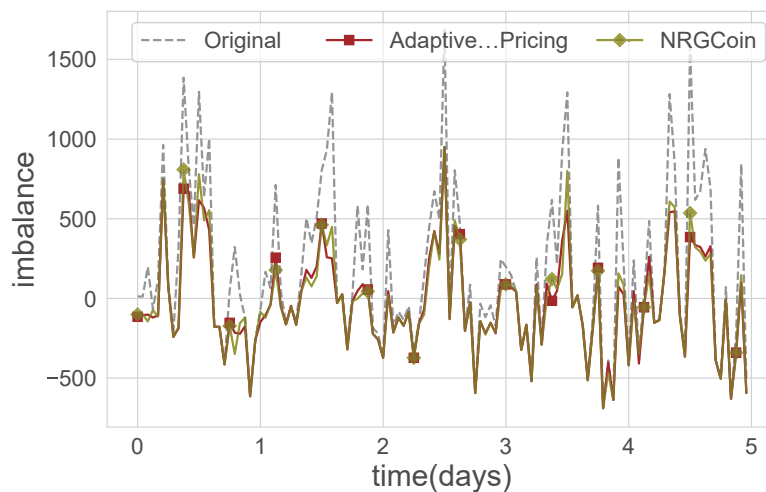


Figure 12. Comparison of adaptive pricing and NRG-Coin pricing mechanisms.

Finally, the fourth use-case was conducted to examine the scalability of our framework in terms of communication complexity as the supported EV population increased. To this end, we plotted in Figure 13 the total number of exchanged messages required for the scheduling of EV charging using our proposed cooperation protocols over a 10 day period, against increasing numbers of supported EVs. It is clear from Figure 13 that there was a *linear* increase in the number of messages exchanged (over 10 days) as the number of EVs increased. As such, this result attests to the scalability of our approach.

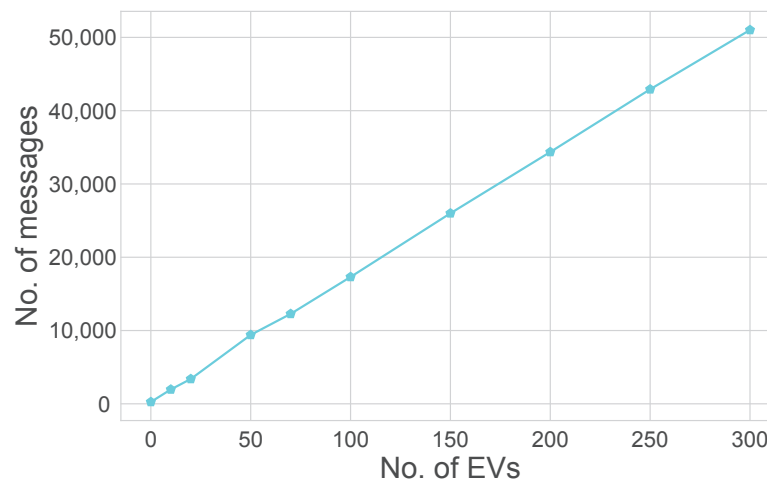


Figure 13. Message count for 10 days vs. the number of EVs.

6. Discussion: Enabling Digital Twins and Real-World Integration

In this study, we have taken several steps towards enabling the use of digital twins for V2G/G2V, and more generally for the smart grid domain, as well as enabling their real-world deployment. We know that currently there are a number of limitations in regard to the real-world application of V2G, e.g., the relatively small number of EVs, the existence of cheaper and better alternatives, high cost and complexity, consumer resistance, etc. [60]. The work presented in this study constitutes a step towards delivering V2G scenarios in the real world since it helps to reduce administration complexity and end-user costs, as we showed in our analysis.

First, we presented the inter-agent and intra-agent control models of ASEME that can be used for different development goals:

- The inter-agent control model (EAC): This can be shared with third parties. It contains both the activities (the basic states) and the topics that must be used by the agents to effectively participate in a protocol.
- The intra-agent control model (IAC): This is used for modeling the agent. It reuses parts of the EAC to ensure that the developed agents can be seamlessly integrated into the open MAS.
- Note that an EAC implementation can also be reused “as-is” by developers, who can use the same platform (e.g., Python) for developing their agents.

In energy markets, stakeholders develop their own business models and can have diverse goals in negotiating with regard to their consumption and offers, and can join and leave the system at any time. All these characteristics point to the relevance of *open multiagent systems* and agent technology in general [5,6]. These open interactions call for common ontologies, communication protocols, and suitable broker and coordination infrastructures to enable interoperability [61].

We have used existing standards (i.e., MQTT) to propose an architecture that is truly open, allowing players to reuse existing agents or build their own. The MAS V2G/G2V framework that we presented in Section 3.3 defines specific communication protocols among the stakeholders and is the basis of an implementation approach that allows for

the evaluation and comparison of different functionalities that could be offered by the various modules in such a setting. By focusing on openness, we allow the extension and customization of such protocols so that they comply with the diversity of real-world approaches. In contrast to previous studies, the semantic schemes of the services offered to agents have been described here in detail. Using our framework as a basis, designers may evaluate their own algorithms, e.g., for generating charging recommendations, for the scheduling of EV charging on a large scale, or for analyzing the effects of alternative pricing strategies. Taken together, the openness and extendability of our proposed architecture, along with its experimental evaluation in simulated settings, as presented in this paper, verify its appropriateness and potential for real-world integration.

We can also provide some guidelines for the deployment of the system. The first step is to determine the entity that will deploy, manage, and maintain the system's backbone, that is, the IoT platform. This entity can be any independent service provider, a power grid regulator, or a government agency, which will also be responsible for giving access to new users. Potential users include all the stakeholders that we have identified in our architecture (see Section 3). Each stakeholder may purchase or develop (or outsource) an application incorporating an agent that will represent them to the platform.

For example, car owners may download an appropriate application in which they can create a profile and connect to the platform. The complexity of the application can be determined based on their needs. A simple application, for example, would reserve a place after a user request. A more sophisticated application could also employ machine-learning techniques to learn the user's habits and automatically reserve a place when needed by informing its user.

Moreover, additional sensors and actuators must be interfaced with the IoT platform, allowing agents to receive measurements and submit actions. Examples include the sensory equipment of EV batteries, the charger controllers, and the various smart meters installed in the buildings. The EV agent could be deployed in an owner-controlled machine, e.g., inside the EV, and appropriate encryption could be established with regard to the messages exchanged in order to ensure the protection of private information. Privacy is also a concern for the other stakeholders in the ecosystem as they too exchange private data, for example, the buying and selling prices of each charging station, buildings' consumption profiles, and so on.

Therefore, our approach is readily deployable and can support real-world trials, thus enabling the use of digital twins in the smart grid domain [12,13]. The engineering approach that we followed possesses the following capabilities that are important in regard to digital twins [62]:

- It allows for synchronization between the physical world and the cyber domain. The design of our system is such that it can incorporate real-time updates even in simulation mode, which reflect the changes in the real world. For example, if a new charging station emerges, then a new CS agent appears in the system and starts pursuing its goals.
- It allows the co-simulation and modeling of subsystems. This V2G/G2V system could be considered as a subsystem for the overall smart grid, or as an instance of many interconnected smart grids. These grids could be hierarchical, i.e., the available power could be determined by a producer or by a higher authority that manages grids. Moreover, one of the participating agents, e.g., a charging station, could itself be a multi-agent system of charging connectors participating in the V2G/G2V system, in the form of one station that can accommodate many vehicles.
- The simulation mode can be used to test and compare various agent strategies, e.g., for coordinated charging/discharging [48,63] or for dealing with battery degradation issues [64,65], and in general for any approach that must be tested before it is finally deployed.

- Each resource is modeled as an agent, thus allowing the system to be scaled regardless of the complexity of interactions. The system scales linearly, as we demonstrated in our experimental evaluation (Figure 13).

Altogether, in comparison with the state of the art, in this study we have put forward a functional system that (a) enables large-scale V2G/G2V, (b) is supported by the use of a digital twin for simulations, (c) uses open protocols for easy adoption and realization by business stakeholders, and (d) allows each participant to adopt their own strategies and algorithms.

7. Conclusions and Future Work

In this paper we have demonstrate how to engineer an open system for the V2G/G2V energy transfer problem domain, and provided its architecture and the implementations of agents as flexible microservices that are interconnected by means of an IoT platform.

We illustrated the development process, starting by capturing the requirements of such a system by reviewing the stakeholders of the application domain and their goals. Then, using the ASEME methodology for IoT-enabled multiagent systems development [18,19], we proceeded to analyze the requirements, proposing the architecture, and then developing the prototype with the innovation of enabling the support of large-scale deployments using IoT technology.

We achieved our objective of proposing an open architecture [7] and od covering diverse business models via the definition of a number of key agent types and the development of open protocols. These types and protocols can be easily extended by any interested stakeholder, according to their needs. Our simulation experiments verify the applicability of our approach, and we have outlined the steps to be taken for its effective integration in the real world, along with the benefits arising from such an integration.

As the first item of our future work, we intend to populate the agents' components with actual machine-learning and recommender algorithms, in order to support the decision-making of agents in relation to various activities and tasks. Furthermore, the deployment and comparison (in simulation mode) of heterogeneous agent behaviors would be of significance, e.g., comparing various strategies for charging/discharging, or comparing different pricing mechanisms adopted by different charging networks. The choice of which behaviors to simulate could be performed according to the corresponding cultural and social values that are prevalent at different deployment sites [60,66].

Another interesting line of work would be to augment our solution with specialized graphical user interfaces. Such interfaces would be quite different for each agent role. For example, the interface for EVs would focus on usability for the elicitation of preferences, whereas that of the MD would focus on data and market analytics, and there could be different versions for the same agent type, as long as they all followed the protocols we defined via their back-end functionality.

Moreover, the openness of our architecture allows for the creation of alternative or additional protocols—e.g., protocols to serve the specific needs of various real-world stakeholders, and to help conceptualize and realize digital twins of actual real-world systems, of which the systematic analysis and the recognition of related opportunities and shortcomings are left as future works. Indeed, it is our aim to conduct a pilot, real-world study of our architectural framework. This will allow us to better evaluate its applicability and to identify interesting business models and necessary extensions of the framework.

The architecture can be extended to allow the incorporation of service-level agreements (SLAs) in the form of smart contracts between the different stakeholders. Smart contracts can define the obligations of the contracting parties, as well as issues related to the quality of service, such as performance, availability, and security [67]. The stakeholders' functionality, modeled using statecharts, allows for the automatic monitoring of the execution of SLAs and the handling of possible violations, e.g., with property statecharts [68] or Symboleo [69].

Finally, it would be interesting to employ our generic engineering approach to different application domains. For instance, this approach could be utilized within the domain of

digital twins for manufacturing, in which agent-based modeling with the use of statecharts has recently been proposed [62]. More generally, we believe that the ideas presented in this paper can be of use and tested in any domain that calls for the engineering of IoT-based open MAS architectural frameworks. In this direction, it would be interesting to develop a code generator for an IoT platform for ASEME models, similarly to the research conducted on an automatic code generator for the JADE framework [70].

Author Contributions: The authors confirm their contributions to the paper as follows: Conceptualization, N.I.S., C.A. and G.C.; investigation, N.I.S., C.A. and G.I.; methodology, N.I.S., C.A., G.I. and G.C.; software, N.I.S., C.A. and G.I.; validation, N.I.S., C.A. and G.I.; supervision, N.I.S. and G.C.; data curation, C.A. and G.I.; writing—original draft, N.I.S., C.A. and G.I.; writing—review and editing, N.I.S., C.A., G.I. and G.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The associated code is available online: <https://github.com/iatrakis/IoT-V2G-G2V> (accessed on 18 March 2023).

Conflicts of Interest: The authors declare no conflict of interest.

Notes

¹ <https://jade.tilab.com/> (accessed on 18 March 2023)

² <https://github.com/SwitchEV/RISE-V2G> (accessed on 18 March 2023).

³ The message queue transport telemetry (MQTT) protocol is an OASIS standard messaging protocol for the Internet of Things, mqtt.org <https://mqtt.org/> (accessed on 18 March 2023).

⁴ REpresentational State Transfer (REST) over Hypertext Transfer Protocol (HTTP).

⁵ More detailed descriptions of the inter- and intra-agent controls and a detailed description of the protocols, including the message syntax and semantics, can be found in our online repository: <https://github.com/iatrakis/IoT-V2G-G2V> (accessed on 18 March 2023).

⁶ This is very useful for experimentation with large agent populations.

⁷ Several battery charging models have been introduced in the past, considering load transfer constraints and mobility patterns (see, e.g., [54–56]). In our study, we do not require any particular models for calculating travel duration and battery SOC, since such values are acquired directly as sensor measurements. Of course, any of the battery charging models proposed to date could be incorporated in each individual agent implementation if deemed necessary by the strategy.

⁸ Specifically, consumption and production data originated from a synthetic dataset generator [59], which was trained on information from the ENTISOE <https://transparency.entsoe.eu> (accessed on 18 March 2023) platform, and on EV data from the MyElectricAvenue <https://eatechnology.com/resources/projects/my-electric-avenue/> (accessed on 18 March 2023) project.

⁹ This was selected because the *lowest prices* was shown in the first use-case to perform better than the *first slot* method. Note that using the top-scoring V2G scheduling method was not an available option, since one of the pricing methods we intended to evaluate in this third use-case, *adaptive pricing*, does not support V2G activities (cf. Section 4.5).

References

- Burke, M.J.; Stephens, J.C. Energy democracy: Goals and policy instruments for sociotechnical transitions. *Energy Res. Soc. Sci.* **2017**, *33*, 35–48. [CrossRef]
- Ketter, W.; Collins, J.; Reddy, P. Power TAC: A competitive economic simulation of the smart grid. *Energy Econ.* **2013**, *39*, 262–270. [CrossRef]
- Ghasempour, A. Internet of things in smart grid: Architecture, applications, services, key technologies, and challenges. *Inventions* **2019**, *4*, 22. [CrossRef]
- Espe, E.; Potdar, V.; Chang, E. Prosumer Communities and Relationships in Smart Grids: A Literature Review, Evolution and Future Directions. *Energies* **2018**, *11*, 2528. [CrossRef]
- Wooldridge, M.; Jennings, N.R. Intelligent Agents: Theory and Practice. *Knowl. Eng. Rev.* **1995**, *10*, 115–152. [CrossRef]
- Huynh, T.D.; Jennings, N.R.; Shadbolt, N.R. An integrated trust and reputation model for open multi-agent systems. *Auton. Agents-Multi-Agent Syst.* **2006**, *13*, 119–154. [CrossRef]

7. Hattab, S.; Chaari, W.L. A generic model for representing openness in multi-agent systems. *Knowl. Eng. Rev.* **2021**, *36*, e3. [[CrossRef](#)]
8. Mahela, O.P.; Khosravy, M.; Gupta, N.; Khan, B.; Alhelou, H.H.; Mahla, R.; Patel, N.; Siano, P. Comprehensive overview of multi-agent systems for controlling smart grids. *CSEE J. Power Energy Syst.* **2020**, *8*, 115–131.
9. Chalkiadakis, G.; Elkind, E.; Wooldridge, M. *Computational Aspects of Cooperative Game Theory*; Synthesis Lectures on Artificial Intelligence and Machine Learning; Morgan & Claypool Publishers: San Rafael, CA, USA, 2011.
10. Hossein Motlagh, N.; Mohammadrezaei, M.; Hunt, J.; Zakeri, B. Internet of Things (IoT) and the energy sector. *Energies* **2020**, *13*, 494. [[CrossRef](#)]
11. Noura, M.; Atiquzzaman, M.; Gaedke, M. Interoperability in internet of things: Taxonomies and open challenges. *Mob. Netw. Appl.* **2019**, *24*, 796–809. [[CrossRef](#)]
12. Brosinsky, C.; Westermann, D.; Krebs, R. Recent and prospective developments in power system control centers: Adapting the digital twin technology for application in power system control centers. In Proceedings of the 2018 IEEE International Energy Conference (ENERGYCON), Limassol, Cyprus, 3–7 June 2018; pp. 1–6.
13. Zhou, M.; Yan, J.; Feng, D. Digital twin framework and its application to power grid online analysis. *CSEE J. Power Energy Syst.* **2019**, *5*, 391–398. [[CrossRef](#)]
14. Shahinzadeh, H.; Moradi, J.; Gharehpetian, G.B.; Nafisi, H.; Abedi, M. IoT architecture for smart grids. In Proceedings of the 2019 International Conference on Protection and Automation of Power System (IPAPS), Tehran, Iran, 8–9 January 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 22–30.
15. Nisan, N. Introduction to Mechanism Design (for Computer Scientists). In *Algorithmic Game Theory*; Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V., Eds.; Cambridge University Press: Cambridge, UK, 2007; pp. 209–242.
16. Akasiadis, C.; Pitsilis, V.; Spyropoulos, C.D. A Multi-Protocol IoT Platform Based on Open-Source Frameworks. *Sensors* **2019**, *19*, 4217. [[CrossRef](#)]
17. Akasiadis, C.; Iatrakis, G.; Spanoudakis, N.; Chalkiadakis, G. An Open MAS/IoT-Based Architecture for Large-Scale V2G/G2V. In *The PAAMS Collection, Proceedings of the Advances in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation, L'Aquila, Italy, 13–15 July 2022*; Dignum, F., Mathieu, P., Corchado, J.M., De La Prieta, F., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 3–14.
18. Spanoudakis, N.I.; Moraitis, P. The ASEME Methodology. *Int. J.-Agent-Oriented Softw. Eng.* **2022**, *7*, 79–107. [[CrossRef](#)]
19. Spanoudakis, N.; Moraitis, P. Engineering Ambient Intelligence Systems Using Agent Technology. *IEEE Intell. Syst.* **2015**, *30*, 60–67. [[CrossRef](#)]
20. Zambonelli, F. Key Abstractions for IoT-Oriented Software Engineering. *IEEE Softw.* **2017**, *34*, 38–45. [[CrossRef](#)]
21. Savaglio, C.; Ganzha, M.; Paprzycki, M.; Bădică, C.; Ivanović, M.; Fortino, G. Agent-based Internet of Things: State-of-the-art and research challenges. *Future Gener. Comput. Syst.* **2020**, *102*, 1038–1053. [[CrossRef](#)]
22. Fortino, G.; Savaglio, C.; Spezzano, G.; Zhou, M. Internet of things as system of systems: A review of methodologies, frameworks, platforms, and tools. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *51*, 223–236. [[CrossRef](#)]
23. International Energy Agency. *Global EV Outlook: Towards Cross-Modal Electrification*; IEA: Paris, France, 2018.
24. Danner, D.; Duschl, W.; de Meer, H. Fair charging service allocation for electric vehicles in the power distribution grid. In Proceedings of the e-Energy '19, Phoenix, AZ, USA, 25–28 June 2019; pp. 406–408.
25. Sarkar, R.; Saha, P.K.; Mondal, S.; Mondal, A. Intelligent Scheduling of V2G, V2V, G2V Operations in a Smart Microgrid. In Proceedings of the e-Energy '20, Virtual Event, 22–26 June 2020; pp. 417–418.
26. Karfopoulos, E.L.; Hatziaargyriou, N.D. A Multi-Agent System for Controlled Charging of a Large Population of Electric Vehicles. *IEEE Trans. Power Syst.* **2013**, *28*, 1196–1204. [[CrossRef](#)]
27. Rigas, E.S.; Karapostolakis, S.; Bassiliades, N.; Ramchurn, S.D. EVLibSim: A tool for the simulation of electric vehicles' charging stations using the EVLib library. *Simul. Model. Pract. Theory* **2018**, *87*, 99–119. [[CrossRef](#)]
28. Jordán, J.; Palanca, J.; del Val, E.; Julian, V.; Botti, V. MASEV: A MAS for the Analysis of Electric Vehicle Charging Stations Location. In *The PAAMS Collection, Proceedings of the Advances in Practical Applications of Agents, Multi-Agent Systems, and Complexity, Toledo, Spain, 20–22 June 2018*; Demazeau, Y., An, B., Bajo, J., Fernández-Caballero, A., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 326–330.
29. Kamboj, S.; Kempton, W.; Decker, K.S. Deploying power grid-integrated electric vehicles as a multi-agent system. In Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, 2–6 May 2011; pp. 13–20.
30. Bellifemine, F.L.; Caire, G.; Greenwood, D. *Developing Multi-Agent Systems with JADE*; Wiley Series in Agent Technology; John Wiley and Sons Ltd.: Hoboken, NJ, USA, 2007.
31. Papadopoulos, P.; Jenkins, N.; Cipcigan, L.M.; Grau, I.; Zabala, E. Coordination of the Charging of Electric Vehicles Using a Multi-Agent System. *IEEE Trans. Smart Grid* **2013**, *4*, 1802–1809. [[CrossRef](#)]
32. Lipman, T.; Callaway, D.; Peffer, T.; von Meier Alexandra, E.A. *Open-Source, Open-Architecture Software Platform for Plug-In Electric Vehicle Smart Charging in California*; California Energy Commission: Sacramento, CA, USA, 2020.
33. Van Aubel, P.; Poll, E. Security of EV-charging protocols. *arXiv* **2022**, arXiv:2202.04631.
34. Käbisch, S.; Peintner, D.; Heuer, J.; Schmutzler, J.; Gröning, S.; Lauterbach, M. The OpenV2G Project. Available online: <https://openv2g.sourceforge.net> (accessed on 18 March 2023).

35. Sheppard, C.; Jenn, A. *Grid-Integrated Electric Mobility Model (GEM) v1.0*; US Department of Energy: Washington, DC, USA, 2021.
36. Lee, Z.; Johansson, D.; Low, S.H. ACN-Sim: An Open-Source Simulator for Data-Driven Electric Vehicle Charging Research. In Proceedings of the e-Energy '19, Phoenix, AZ, USA, 25–28 June 2019; ACM: New York, NY, USA, 2019; pp. 411–412.
37. *Unified Modeling Language, Superstructure, V2.1.2*; Technical Report formal/07-11-02; Object Management Group: Needham, MA, USA, 2007.
38. Busetta, P.; Howden, N.; Rönquist, R.; Hodgson, A. Structuring BDI agents in functional clusters. In Proceedings of the International Workshop on Agent Theories, Architectures, and Languages, Orlando, FL, USA, 15–17 July 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 277–289.
39. Braubach, L.; Pokahr, A.; Lamersdorf, W. Extending the capability concept for flexible BDI agent modularization. In Proceedings of the International Workshop on Programming Multi-Agent Systems, Utrecht, The Netherlands, 26 July 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 139–155.
40. Padgham, L.; Winikoff, M. *Developing Intelligent Agent Systems: A Practical Guide*; Wiley Series in Agent Technology; Wiley: Hoboken, NJ, USA, 2004; p. 240.
41. Trencansky, I.; Cervenka, R. Agent Modeling Language (AML): A comprehensive approach to modeling MAS. *Informatica* **2005**, *29*, 391–400.
42. Harel, D.; Naamad, A. The STATEMATE Semantics of Statecharts. *ACM Trans. Softw. Eng. Methodol.* **1996**, *5*, 293. [[CrossRef](#)]
43. Mazouzi, H.; Seghrouchni, A.E.F.; Haddad, S. Open Protocol Design for Complex Interactions in Multi-agent Systems. In Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2, AAMAS '02, Bologna, Italy, 15–19 July 2002; ACM: New York, NY, USA, 2002; pp. 517–526.
44. Rathor, S.K.; Saxena, D. Energy management system for smart grid: An overview and key issues. *Int. J. Energy Res.* **2020**, *44*, 4067–4109. [[CrossRef](#)]
45. Kabisch, S.; Schmitt, A.; Winter, M.; Heuer, J. Interconnections and Communications of Electric Vehicles and Smart Grids. In Proceedings of the 2010 First IEEE International Conference on Smart Grid Communications, Gaithersburg, MD, USA, 4–6 October 2010; pp. 161–166.
46. El-hawary, M.E. The Smart Grid—State-of-the-art and Future Trends. *Electr. Power Components Syst.* **2014**, *42*, 239–250. [[CrossRef](#)]
47. Weiss, G. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1999.
48. Carli, R.; Dotoli, M. A Distributed Control Algorithm for Optimal Charging of Electric Vehicle Fleets with Congestion Management. 15th IFAC Symposium on Control in Transportation Systems CTS 2018. *IFAC-PapersOnLine* **2018**, *51*, 373–378. [[CrossRef](#)]
49. Decker, K.; Sycara, K.; Williamson, M. Middle-agents for the internet. In Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI 97), Nagoya, Japan, 23–29 August 1997; pp. 578–583.
50. Spanoudakis, N.; Moraitis, P. An Agent Modeling Language Implementing Protocols through Capabilities. In Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Sydney, Australia, 9–12 December 2008; IEEE Computer Society: Washington, DC, USA, 2008; Volume 2, pp. 578–582.
51. Mihaylov, M.; Jurado, S.; Avellana, N.; van Moffaert, K.; de Abril, I.M.; Nowe, A. NRGcoin: Virtual currency for trading of renewable energy in smart grids. In Proceedings of the 11th International Conference on the EU energy market (EEM14), Krakow, Poland, 28–30 May 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–6.
52. Akasiadis, C.; Chalkiadakis, G. Decentralized Large-Scale Electricity Consumption Shifting by Prosumer Cooperatives. In Proceedings of the ECAL, Hague, The Netherlands, 29 August–2 September 2016; pp. 175–183.
53. Valogianni, K.; Ketter, W.; Collins, J.; Zhdanov, D. Sustainable electric vehicle charging using adaptive pricing. *Prod. Oper. Manag.* **2020**, *29*, 1550–1572. [[CrossRef](#)]
54. Koufakis, A.M.; Rigas, E.S.; Bassiliades, N.; Ramchurn, S.D. Offline and online electric vehicle charging scheduling with V2V energy transfer. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 2128–2138. [[CrossRef](#)]
55. Angelidakis, A.; Chalkiadakis, G. Factored MDPS for Optimal Prosumer Decision-Making. In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '15, Istanbul, Turkey, 4–8 May 2015; International Foundation for Autonomous Agents and Multiagent Systems: Richland, SC, USA, 2015; pp. 503–511.
56. Kelly, J.C.; MacDonald, J.S.; Keoleian, G.A. Time-dependent plug-in hybrid electric vehicle charging based on national driving patterns and demographics. *Appl. Energy* **2012**, *94*, 395–405. [[CrossRef](#)]
57. Liao, J.T.; Huang, H.W.; Yang, H.T.; Li, D. Decentralized V2G/G2V Scheduling of EV Charging Stations by Considering the Conversion Efficiency of Bidirectional Chargers. *Energies* **2021**, *14*, 962. [[CrossRef](#)]
58. Kempton, W.; Tomic, J.; Letendre, S.; Brooks, A.; Lipman, T. *Vehicle-to-Grid Power: Battery, Hybrid, and Fuel Cell Vehicles as Resources for Distributed Electric Power in California*; UC Davis Research Reports; UC Davis, Institute of Transportation Studies: Davis, CA, USA, 2001.
59. Charalambidis, G.; Akasiadis, C.; Rigas, E.S.; Chalkiadakis, G. A Realistic Dataset Generator for Smart Grid Ecosystems with Electric Vehicles. In Proceedings of the Thirteenth ACM International Conference on Future Energy Systems, e-Energy '22, Virtual Event, 28 June–1 July 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 432–433. [[CrossRef](#)]
60. Noel, L.; Zarazua de Rubens, G.; Kester, J.; Sovacool, B.K. Navigating expert skepticism and consumer distrust: Rethinking the barriers to vehicle-to-grid (V2G) in the Nordic region. *Transp. Policy* **2019**, *76*, 67–77. [[CrossRef](#)]

61. Weyns, D.; Michel, F. Agent Environments for Multi-agent Systems—A Research Roadmap. In *Proceedings of the Agent Environments for Multi-Agent Systems IV, Paris, France, 6 May 2014*; Weyns, D., Michel, F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 3–21.
62. Ragazzini, L.; Negri, E.; Fumagalli, L. Modelling Manufacturing Systems for Digital Twin Through Communicating Finite State Machines. In *Proceedings of the Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future, Valencia, Spain, 3–4 October 2019*; Borangiu, T., Trentesaux, D., Leitão, P., Eds.; Springer International Publishing: Cham, Switzerland, 2023; pp. 85–95.
63. Stüdtli, S.; Crisostomi, E.; Middleton, R.; Shorten, R. A flexible distributed framework for realising electric and plug-in hybrid vehicle charging policies. *Int. J. Control* **2012**, *85*, 1130–1145. [[CrossRef](#)]
64. Scarabaggio, P.; Carli, R.; Cavone, G.; Dotoli, M. Smart Control Strategies for Primary Frequency Regulation through Electric Vehicles: A Battery Degradation Perspective. *Energies* **2020**, *13*, 4586. [[CrossRef](#)]
65. Yan, G.; Liu, D.; Li, J.; Mu, G. A cost accounting method of the Li-ion battery energy storage system for frequency regulation considering the effect of life degradation. *Prot. Control Mod. Power Syst.* **2018**, *3*, 1–9. [[CrossRef](#)]
66. Rigas, E.; Akasiadis, C.; Vardaki, E.; Chalkiadakis, G. AI and Social Anthropology for Large-Scale Vehicle-to-Grid Schemes. In *Proceedings of the 12th Hellenic Conference on Artificial Intelligence, Corfu, Greece, 7–9 September 2022*; pp. 1–4.
67. Bunse, C.; Klingert, S.; Schulze, T. Greenslas: Supporting energy-efficiency through contracts. In *Proceedings of the Energy Efficient Data Centers: First International Workshop, E 2 DC 2012, Madrid, Spain, 8 May 2012*; Revised Selected Papers 1; Springer: Berlin/Heidelberg, Germany, 2012; pp. 54–68.
68. Mens, T.; Decan, A.; Spanoudakis, N.I. A method for testing and validating executable statechart models. *Softw. Syst. Model.* **2019**, *18*, 837–863. [[CrossRef](#)]
69. Sharifi, S.; Parvizimosaed, A.; Amyot, D.; Logrippo, L.; Mylopoulos, J. Symboleo: Towards a specification language for legal contracts. In *Proceedings of the 2020 IEEE 28th International Requirements Engineering Conference (RE), Zurich, Switzerland, 31 August–4 September 2020*; IEEE: Piscataway, NJ, USA, 2020; pp. 364–369.
70. Spanoudakis, N.; Moraitis, P. Modular JADE Agents Design and Implementation Using ASEME. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Washington, DC, USA, 31 August–3 September 2010*; Volume 2, pp. 221–228. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

The Hidden Side of Electro-Mobility: Modelling Agents' Financial Statements and Their Interactions with a European Focus

Jonatan J. Gómez Vilchez ^{1,*} and Roberto Pasqualino ²¹ European Commission, Joint Research Centre (JRC), 21027 Ispra, Italy² Department of Land Economy, University of Cambridge, Cambridge CB39EP, UK

* Correspondence: jonatan.gomez-vilchez@ec.europa.eu or gomez@te3modelling.eu; Tel.: +39-0332-78-53-18

Abstract: While much attention has been given, to date, to subsidies and taxes, the literature on the topic is yet to address less visible aspects of electro-mobility. These include the interactions among players, including money exchanges, and balance sheet issues. Analysing these is needed, as it helps identify additional mechanisms that may affect electro-mobility. This paper reports a modelling exercise that applies the system dynamics method, with its focus on stock and flow variables. The resulting simulation model captures the financial statements of several macro agents. The results show that the objective of the study is met: the model remains 'stock-flow consistent', meaning that assets and equity and liabilities balance out. By attaining this, the model serves as a coherent framework that makes the "hidden" side of electro-mobility visible, for the first time, based on current state-of-the-art, with the implication that it facilitates the analysis of potential financial factors that may either jeopardise or be conducive to faster road electrification. We conclude that the incorporation of the financial statements of key electro-mobility agents and their interlinkages in a simulation model is both a feasible and desired property for policy-relevant models.

Keywords: accounting framework; agent; automotive; electro-mobility; financial statement; simulation; stock flow consistency; system dynamics

Citation: Gómez Vilchez, J.J.; Pasqualino, R. The Hidden Side of Electro-Mobility: Modelling Agents' Financial Statements and Their Interactions with a European Focus. *Systems* **2023**, *11*, 132. <https://doi.org/10.3390/systems11030132>

Academic Editors: Philippe Mathieu, Juan M. Corchado, Alfonso González-Briones and Fernando De la Prieta Pintado

Received: 3 January 2023
Revised: 14 February 2023
Accepted: 16 February 2023
Published: 1 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Background

The process of road transport electrification is complex, and the rate of growth of electric vehicle (EV) sales and stock is dependent on the actions of several major players. Focusing on electro-mobility in Europe, the Powertrain Technology Transition Market Agent Model (PTTMAM) represents four of these, grouped as market agents: Authorities, Infrastructure Providers, Vehicle Manufacturers and Users [1].

In this model, each agent makes a series of decisions which have an influence on one or more of the other agents (see Figure 1). For instance, users demand vehicles that are produced by manufacturers. While these decisions were modelled for aspects that tend to be rather visible (e.g., authorities set emission targets and infrastructure providers deploy recharging infrastructure), other aspects that are thought to be important, as well, tend to be less visible (remain "hidden"). These include, prominently, the financial position of each agent and the money inflows and outflows that determine it.

For example, Regulation (EU) 2019/631 stipulates that "the amounts of the excess emissions premium should be considered as revenue for the general budget of the Union" and the European Commission should consider the feasibility of allocating them to a fund "to support re-skilling, up-skilling and other skills training of workers in the automotive sector" [2] (p. 19). This is clearly a potential transfer of money from vehicle manufacturers to a public institution and, in turn, possibly to certain households, directly or indirectly via manufacturers.

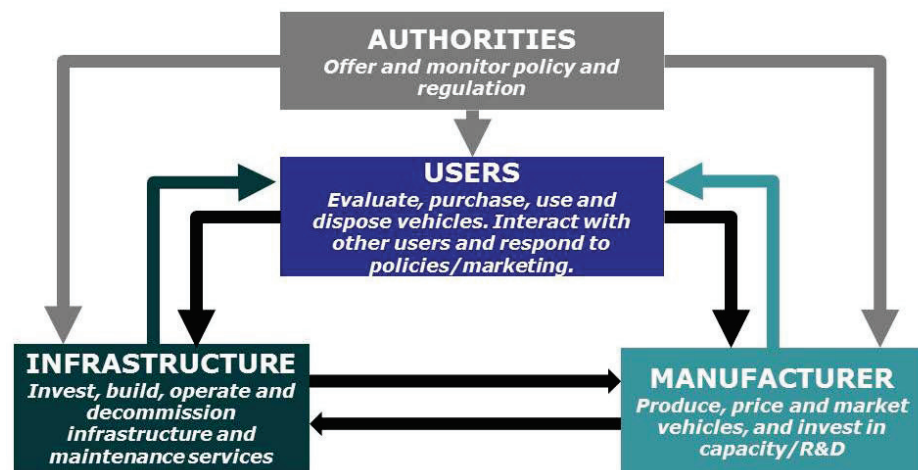


Figure 1. PTTMAM’s market agents, their decisions and linkages (Source: [3]).

Another example is the direct government support to original equipment manufacturers (OEMs). For instance, [4] reported EUR 295 million in government grants and subsidies in 2020, part of which was for alternative drive systems. [5] reported EUR 1 billion in government grants in the same year, though this OEM includes liquidity on favourable terms given by the European Central Bank (ECB) in that category.

A more indirect way of Member State (MS) support to vehicle manufacturers is as follows: the European Investment Bank (EIB) receives annual contributions from MSs; in turn, the EIB offers finance support to the automotive sector, as can be seen in Table 1. This table summarizes the evidence on EIB support to electro-mobility. Though the list is probably not exhaustive, it can yield a preliminary estimate of cumulative EIB finance support to the European automotive sector: ca. EUR 7 billion for research and development (R&D) actions and EV production support between 2010 and 2020.

Table 1. European Investment Bank’s finance support to the automotive sector: selected results on R&D and EV production.

Year	Name/Area	Firm	Finance [EURmio]
2010	R&D trucks	Daimler	400
2010	Environmentally friendly vehicles	Ford	450 *
2010	EV and battery production	Nissan	220
2011	Research to meet emission targets	Fiat	250
2011	Hybridisation	BMW	325
2011	R&D (electrification)	Renault	180
2012	R&D (emissions/safety)	Daimler	300
2013	Innovative technologies	BMW	400
2013	Research to meet emission targets	Renault	400
2013	FCEV	Daimler	400
2014	R&D trucks	Daimler	500
2014	Innovative powertrains	VW	500
2015	R&D (efficient engines)	FCA	600
2016	R&D (alternative fuel/hybrid engines)	FCA	250
2017	Innovative powertrains	PSA	250
2017	R&D (hybrid/electric)	Volvo Cars	245
2018	Hybrid and electric vehicles	FCA	420
2019	Electric motor development	PSA/NIDEC	145
2020	EV (BEV/PHEV) production	FCA (Stellantis)	300
2020	PHEV production/R&D (automation)	FCA (Stellantis)	485

* GBP. Source: own collection from [6].

1.2. Current State of the Research Field

The three examples mentioned in the previous section illustrate the importance of also taking into account the money flows among the key agents of the electro-mobility eco-system and, consequently, their financial positions. The extent to which this has been examined in the existing literature is considered in this section.

One of the most recent studies, which focuses on the electro-mobility sector and energy transition, applies hybrid econometric dynamic systems modelling based on Post-Keynesian economic theory, to address the energy transport transition in China, India, Japan, the United Kingdom and United States (US) [7]. Its objective is to demonstrate that the policy mix can support achieving tipping points where reinforcing learning and positive dynamics can support the transition relying on market driven forces alone. Similarly, [8] demands for economic modelling to investigate decarbonisation transitions at the global level, demanding financial incentives to the power and transport sector, still, with little ability to replicate the real structure of those sectors. In fact, those models apply policies while ignoring the “hidden” financial side of electro-mobility, leading to missing important leverage points that could support policy makers in addressing a faster energy and road transport transition.

Acknowledging that the literature on electro-mobility has expanded in the last few years, two previous review exercises reported by one of the authors are highlighted in [9], where system dynamics (SD) was identified as a prominent method to investigate the uptake of electric cars; in [10], a more in-depth review of a selection of SD models was performed. In comparison to earlier work (e.g., [11,12]), the current study extends previous modelling efforts by modelling balance sheets and assuring the stock-flow consistency (SFC) conditions, meaning an approach in which “the stocks and flows of both real and financial variables must be fully integrated, along with an explicit consideration of their dynamics” (Lavoie in [13]) (p. vii).

SFC modelling was prominently reported by [14]. The other author adopted the SFC modelling approach in previous work: [15] presented the Economic Risk, Resources and Environment (ERRE) model, which applies fossil fuels, agricultural and climate limits as boundaries to a growing economy. In ERRE, “a balance sheet approach is employed for every economic agent to assure stock-and-flow consistency in the economy as a whole” (p. 184).

The need to account for financial aspects when modelling the low-carbon energy transition was highlighted by [16]. Turning to the more recent literature dealing with SFC and the low-carbon energy transition, the transmission channels of climate finance policies (the carbon tax and green supporting factor) were examined by [17], who also identified reinforcing feedback processes when modelling six sectors/agents. Using a similar structure for agents, [18] also investigated carbon taxation, with the finding that it leads to a higher level of green investments as a result of financing by commercial banks. According to [19], “a crucial gap remains, preventing macro model-based analysis of financing barriers and policy interventions that may accelerate the energy transition”.

We identify a persistent research gap: the balance sheets of the key electro-mobility players have, to the best of our knowledge, not been coherently modelled. We think that this is relevant because the financial division of an OEM tends to account for a sizeable proportion of the firm’s balance sheet and to interact with other players such as authorities, banks and households. To the authors’ knowledge, this aspect is not yet covered in the literature, and we take this opportunity to address this inconsistency, as it can be relevant to policy targeting an acceleration of the road transport transition. We adopted the methodology indicated in the next sub-section (which lists additional materials) to address this gap.

1.3. Objective, Focus and Structure

The present paper describes the approach followed to create a simulation model that represents the financial statements of those players. This was a prerequisite for the objective of the study: capturing the interactions (in terms of money flows) among key market agents, while ensuring that the model remains ‘stock-flow consistent’.

The present study covers the global vehicle market, with a focus on the European conditions. For the purpose of this work, the following changes were implemented to earlier versions of the PTTMAM model [1]: (i) the ‘infrastructure providers’ were renamed ‘suppliers’, and ‘users’ became ‘households’; (ii) the banking sector was modelled by including a conglomerate of private/commercial banks (selling also vehicle insurance), the European Central Bank (ECB) and the European Investment Bank (EIB); and (iii) the last two represent new sub-agents of the market agent group ‘authorities’, which now includes government as a separate sub-agent.

The structure of the paper is as follows: after this short Introduction, the methods are described in Section 2. The results are reported in Section 3. Section 4 offers a discussion. An appendix completes this paper with additional info on the model.

2. Materials and Methods

2.1. Methodology

The methodology was implemented following a series of steps:

1. The financial statements of OEMs (i.e., vehicle manufacturers) were collected from their annual reports and data from the U.S. Securities and Exchange Commission filings. As for any other firm, the three main financial statements of OEMs are: a balance sheet, an income statement and a cash flow statement. Each of them facilitates the analysis of, respectively, solvency, profitability and liquidity. There are two broad economic and financial decisions firms need to make: investment (maximise profitability) and financing (minimise capital costs). Whereas the former is visible in the assets side of a balance sheet, the latter can be reflected in the equity and liabilities section [20]. Instead of profit maximising, “some companies may respond to daunting balance-sheet damage by minimizing debt” [21] (p. xii). This is a sufficient reason for us to model debt explicitly.
2. The statements were checked, their key elements identified, and the connections deduced. To support our task, the following materials were also used: [22–24].
3. Two balance sheet items (PPE and inventories) require an explanation, as their values can be affected by alternative assumptions. For each of them, we first read what the accounting rules are; we then check what has been assumed in previous SD work and identify what most OEMs adopt in practice.
4. The information collected from the previous steps was implemented in the simulation software environment ‘Vensim’. A reference that was used for the preliminary version is [25].
5. The variables for the initialisation of the model were created and initial values (see Table 2; money values are expressed in nominal terms) were assigned to those, so that the behaviour resulting from the structure represented in the model (step 4) could be simulated for the period 2005–2030. This was based on numerical integration using Euler, with a time step of 0.25 year (see, e.g., [26]).
6. The structure was refined until there were, a priori, no financial leakages in the modelled system.
7. The monetary structure was linked with the physical structure (e.g., revenues from selling vehicles) from which a series of key performance indicators (KPIs) could be computed.

The outlined methodological approach relies on the following steps. The initial part is statistical and includes the sourcing of relevant initial values. Furthermore, the ‘double-entry bookkeeping’ rule from the field of accounting [27] is imposed. The implementation in the software requires the use of stock and flow variables, which are connected in stock and flow structures (see, e.g., [28]). As stated by [29], stocks and flows are not independent, as the former result from the latter, and flows are also influenced by stocks. Thus, a two-way relationship between these variables may be posited. At the core of this methodology is the reliance on the SD method, which is well-suited for searching for the economy’s feedback structure [30]. Ref. [13] concluded that SD is a robust method to assure SFC conditions are met.

2.2. Model

2.2.1. Overview

Figure 2 provides an overview of the developed model. As can be seen, the private and public sectors interact in several ways, as transactions among agents take place in our simulated environment. Initial stock values do not reflect data collection but are assumed for the purpose of demonstrating the modelling framework. On model testing, see Appendix A. The simulation model is freely available at the webpage shown in the Supplementary Materials.

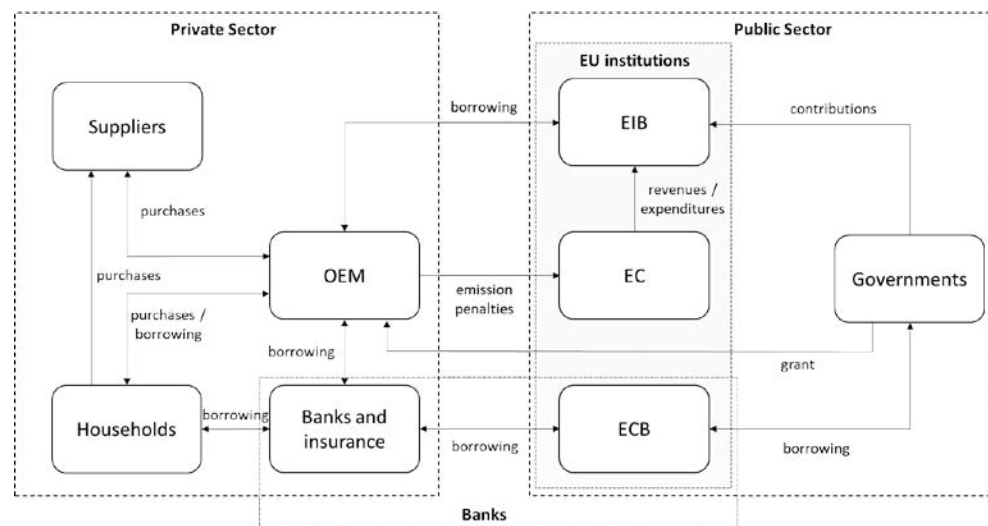


Figure 2. Overview of the conceptual model, with interactions among agents.

Focusing on the interlinkages among ‘Banks and insurance’, ‘Households’ (HHs) and ‘OEMs’, Figure 3 shows four balancing (B) feedback loops and five reinforcing (R) ones. They describe the debt dependency process, the amplifying nature of borrowing to acquire liquidity and pay interest, as well as the counteracting effects of debt adjustment and repayment. This is highlighted in Figure 3 as the “hidden” side of electro-mobility and forms a core component of this paper. At the top of the figure, a key aspect of electro-mobility is made visible, as EVs are still priced higher than conventional vehicles and thus require substantial upfront payments from HHs. While this causal loop diagram informed the modelling building process, it remains a simplification, as the link between borrowing from the OEM and vehicle purchase is mediated in the simulation model by the stock ‘wealth HH debtor’. Moreover, the figure provides a stylised overview, with ‘Banks and insurance’ (green area) overlapping the other two agents.

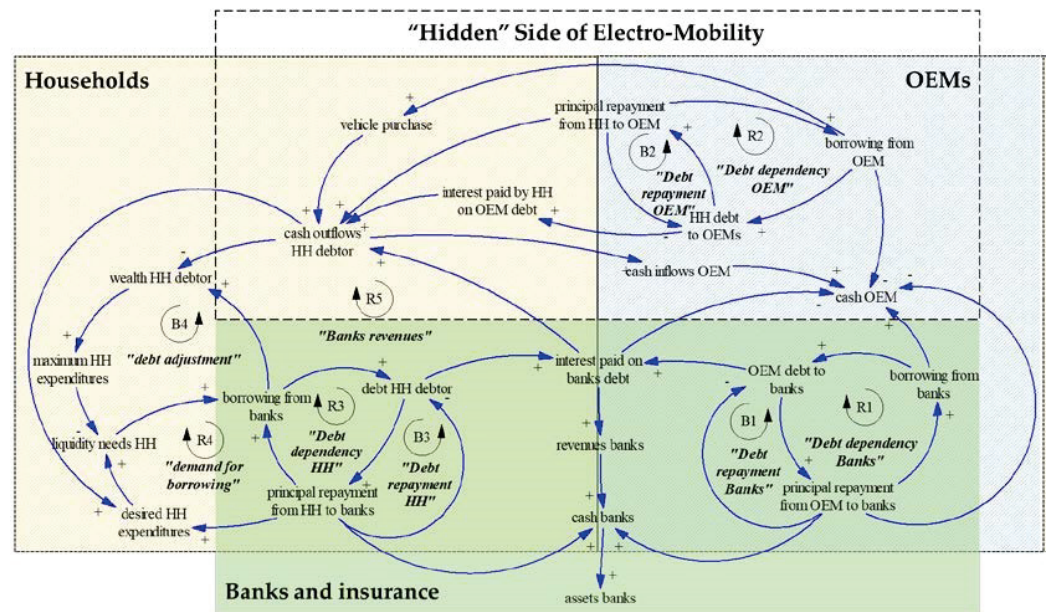


Figure 3. Causal loop diagram with feedback processes involving three agents.

2.2.2. Authorities

Authorities represent the public sector, which, as indicated above, consists of governments and selected EU institutions. Table 2 lists the initial values of these sub-agents.

Table 2. Initial values of the authorities’ balance sheets [EUR].

Sub-Agent	Assets	Liabilities & Equity
ECB	6×10^{12}	6×10^{12}
EIB	2.8×10^{11}	2.8×10^{11}
Government	10^{11}	10^{11}

Source: own assumptions.

The government is defined as a single entity (i.e., without disaggregation by country) in the present version of the model to facilitate the analysis. The liabilities side of the balance sheet consists of bank debt (in this hypothetical case, to the central bank). The simulated government budget is affected by three revenue items (corporate tax, energy tax, value-added tax (VAT)) and four expenditures:

- Public infrastructure expenditures, calculated as follows: the historical and projected electric vehicle supply equipment (EVSE) values for the world, disaggregated into slow- and fast-charging points, provided by [31], are incorporated into the model. The project values reflect the STEPS scenario and are interpolated linearly as needed. The resulting estimates are used to compute annual deployment under the assumption that EVSE is long-lived. Next, the cost of the EVSE is as follows: EUR 9000/point for slow and EUR 100,000/point for fast. We work under the premise that the government provides the required funding for the deployment of such publicly accessible infrastructure.
- Public R&D expenditures: this variable is included and linked to OEMs to facilitate the analysis of government grants to the automotive industry. However, in the current version of the model, it takes a value of zero.
- Public transport procurement: all buses are purchased by this agent.
- Purchase subsidies: estimates of government purchase subsidies for the period 2016–2021 were collected from [32], with a linear decline to zero in 2025 assumed.

Concerning revenues, it is assumed that taxes represent half of the fuel price. The simulated VAT rate is 20%.

EU institutions are modelled in a very basic fashion (see Figure 4 for the example of the ECB). As can be seen in this figure, ECB assets consist of loans (to commercial banks and governments) and cash. The grey variables within the <> signs are known in the SD literature as ‘shadow variables’ and represent variables that are defined in another part of the model. For instance, ‘debt to central bank’ is defined as a stock variable in the ‘Banks and insurance’ agent.

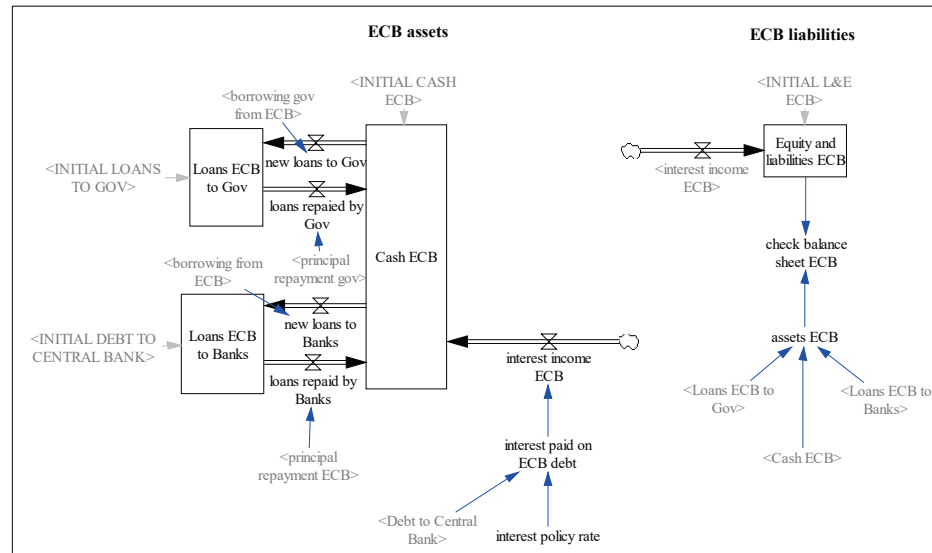


Figure 4. Overview of the ECB structure in the model. Note: the rectangles represent stock variables; flow variables are represented by valves and pipes.

The interest policy rate is fixed at 3%. Concerning the EIB, a favourable rate of 4% is available to OEMs. To make the model operational, it is assumed that the potential revenues from missing vehicle emissions targets are available to the Union’s budget via the EIB.

2.2.3. Banks and Insurance Firms

This agent represents a conglomerate of commercial/private banks that engage also in insurance services. The insurance premium is assumed to be EUR 500 vehicle/year. The annual percentage rate (APR) is simulated with a constant value of 5%. Table 3 shows the initial values of the balance sheet items modelled for this agent.

Table 3. Initial values of the Banks and insurance’ balance sheets [EUR].

Balance Sheet Item	From OEM (Automotive)	From OEM (Financial)
Cash and cash equivalents		10^{11}
Loans to consumers		7.87×10^{12}
OEM debt (short-term)	1.25×10^{11}	7.21×10^{11}
OEM debt (long-term)	1.73×10^{11}	1.02×10^{12}
Deposits		4.5×10^{12}
Debt to central bank		4.5×10^{12}
Equity		10^{12}
Assets		10^{13}
Liabilities and equity		10^{13}

Source: own assumptions.

2.2.4. Households

HHs are divided into two sub-agents: creditor and debtor. The key difference is that the latter holds debt (to banks and OEMs). Another difference is that, while the debtor HH receives the labour wages from vehicle manufacturing, the dividends are accrued to the creditor HH. The wealth of this sub-agent is stored as deposits in the private banking system. As for the other agents, Table 4 shows the key initial values. Most of the cars (with the exception of those registered by OEMs for leasing/rental purposes) are purchased by these two sub-agents, with an equal split in share. The historical and projected global battery electric vehicle (BEV) stock, disaggregated into vehicle type, provided by [31], was fed into the model.

Table 4. Initial values of the HHs' balance sheets [EUR].

Balance Sheet Item	Creditor	Debtor
Wealth (assets)	4.5×10^{12}	9.04×10^{12}
Bank loans	0	7.87×10^{12}
Debt to OEM (short-term)	0	5.19×10^{11}
Debt to OEM (long-term)	0	6.51×10^{11}
Equity	4.5×10^{12}	0
Assets		1.35×10^{13}
Liabilities and equity		1.35×10^{13}

Source: own assumptions.

2.2.5. Suppliers

This agent represents a conglomerate of raw material, battery, energy, infrastructure and vehicle maintenance providers. The corresponding items related to these become sources of annual revenue for this agent. Its assets are initially valued at EUR 1e+12. Suppliers deploy the publicly accessible EVSE commissioned by governments, with depreciation influenced by an average lifetime of 20 years for this asset. Moreover, the Suppliers are the agent purchasing freight vehicles (vans and trucks).

Three types of fuels are supplied by this agent: petrol and diesel for ICEV (the former for cars, and the latter for the rest) and electricity for BEVs.

The markup over cost set by the Suppliers is 20% (a reasonable assumption; see, e.g., [33]), both for raw materials and batteries. The maintenance cost is EUR 300 vehicle/year (for a range by manufacturer see [34]). See the time-variant values used for the exchange rate and the battery and fuel prices in the Supplementary Materials.

2.2.6. Vehicle Manufacturers

By far, the most elaborate agent in our model is the OEM. This agent includes a hypothetical domestic and foreign OEM, each one holding a market share of 50%. Table 5 shows the initial values assigned to each of the two OEM conglomerates, comparing total assets with the sum of liabilities and equity (L&E). Each of them is duplicated, for OEMs are subscribed in our model into Automotive and a Financial Divisions, in line with the information gathered from their annual reports. As can be seen, simplifications of several items were made, and their weight greatly differs by division. For the rest of the agents, these statements are more simple. As expected, OEM debt becomes an asset for commercial banks (recall Table 3). Conversely, HH debt tied to vehicle purchases is captured in financial services receivables (current and noncurrent), which are an asset to the OEM. The evidence on car finance is provided by, for example, [35].

Table 5. Initial values of the OEMs' balance sheet [EUR].

Balance Sheet Item	Automotive Division	Financial Division
Cash and cash equivalents	9.73×10^{10}	1.10×10^{10}
Securities	8.43×10^{10}	5.52×10^9
Trade receivables	5.51×10^{10}	0
Financial services receivables	0	2.60×10^{11}
Inventories	8.75×10^{10}	0
Property, plant & equipment	2.16×10^{11}	0
Intangibles	2.40×10^{10}	0
Leases	0	1.40×10^{11}
Financial services noncurrent	0	3.26×10^{11}
Trade payables	1.07×10^{11}	0
Debt (short-term)	6.27×10^{10}	3.60×10^{11}
Debt (long-term)	8.96×10^{10}	5.08×10^{11}
Retained earnings		1.55×10^{11}
Reserves		2.31×10^{10}
Assets		1.31×10^{12}
Liabilities and equity		1.31×10^{12}

Source: own assumptions as a result of simplifying and aggregating the information contained in publicly available financial statements, as reported by several OEMs.

As hinted above, the first balance sheet item requiring an explanation is PPE, which is affected by depreciation. According to the Generally Accepted Accounting Principles (GAAP) and International Financial Reporting Standards (IFRS), acceptable depreciation methods include straight-line, accelerated and units-of-production methods [22]. Straight-line depreciation was adopted in the SD accounting model by [36]. While there are differences across individual OEM and years, our analysis of OEM financial information leads us to conclude that the straight-line method was the main one adopted over the past decade in this industry. We thus implement this method in our model, following [36]. Assuming that annual vehicle sales remain constant (see Section 2.2.4), plant acquisition is kept constant at EUR 10 billion annually. Investment in vehicles to generate lease revenues amount to EUR 55 billion per year. Annual depreciation and amortisation expenses equal EUR 64.5 billion.

The second item is inventories, whose valuation differs by method. While GAAP accepts the weighted-average cost, first-in, first-out (FIFO) and last-in, first-out (LIFO) methods, IFRS prohibits the use of LIFO [22]. Implementing the average cost method in his model, [36] regards it as the only reasonable one to account for inventories in an SD model, acknowledging that this method is not used by most firms. In our model, we compute physical inventories and their values for four types of vehicles: cars, vans, trucks and buses. These are kept constant over the simulation period.

Table 6 shows additional assumptions made for OEMs. Furthermore, it is assumed that OEMs self-register cars for leasing/rental purposes, with the fleet owned by the Financial Division increasing from almost 14 million cars to almost 27.5 million in 2030. An annual transfer amounting to EUR 30 billion from the Automotive to the Financial Division is simulated.

Focusing on the different types of vehicles modelled, Table 7 lists key assumptions. Each vehicle type is also disaggregated into an internal combustion engine vehicle (ICEV) and a BEV. While the labour costs are assumed to be the same for the two technologies, besides the battery the material costs differ. The table also shows two assumptions that are important for the calculation of the fleet's energy demand, which is needed to compute expenditures on energy products.

Table 6. Further model assumptions for OEMs.

Constant/Variable	Value	Justification
Administrative intensity	4%	Own analysis ¹
Corporate tax rate	30%	[5] (p. 128)
Dividend distribution ratio	40%	[4] (p. 133)
Emission penalties	EUR 0	Assuming targets are met
Lifetime plant	50 yr	Values range up to 60 yr
Lifetime vehicle	5 yr	Values range up to 10 yr
Marketing intensity	10%	Own analysis ¹
R&D intensity	6%	Own analysis ¹
Revenues from leasing	EUR 4900/car	Own assumption
Spread over APR	1%	Own assumption

¹ Forthcoming. ‘Intensity’ is relative to sales revenues. Lifetime values sourced from OEM reports.

Table 7. Assumptions, by type of vehicle.

Constant/Variable	Unit	Car	Van	Truck	Bus
Battery capacity ¹	kWh	24/30/50	24/30/50	300	250
Labour cost	EUR/vehicle	5000	3750	25,000	50,000
Material cost [BEV]	EUR/vehicle	7845	4095	32,069	114,224
Material cost [ICEV]	EUR/vehicle	5000	3750	25,000	50,000
Annual mileage	km/vehicle	12,000	24,000	80,000	110,000
Fuel efficiency [BEV]	kWh/km	0.2	0.2	1.3	1.3
Fuel efficiency [ICEV]	litre/km	0.08	0.08	0.36	0.36

¹ The first value is used for 2005–2014, the second for 2015–2019 and the third for 2020–2030.

3. Results

3.1. Evolution of the Vehicle Fleet

A selection of the model results for the various agents is presented below. All the figures refer to a simulation run named ‘Current’. Figure 5 shows the evolution of the vehicle stock, disaggregated by the type of vehicle and technology. The BEVs exhibit growth at the expense of the ICEVs, though the latter still dominates at the global level in 2030. This is the type of output most models consider, but what are the potential economic and financial implications for the agents involved in this system? The “hidden” side of electro-mobility becomes visible, based on our simulation framework, in the next figures.

3.2. Balance Sheets

As vehicle sales require vehicle production and demand, and both are partly financed with debt, it becomes useful to consider the structure of the liabilities and how it is connected to other agents’ assets. For instance, HH debt to OEMs for the purpose of purchasing vehicles constitutes an asset for the latter, and the debt of the two agents to private banks represents a fraction of the total assets owned by these. EIB loans to OEMs to facilitate cleaner vehicle production also feature in the assets side of the EIB balance sheet. The Central Bank’s assets may also be formed of loans to governments and private banks. This means that the overall behaviour of the system is the result of the interaction between the macro agents (recall Figure 2).

Figure 6 shows the simulated evolution of assets (and thus L&E) for two agents. While the chart on the left shows the output of the two public banks, the behaviour of private banks and the government can be seen in the chart on the right. The balance sheet expands in all cases, though very slowly in the case of the EIB.

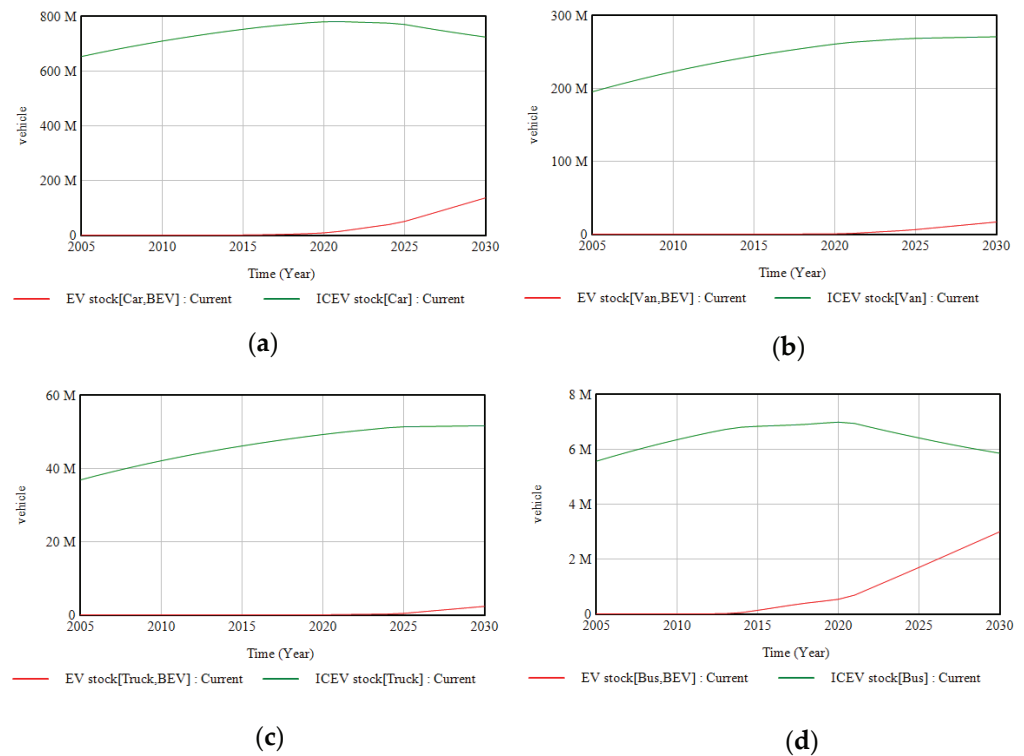


Figure 5. Simulated vehicle stock (2005–2030), by type of vehicle and powertrain (a) cars, (b) vans, (c) trucks, (d) buses.

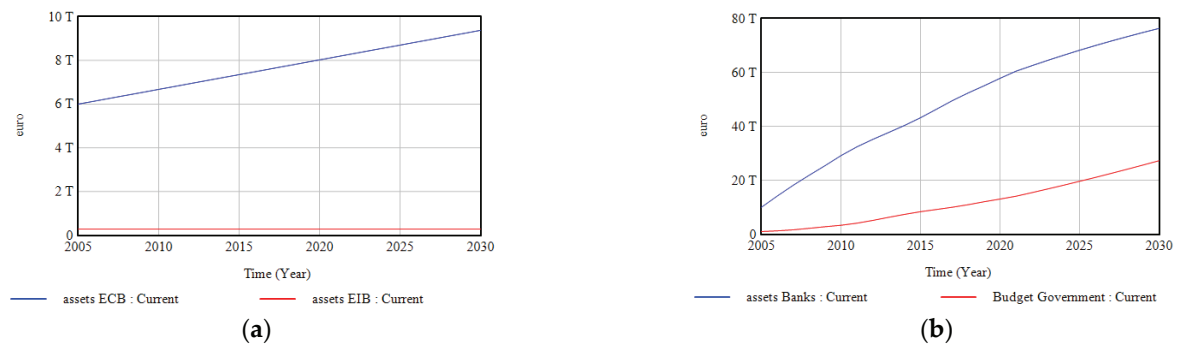


Figure 6. Balance sheet simulation of: (a) public banks; (b) private banks and government.

Similarly, Figure 7 shows the simulated evolution of the balance sheet for the rest of the private sector. As can be seen on the left chart, the wealth of the creditor HH quickly overtakes that of the debtor, which declines slowly towards the end of the simulation. The chart on the right shows the growing assets of the suppliers and vehicle manufacturers, which diverge from a similar base in 2005 until the assets of each OEM doubles in size those of the suppliers in 2030.

What drives the changes in balance sheets are flow variables. For instance, the annual interest paid by HH debtors to OEMs amounts to slightly more than EUR 35 billion (or EUR 1083 per vehicle sold). This is an example of additional results generated by the model which, for reasons of space, cannot be reported here (see the Supplementary Materials instead). Variable OEM profits (growing, in particular, in the Financial Division) not only accumulate into retained earnings but also boosts dividends, which increase the wealth of creditor HHs. The simulated behaviour of ‘net savings HH creditor’ is always positive but exhibits a downwards trend, with swings between 2008 and 2020. This variable affects the stock and flow structure related to bank deposits, which leads to bank deposits growing until 2030 at a slower rate.

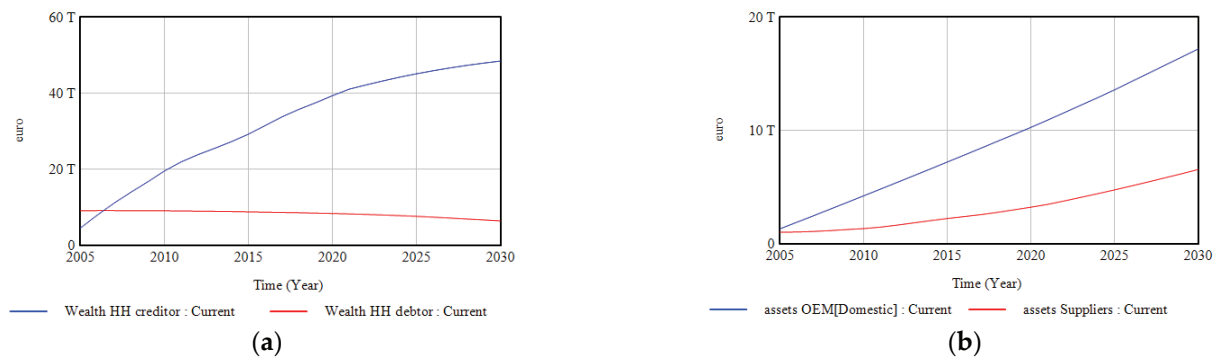


Figure 7. Balance sheet simulation of: (a) households; (b) firms.

Moreover, the flow variables tend to be influenced by business operations and items from the income statement. Besides the physical ones (e.g., sales in Section 3.1), money items such as ‘dividends paid’ play a role. This flow is determined by the dividend distribution ratio (Table 6) and the net income, which is, in turn, used to compute a key performance indicator (KPI), examined next.

3.3. Key Performance Indicators

The computation of KPIs facilitates so-called ratio analysis. Our analysis of OEM reports confirmed our expectation that KPIs are important to OEM decision-makers. While the purpose was not to create a model to improve the behaviour of the automotive industry from an OEM perspective, it is useful to report the four standard profitability KPIs and an operating return KPI: the return on equity (ROE). This is done in Table 8, which also lists, for comparability, typical values found in the literature. Each KPI was computed following [22]. The ROE is reported for the Financial Division, as is common in the sector, while the profitability ratios correspond to the Automotive Division. In our simulation, the ROE peaks with a value of almost 16% in 2007, the year in which the global financial crisis began. As other sources of income were not assumed, our earnings before interest and taxes (EBIT) are the same as the operating income, and thus the two ratios show the same values.

Table 8. Simulated values of the financial KPIs and benchmarks.

KPI	Simulated ¹	25%	Median	75%
Gross margin	42/37%	28%	43%	63%
Operating margin	20/16%	6%	12%	22%
EBIT margin	20/16%	5%	11%	18%
Net profit margin	13/11%	2%	7%	15%
ROE	7/4%	3%	10%	18%

¹ These are our model results, with the first value corresponding to 2005 and the second to 2030. The values for the last three columns correspond to quartiles from large US firms in 2018 and were sourced from Table 2.4 in [24].

By adding KPIs, the decisions of OEMs may be more sensibly modelled. In a way, this helps integrate the physical and financial sides of the OEM business. Our simulated KPI values are within the ranges derived from the data. The fact that the ones for profitability are closer to the 75% quartile while, in reality, some OEMs struggle to exhibit strong KPI performance is likely to be due to our assumption that the global vehicle market is served by a duopoly. In practice, OEMs set target margins that may be attained or not. Moreover, those targets may vary by year, depending on the management’s perception of business and international conditions. For instance, [37] reports how distant their adjusted target range for the automotive EBIT margin was from their target range.

4. Discussion

The automotive industry is undergoing radical transformation. The challenges faced by the sector in response to the presence of climate urgencies were clearly articulated by [38]. The importance of analysing the balance sheet to gauge the success of the business was highlighted by [39]. There seems to be an emerging need, relatively neglected in the existing literature, for researchers and policy-makers to put the potential emission penalties into the broader context of OEMs' financial position and to understand the channels through which money flows (e.g., to promote R&D in cleaner vehicles, to finance zero-emission powertrain sales) among market agents. The present paper represents one step forward towards addressing this need.

The paper describes a simulation model to facilitate such analyses. We came across the work by [36] after we built our accounting framework but incorporated some of the features suggested by the author (recall Section 2.2.6).

Model testing was proposed as a means to demonstrate the standard behaviour of the model without aiming to explore either large sensitivities and uncertainties in every single parameter, or behaviour reproducibility tests based on historical data. This is in line with the purpose of this paper, which was to highlight the existence of the "hidden" side of the automotive sector, with OEMs ultimately behaving as financial institutions providing loans and receiving interest payments from their clients. Standard tests in line with the SD literature were performed, such as integration error tests, and unit consistency as indicated in Appendix A.

One way of exploring uncertainty is to compute alternative vehicle sales growth rates. This would then alter OEM manufacturing capacity needs and, in turn, investment in PPE. According to [40], retained earnings finance most EU firm investments. This would have implications for Figure 7b, which would also change if greater bargaining power were accorded to Suppliers. By making the model available, we facilitate this type of analysis being carried out by the interested reader.

The model assures full consistency in keeping the economic and financial flows within the system. By interlinking the money flows among the agents, it is possible to trace how money circulates within the automotive sectors. In particular, this allows analysts to keep track of cumulative public and private expenditures, such as purchase subsidies and R&D expenditures, respectively.

The simulated vehicle fleet suggests that petrol and diesel-powered vehicle remain the dominant technology until 2030, which would have environmentally damaging consequences unless other actions are taken. This picture could be altered if government grants and subsidies in support of innovations that lower emissions were simulated (recall Section 2.2.2). The important role public banks have to play in the transition to net zero has most recently been highlighted by [41].

Turning to our balance sheet results, by distinguishing between debtor and creditor HHs, we are in a position to start investigating distributional issues (e.g., automotive dividends and wages). However, a key limitation of our work is that it depicts HHs and governments as aggregated entities. A more realistic representation would disaggregate them by country. We felt that such level of detail would obscure the present exercise and be, thus, counterproductive.

While debt was modelled and securities were included explicitly in the OEMs' assets, the capital and financial markets were portrayed in a simple manner. No explicit reference was made to, for example, money markets, commercial papers or corporate bonds (see, e.g., [42]) or risk management (see OEMs' reports). No leverage, liquidity and valuation ratios were calculated, as outside of the main purpose of our model, but these could be easily added in the system, based on the version proposed.

The focus was on interrelated patterns rather than realistic numerical outcomes. While these may not correspond to actual data, setting the initial asset values to EUR 1 trillion for governments and suppliers (close to the estimated value for OEMs) was not completely undeliberate, for it facilitated the comparison of behaviours unfolding from a similar base.

The sum of all the assets in our model amount to EUR 150.6 trillion in 2020. To put this into context, [43] estimated that the world economy had USD 1540 trillion in assets in the same year, of which one-third corresponded to real assets. Further data refinements may be made in the future: for the ECB, using [44], for the EIB, relying on [45], and for governments, using the Public Sector Balance Sheet (PSBS) The database is available at [46].

We conclude that the incorporation of the financial statements of key electro-mobility agents and their interlinkages in a simulation model is both feasible and a desired property for more realistic and policy relevant models. After all, the process of electrification does not follow solely from policy prescriptions but is the result of the way the key players digest relevant information, including financial. Taking these aspects into account in a modelling framework leads to an explicit generic representation of the banking sector. The downside of this is that the model becomes larger and understanding it more demanding. Still, we conclude that the benefits of this approach outweigh its costs, as it brings a perspective other modelling tools neglect.

Though we modelled the global vehicle market, we opted for defining authorities in European terms. For this reason, references to the ECB, EIB and the EU were made. This was done to emphasise the emission penalties that are relevant in the EU context (as this was one of the points made in the Introduction). However, such agents can be defined in more general terms (e.g., central bank instead of the ECB or public investment bank instead of the EIB), and the structure can, in principle, be applied to other markets. As a matter of fact, no excess emissions leading to penalties were simulated. To compute this in a robust manner, a representation of the existing regulations is needed. This is indeed the key connection point between the model proposed here and PTTMAM, which models HHs ('Users') and 'Authorities' with disaggregation by country. Such model integration will be pursued in future work. While there may be physical constraints that limit the speed of EV uptake (e.g., battery supply bottlenecks), the resulting model upgrade would enable the analysis of potential financial aspects.

The SFC framework proposed here makes the "hidden" side of electro-mobility visible, with the implication that it facilitates the analysis of potential financial constraints that might also jeopardise faster road travel electrification. Conversely, it may help identify the levers in the system for more effective financial support.

Supplementary Materials: Supporting information can be downloaded at <http://data.europa.eu/89h/2086b2cb-3f20-4241-b8f8-00fa99969f86>.

Author Contributions: Conceptualization, J.J.G.V.; methodology, J.J.G.V. and R.P.; software, J.J.G.V. and R.P.; validation, J.J.G.V. and R.P.; formal analysis, J.J.G.V. and R.P.; data curation, J.J.G.V.; writing—original draft preparation, J.J.G.V.; writing—review and editing, R.P.; visualization, J.J.G.V. and R.P.; project administration, J.J.G.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data is included in the Supplementary Materials.

Acknowledgments: We are very grateful to two anonymous reviewers and an internal reviewer for their comments. We are also grateful to three anonymous reviewers from the Transportation Research Board's Standing Committee on Economics and Finance for their helpful remarks on a preliminary version of this work. We also thank Giorgos Fontaras for his support. The views expressed are purely those of the author and may not, in any circumstances, be regarded as stating an official position of the European Commission.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Documenting the model with the System Dynamics Model Documentation and Assessment Tool (SDM-Doc) proposed by [47] reveals that the model contains 451 variables, of which 55 are stocks. The documentation file, which contains the model's full code, is available in the Supplementary Materials.

Two SFC tests are performed, one for each individual agent (i.e., assets and liabilities match within every agent at every point in time), and a second one at the system level (i.e., financial institutions' liabilities match the liquidity in the entire economy). Concerning the second test, which determines whether consistency at the system level (i.e., for all the agents involved) is attained, the results are shown in Figure A1. The deviations from zero in our checks (visible in Figure A1 and available also in the model) are rather small, that is, within EUR 0.1, out of the large numbers used to initialize the stock and cash flow variables to several billion Euros, as indicated in Tables 2 and 3). This also partially works as an integration test in the model. This SD model is a continuous time model adopting Euler type integration at time steps of 0.25 years. Due to the stock calculation inherent in the software, the system makes tiny approximations of the large numbers that have been used to run the model. The tiny unbalances, shown in Figure A1, are simply due to this, and we can consider these as 0, thus satisfying the SFC condition.

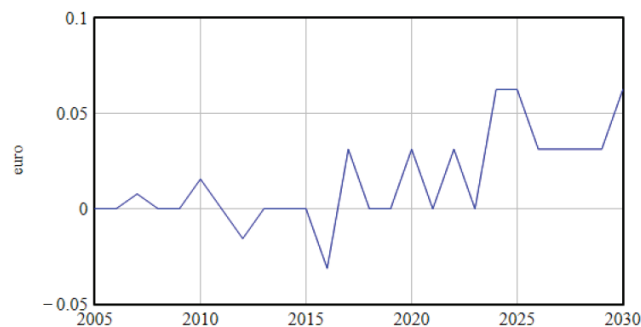


Figure A1. Outcome of the SFC check.

The integration error test was also performed in line with the standard SD literature, thus demonstrating that behaviour does not change below a certain delta time, and minimising the computation power to run the simulation to the minimum possible [28]. Systems boundary adequacy and structure assessment tests were carried out during the phase of construction of the model on an iterative basis, and the test for unit consistency was passed with the completed version of the model.

References

- Harrison, G.; Thiel, C.; Jones, L. *Powertrain Technology Transition Market Agent Model (PTTMAM): An Introduction*; Publications Office of the European Union: Luxembourg, 2016.
- European Union. Regulation (EU) 2019/631 of the European Parliament and of the Council of 17 April 2019 Setting CO2 Emission Performance Standards for New Passenger Cars and for New Light Commercial Vehicles. 2019. Available online: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32019R0631> (accessed on 1 December 2022).
- Joint Research Centre. Powertrain Technology Transition Market Agent Model (PTTMAM). 2022. Available online: https://joint-research-centre.ec.europa.eu/scientific-tools-and-databases/powertrain-technology-transition-market-agent-model-pttmam/powertrain-technology-transition-market-agent-model-pttmam_en (accessed on 1 December 2022).
- Daimler. *Annual Report 2020*; Daimler AG: Stuttgart, Germany, 2021.
- Volkswagen. *Annual Report 2020*; Volkswagen (VW) AG: Wolfsburg, Germany, 2021.
- European Investment Bank. European Investment Bank (EIB). 2021. Available online: www.eib.org (accessed on 1 December 2022).
- Lam, A.; Mercure, J.-F. Which policy mixes are best for decarbonising passenger cars? Simulating interactions among taxes, subsidies and regulations for the United Kingdom, the United States, Japan, China, and India. *Energy Res. Soc. Sci.* **2021**, *75*, 101951. [CrossRef]
- Mercure, J.-F.; Salas, P.; Vercoulen, P.; Semieniuk, G.; Lam, A.; Pollitt, H.; Holden, P.B.; Vaklifard, N.; Chewpreecha, U.; Edwards, N.R.; et al. Reframing incentives for climate policy action. *Nat. Energy* **2021**, *6*, 1133–1143. [CrossRef]

9. Jochem, P.; Gómez Vilchez, J.J.; Ensslen, A.; Schäuble, J.; Fichtner, W. Methods for forecasting the market penetration of electric drivetrains in the passenger car market. *Transp. Rev.* **2018**, *38*, 322–348. [CrossRef]
10. Gómez Vilchez, J.J.; Jochem, P. Simulating vehicle fleet composition: A review of system dynamics models. *Renew. Sustain. Energy Rev.* **2019**, *115*, 109367. [CrossRef]
11. Gómez Vilchez, J.J.; Jochem, P.; Fichtner, W. Interlinking major markets to explore electric car uptake. *Energy Policy* **2020**, *144*, 111588. [CrossRef]
12. Gómez Vilchez, J.J.; Thiel, C. Simulating the battery price and the car-mix in key electro-mobility markets via model coupling. *J. Simul.* **2020**, *14*, 242–259. [CrossRef]
13. Cavana, R.Y.; Dangerfield, B.C.; Pavlov, O.V.; Radzicki, M.J.; Wheat, I.D. *Feedback Economics: Economic Modeling with System Dynamics*; Springer International Publishing: Berlin/Heidelberg, Germany, 2021.
14. Godley, W.; Lavoie, M. *Monetary Economics: An Integrated Approach to Credit, Money, Income, Production and Wealth*; Palgrave Macmillan: London, UK, 2007.
15. Pasqualino, R.; Jones, A.W. *Resources, Financial Risk and the Dynamics of Growth: Systems and Global Society*; Taylor & Francis: Abingdon, UK, 2020.
16. Battiston, S.; Monasterolo, I.; Riahi, K.; van Ruijven, B.J. Accounting for finance is key for climate mitigation pathways. *Science* **2021**, *372*, 918–920. [CrossRef] [PubMed]
17. Dunz, N.; Naqvi, A.; Monasterolo, I. Climate sentiments, transition risk, and financial stability in a stock-flow consistent model. *J. Financ. Stab.* **2021**, *54*, 100872. [CrossRef]
18. Xing, X.; Pan, H.; Deng, J. Carbon tax in a stock-flow consistent model: The role of commercial banks in financing low-carbon transition. *Financ. Res. Lett.* **2022**, *50*, 103186. [CrossRef]
19. Sanders, M.; Serebriakova, A.; Fragkos, P.; Polzin, F.; Egli, F.; Steffen, B. Representation of financial markets in macro-economic transition models—A review and suggestions for extensions. *Environ. Res. Lett.* **2022**, *17*, 083001. [CrossRef]
20. Sanz, R.A.; Arias, J.G.; Fidalgo, J.M.G.; García, R.M. *Finanzas Empresariales*; Centro de Estudios Ramón Areces: Madrid, Spain, 2016.
21. Koo, R.C. *The Holy Grail of Macroeconomics: Lessons from Japan's Great Recession*; Wiley: Hoboken, NJ, USA, 2011.
22. KPMG. *IFRS® Compared to US GAAP*; KPMG International Limited: Amstelveen, The Netherlands, 2020.
23. Ittelson, T. *Financial Statements: A Step-by-Step Guide to Understanding and Creating Financial Reports*; CAREER Press: Wayne, NJ, USA, 2020.
24. Berk, J.; DeMarzo, P. *Corporate Finance*; Pearson: London, UK, 2019.
25. Yamaguchi, K. *Money and Macroeconomic Dynamics—Accounting System Dynamics Approach*; Japan Futures Research Center: Awaji Island, Japan, 2020; Available online: <http://www.muratopia.org/Yamaguchi/MacroBook.html> (accessed on 10 February 2023).
26. Bossel, H. *Systems and Models: Complexity, Dynamics, Evolution, Sustainability*; BoD—Books on Demand: Norderstedt, Germany, 2007.
27. Sterman, J.D. *Business Dynamics: Systems Thinking and Modeling for a Complex World*; McGraw-Hill/Irwin: Boston, MA, USA, 2000.
28. López, B.B.; Julve, V.M. *Fundamentos de Contabilidad Financiera*; Ediciones Pirámide: Madrid, Spain, 2021.
29. Requeijo, J.; González, J.R. *Técnicas Básicas de Estructura Económica*; Delta Publicaciones: Kiel, WI, USA, 2007.
30. Wheat, I.D. The feedback method of teaching macroeconomics: Is it effective? *Syst. Dyn. Rev.* **2007**, *23*, 391–413. [CrossRef]
31. International Energy Agency. *Global EV Data Explorer*; International Energy Agency (IEA): Paris, France, 2021; Available online: <https://www.iea.org/articles/global-ev-data-explorer> (accessed on 1 December 2022).
32. International Energy Agency. *Global EV Outlook 2022*; International Energy Agency (IEA): Paris, France, 2022.
33. Blatcher, R. Distributor Markup and Profit Margins in the Supply Chain. PROS. 2018. Available online: <https://pros.com/learn/blog/distributor-pricing-profit-margins-pricing-markups> (accessed on 1 December 2022).
34. Organización de Consumidores y Usuarios. *Los coches más fiables*; Organización de Consumidores y Usuarios (OCU), 2022. Available online: <https://www.ocu.org/coches/coches/noticias/coches-mas-fiables> (accessed on 1 December 2022).
35. Eurofinas. *Annual Survey 2021*; Eurofinas: Brussels, Belgium, 2022.
36. Pierson, K. Operationalizing Accounting Reporting in System Dynamics Models. *Systems* **2020**, *8*, 9. [CrossRef]
37. Wells, P.E. *The Automotive Industry in an Era of Eco-Austerity: Creating an Industry as If the Planet Mattered*; Edward Elgar: Cheltenham, UK, 2010.
38. Graham, B.; Dodd, D.L.F. *Security Analysis: The Classic 1951 Edition*; McGraw-Hill Education: New York, NY, USA, 2005.
39. de Haan, J.; Oosterloo, S.; Schoemaker, D. *Financial Markets and Institutions*; Cambridge University Press: Cambridge, UK, 2015.
40. European Investment Bank. *Energy Crisis Makes Public Banks Even More Important*; European Investment Bank (EIB), 2022. Available online: https://www.eib.org/en/stories/energy-crisis-net-zero-transition?utm_source=mailjet&utm_medium=Email&utm_campaign=november&utm_content=na (accessed on 1 December 2022).
41. Bayerische Motoren Werke. *Annual Report 2019*; Bayerische Motoren Werke (BMW) AG: Munich, Germany, 2020.
42. Stigum, M.; Crescenzi, A. *Stigum's Money Market, 4E*; McGraw Hill LLC: New York, NY, USA, 2007.
43. McKinsey. In *The Rise and Rise of the Global Balance Sheet—How Productively Are We Using Our Wealth?* McKinsey Global Institute: Washington, DC, USA, 2021.
44. European Central Bank. *Annual Consolidated Balance Sheet of the Eurosystem*; European Central Bank (ECB), 2022. Available online: <https://www.ecb.europa.eu/pub/annual/balance/html/index.en.html> (accessed on 1 December 2022).

45. European Investment Bank. Shareholders. *Shareholders*; European Investment Bank (EIB), 2022. Available online: <https://www.eib.org/en/about/governance-and-structure/shareholders/index.htm> (accessed on 1 December 2022).
46. International Monetary Fund. *Public Sector Balance Sheet (PSBS) Database*; International Monetary Fund (IMF), 2022. Available online: <https://data.imf.org/?sk=82A91796-0326-4629-9E1D-C7F8422B8BE6> (accessed on 1 December 2022).
47. Martinez-Moyano, I.J. Documentation for model transparency. *Syst. Dyn. Rev.* **2012**, *28*, 199–208. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Enhancing Local Decisions in Agent-Based Cartesian Genetic Programming by CMA-ES [†]

Jörg Bremer ^{1,*} and Sebastian Lehnhoff ^{2,‡}¹ Department of Computing Science, University of Oldenburg, 26129 Oldenburg, Germany² Energy Informatics Group, University of Oldenburg, 26129 Oldenburg, Germany

* Correspondence: joerg.bremer@uni-oldenburg.de; Tel.: +49-441-9722-736

[†] This paper is an extended version of our paper published in *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation. The PAAMS Collection 20th International Conference, PAAMS 2022, L'Aquila, Italy, 13–15 July 2022, Proceedings.*[‡] These authors contributed equally to this work.

Abstract: Cartesian genetic programming is a popular version of classical genetic programming, and it has now demonstrated a very good performance in solving various use cases. Originally, programs evolved by using a centralized optimization approach. Recently, an algorithmic level decomposition of program evolution has been introduced that can be solved by a multi-agent system in a fully distributed manner. A heuristic for distributed combinatorial problem-solving was adapted to evolve these programs. The applicability of the approach and the effectiveness of the used multi-agent protocol as well as of the evolved genetic programs for the case of full enumeration in local agent decisions has already been successfully demonstrated. Symbolic regression, n-parity, and classification problems were used for this purpose. As is typical of decentralized systems, agents have to solve local sub-problems for decision-making and for determining the best local contribution to solving program evolution. So far, only a full enumeration of the solution candidates has been used, which is not sufficient for larger problem sizes. We extend this approach by using CMA-ES as an algorithm for local decisions. The superior performance of CMA-ES is demonstrated using Koza's computational effort statistic when compared with the original approach. In addition, the distributed modality of the local optimization is scrutinized by a fitness landscape analysis.

Keywords: Cartesian genetic programming; multi-agent system; COHDA; distributed optimization; CMA-ES

Citation: Bremer, J.; Lehnhoff, S. Enhancing Local Decisions in Agent-Based Cartesian Genetic Programming by CMA-ES. *Systems* **2023**, *11*, 177. <https://doi.org/10.3390/systems11040177>

Academic Editors: Philippe Mathieu, Juan M. Corchado, Alfonso González-Briones and Fernando De la Prieta Pintado

Received: 20 February 2023

Revised: 7 March 2023

Accepted: 14 March 2023

Published: 28 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In [1], a variant of genetic programming (GP) has been proposed that uses a lattice layout of the nodes instead of a tree and is thus called Cartesian genetic programming (CGP). Since then it has become quite popular; it has been broadly adopted [2] and applied to many different use cases and applications [3–5].

Programs in CGP are encoded by an integer-based representation of a directed graph. In this way, the alleles encode the addresses of the other nodes, which serve as data sources for their own functions, or they encode functions by their addresses in a look-up table. The data addresses always refer to the outputs previously calculated by other function nodes that are further ahead in the execution row. Later versions have also experimented with float-based representations [6]. In order to organize the graph in an optimized way (with respect to solving a given problem), so far, only centralized algorithms are used.

On the other hand, systems with self-organizing capabilities, such as multi-agent-based systems, are widely seen as a promising method for coordination and distributed optimization in cyber-physical systems (CPS) with a large number of distributed entities, such as sensing or operational equipment [7,8], and especially for horizontally distributed

control tasks [9]. Solving a genetic programming problem can also be seen as such a system [10].

Although up to now most cyber-physical systems are primarily on a semi-autonomous level, future CPS will show a degree of autonomy that makes them hard to control (cf. [11,12]). The autonomy in CPS emerges, for example, from integrated concepts, such as self-organization principles, that are used for coordination inside the system as well as for reaching autonomy. Autonomy may also be achieved by the integration of artificial intelligence (AI). Looking at the example of the European Union, such AI-enabled algorithms are stipulated in [13]. Contemporary cyber-physical systems already comprise a huge number of physical operation and sensing equipment that has to be orchestrated for secure and reliable operation. The electric energy grid, modern transportation systems, and environmental management systems [14] are prominent examples of such large-scale CPS. Multi-agent systems and self-organization principles have now been used for many different applications. Examples can be found in [15–17].

A growing degree of autonomy is desirable to achieve in future CPS [18,19] because the size of the systems, and thus the complexity, steadily grows. Thus, the sizes of the optimization problems and coordination tasks grow as well. Due to some limitations in predictability, adaptive control is desirable and often translated into self-organization approaches. Thus, self-organization seems most promising for the design of adaptive systems [20].

A targeted design of a specific emergent behavior is difficult to achieve, especially when using just general-purpose programming languages [21]. Consequently, the authors in [22] proposed learning, in self-organized systems, how to solve new problems in a decentralized way. For the purpose of swarm-based optimization, the feasibility has successfully been demonstrated [10]. So far, the evolution of learned problem-solving programs was achieved by using a centralized algorithm. On the other hand, that use case constitutes the first reason to distribute the evolution of CGP programs: to enable a swarm to achieve program evolution by itself in a likewise decentralized manner.

For the evolution of Cartesian genetic programs, a $(1 + \lambda)$ -evolution strategy is often used. As a single operator for variation, the mutation is harnessed. A mutation may operate on all or only a subset of the active genes [23,24]. Although initially thought of to be of low or even no use [1], crossover can help a lot if multiple chromosomes have independent fitness assessments [25]. An in-depth analysis of some reasons can be found in [10], in which the authors scrutinize the fitness landscape as an example of learning some meaningful behavior inside a swarm. This is not the case in our application, so we will also not make any use of operators, such as crossover, and restrict ourselves to the mutation. Recently, some distributed use cases have been contemplated that showed that CGP evolution can be very time-consuming when solved by a centralized algorithm [22].

In the case of standard GP, distributed versions have already been in place for a while [26]. Although distributed GP is closely related to CGP due to their representations, distributed CGP has been so far missing. So, another reason to distribute CGP is the acceleration by parallelization and distribution of the evolution process and thus of the computational burden.

Optimization in multi-agent systems has now been researched in various directions and brought up a multitude of synchronization concepts and [27–29] decentralized optimization protocols [30,31]. A good overview can be found in [32]. In [33], an algorithmic level decomposition [34] of CGP evolution has been proposed. This was achieved by using an agent-based, distributed approach to solving [35]. In [36] the fully decentralized, agent-based approach combinatorial optimization heuristics for distributed agents (COHDA) has been proposed as a solution to problems that can be decomposed on an algorithmic level. The general concept is closely related to cooperative coevolution [37]. The key concept of COHDA is an asynchronous iterative approximate best-response behavior. Each agent is responsible for one dimension of the algorithmic problem decomposition. The intermediate solutions of other agents (represented by published decisions) are regarded as temporarily

fixed. Thus, each agent only searches along a low-dimensional cross-section of the search space and thus has to solve merely a simplified sub-problem. Nevertheless, for evaluation of the solution, the full objective function is used after the aggregation of all agents' contributions. In this way, the approach achieves an asynchronous coordinate descent with the additional ability to escape local minima by parallel searching different regions of the search space and because former decisions can be revised if newer information becomes available. This approach is especially suitable for large-scale problems [38].

To adapt COHDA to CGP, the chromosome that encodes the graph representation is split up into sub-chromosomes for each node ([33]). Assigning the best alleles to a chromosome that encodes a computation node is then regarded as the low-dimensional optimization problem of a single agent. Thus, to each node, exactly one agent is assigned. The multi-agent system jointly evolves the program with agents that can be executed independently and fully distributed.

Evolving the problem requires frequent decisions on the (probably) best assignment of the functions and the respective wiring of the inputs made by individual agents. So far, the overall approach had been tested with agents fully enumerating through the set of all possible solutions. This was feasible as the test problems were small enough, and the aim was to evaluate the overall approach without any randomness inside an agent's decision function. Here, we extend this approach by using a heuristic for deciding the best local solutions. As this heuristic needs to be called upon many times during the agent negotiations, we chose to use the covariant matrix adaption evolution strategy (CMA-ES), which is well known for the property of using just a low budget of objective evaluations. Thus, the contributions of this paper are the adaption of CMA-ES as a solver for the local optimization of an agent's decision routine; an analysis of the individual and time variable complexity of the local optimization problems, and additional results demonstrating the superiority of the CMA-ES approach.

The rest of the paper is organized as follows. We start with a recap of both technologies that are combined into the distributed CGP. We describe the adaption of COHDA to CGP and how CMA-ES is adapted to suit the local optimization problem of an agent's decision. The applicability and the effectiveness of the enumeration approach are demonstrated using problems from symbolic regression, n-parity problems, and classification problems. Finally, we demonstrate the superiority of the CMA-ES approach for larger problems. We justify the choice by analyzing the trace of the modality of the agent's local optimization problems with a fitness landscape analysis. We conclude with a prospective view of further use cases and possible extensions.

2. Distributing Cartesian Genetic Programming

In CGP, computer programs are encoded as graph representations [39]. In general, CGP is an enhancement of a method that was originally developed for the use case of evolving digital circuits [40,41]. CGP already demonstrated its capabilities in synthesizing complex functions. Several different use cases from different fields have so far been scrutinized, for example, for image processing [42] or neural network training [4]. Moreover, some additions to CGP have been developed. Examples comprise recurrent CGP [23] (as in recurrent neural networks) or self-modifying CGP [43]. The authors in [33] used standard CGP. As the extension presented here improves the original approach in an internal sub-process, we also used standard CGP.

A chromosome in CGP comprises function-encoding genes and connection- and output-encoding genes. Together, they encode a computational graph (actually, a grid) that represents the executable program. The example in Figure 1 shows a graph with six computational nodes, two inputs, and also two outputs. The allele that encodes a function represents the index in an associated look-up table (from 0 to 3 in the example) with a list of all functions.

Each computation node is encoded by a gene sequence consisting of the function look-up index and the connected input (input into the system or output of another computation

node). These are the parameters that are fed into the function. Hence, the total length of each function node gene sequence is $n + 1$ with n being the arity of the function plus one allele for the function index. The graph in standard CGP is an acyclic one. Parameters that are fed into a computation node may only be collected from nodes that have already been executed or from the inputs into the system. Thus, standard CGP works with a predefined execution order. Outputs can be connected to any computation node (representing the computation result of the output) or directly to any system input. Not all outputs of computational nodes are used as input for other functions. There might be no connections. Usually, many such unused computational nodes occur in evolved CGP [40] programs. These nodes are inactive, do not contribute to the encoded program’s output, and are not executed during the interpretation of the program. Thus, they do not cause any delay in computation.

Phenotypes are of variable length. In contrast, the size of the chromosome is static. As functions may have different arities, some genes may remain unused as input connections. Using an intermediate output of an inner node is also not mandatory. In this way, the fact that evolution is mainly responsible for rewiring makes CGP special.

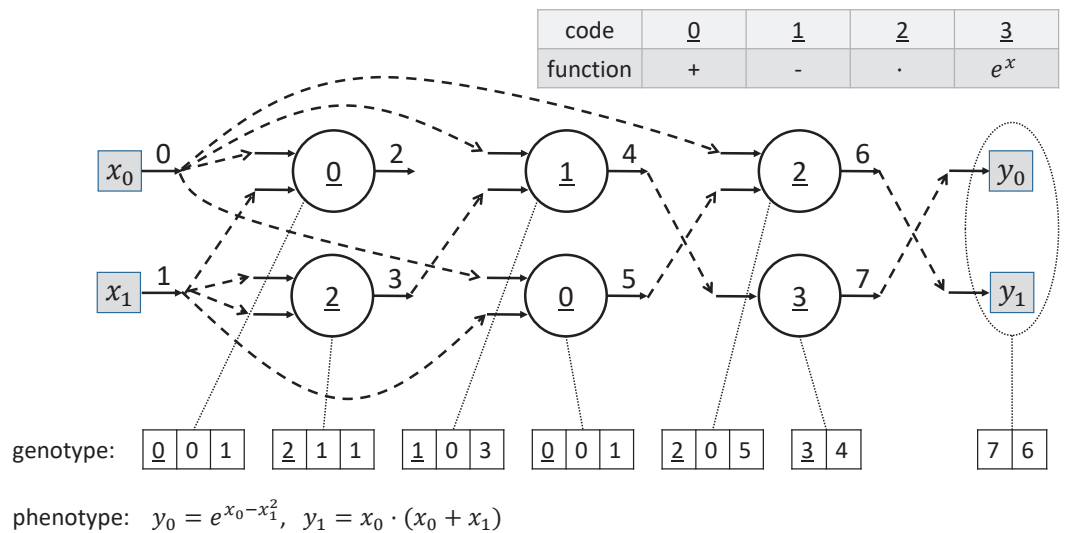


Figure 1. Computational graph and its genotype and phenotype representation in Cartesian genetic programming; modified after [22].

COHDA has been introduced as a distributed multi-agent solution to distributed energy management problems [44]. Since then, it has been used for many different use cases. Examples include coalition structure formation [45], global optimization [38], trajectory optimization for unmanned air vehicles [46], or surplus distribution based on game theory [47].

In [33], COHDA has been used for distributing the evolution of Cartesian genetic programs. When COHDA is executed, each participating agent reacts to updated information from other agents. This is completed by adapting the own previous decision on a possible (local) solution candidate. In the original use case, COHDA agents represented a decentralized energy unit in distributed energy management. In this way, agents had to select an energy generation scheme for their own controlled energy unit. The selection had to be carried out such that it enables a group of energy-generation units to jointly fulfill an energy product (e.g., from some energy market) as well as they can. So, each agent had to decide on the local energy-generation profile only for a single generator. All decisions were always based on (intermediate) selected production schemes of other agents.

We start by we explaining the agent protocol that is the base algorithm for CGP program evolution.

In [36], COHDA was introduced to solve a problem known as predictive scheduling [48]. This is a problem from the smart grid domain and serves as an example here.

COHDA works with an asynchronous iterative approximate best-response behavior, which means that all agents coordinate themselves by updating knowledge and exchanging information about each other. The agents make local decisions solely based on this information. The general protocol works in three repeatedly executed steps.

However, in the beginning, the agents are drawn together first by an artificial communication topology. As a first step, a small-world topology [49] has proven useful and is the most used topology. For this reason, we also use a small-world topology. Starting with an arbitrarily chosen agent and then passing it a message containing just the global objective, each agent repeatedly goes through three stages: perception, decision, and action (cf. [50]).

Perception phase: In this first phase, the agent prepares for local decision-making. Each time the agent receives a message from one of the neighboring agents (which precedes the directed communication topology), the data that are contained in the message are included into their own knowledge base. The data that come with the message consist of the updated local decision of the agent that sent the message and the transient information on the decisions of other agents that led to the previous agent's decision. After updating the local knowledge with the received information, usually, a local decision is made based on this information. In order to better escape local minima, agents may postpone a decision until more information has been collected [51].

Decision phase: Here, the agent has to conduct a local optimization to yield the best decision for its own local action that puts the coalition forward as best as possible. To complete this, each agent solves a low-dimensional part of the problem. The term "dimension", in this context, can also refer to a sub-manifold containing low-dimensional local solutions as a fraction of a much higher-dimensional global search space. In the smart grid use case, for example, a local contribution to the global solution (the energy generation profile for a large group of independently working energy resources) consists of a many-dimensional real-valued vector describing the amount of generated energy per time interval for one single device. In the CGP case, a local solution would consist of a local chromosome encoding the functions and inputs of a single node. Other agents have made local decisions before, and, based on the gathered information about the (intermediate) local decisions of these agents, a solution candidate for the local (constrained) search space is sought. To this decision phase, we added CMA-ES for better local optimization.

Action phase: In the last stage, the agent compares the fitness of the best-found solution with the previous solution. For comparison, the global objective function is used. When the new solution has a better fitness (or a lower error, depending on the specific problem setting), the agent finally broadcasts a message containing its new local solution contribution together with everything it has learned from the other agents and their current local solution contributions (the decision base) to its immediate neighbors in the communication topology. Receiving agents then execute these three phases from scratch, leading probably to revised local solution contributions and thus to further improved overall solutions.

If an agent is not able to find a local solution contribution that improves the overall solution, no message is sent. If no agent can find any improvement, the process has reached at least the local optimum and eventually ceases because no more messages are sent.

After the system has produced a series of intermediate solutions, the heuristic eventually terminates in a state where all agents know an identical solution. This one is taken as the final solution of the heuristic. Properties such as guaranteed convergence and local optimality have been formally proven [44]. Moreover, after a short setup time, COHDA possesses the anytime property. Thus, the agent protocol may be stopped at any time with a valid (sub-optimal) solution, if necessary.

The agent approach can be adapted to CGP as follows. Each agent is responsible for a single node in the program graph. In general, there are two types of agents, the function node agents a_{f_i} responsible for the function node f_i and the output agents a_{y_j} responsible

for the output y_j . The task of both agent types is rather similar but can be distinguished by the local search space. Each function node agent is responsible for exactly one node and thus internally just manages the code (look-up table address) of the function and the respective input addresses as a variable number of integers depending on the arity of the function. This list of integers is just one sub-chromosome of a complete solution. At the same time, every agent has some knowledge about the intermediate assignment of alleles to chromosomes of other agents. These are immutable for this agent. Together (their own gene set and the knowledge of others' gene sets), the genotype of a complete solution, and, subsequently, a phenotype solution can be constructed by an agent.

If an agent that is responsible for a function node receives a message, it updates its own knowledge about the other agents. Each agent knows the most recent chromosomes from other agents. If newer information is received with the message, the outdated gene information is updated. In the case that such far unknown information arrives, additional genes are integrated into the agent's own belief. After the data update, the agent has to make a decision about its own chromosome. This decision is a local optimization process previously solved by enumerating all of the solution candidates [33].

The known genes of the other agents are temporarily treated as fixed for the duration of the agent's current decision-making. Each agent may make modifications only to its own chromosome. Nevertheless, the genes of the other agents may afterward again be altered by the respective agents as a reaction to previously made alterations. If an agent makes a new local decision, it solves for the global problem of finding a good genotype but may only mutate its own local chromosome. Figure 2 shows this situation as an example of the agent a_{f_3} that is responsible for the function node f_3 .

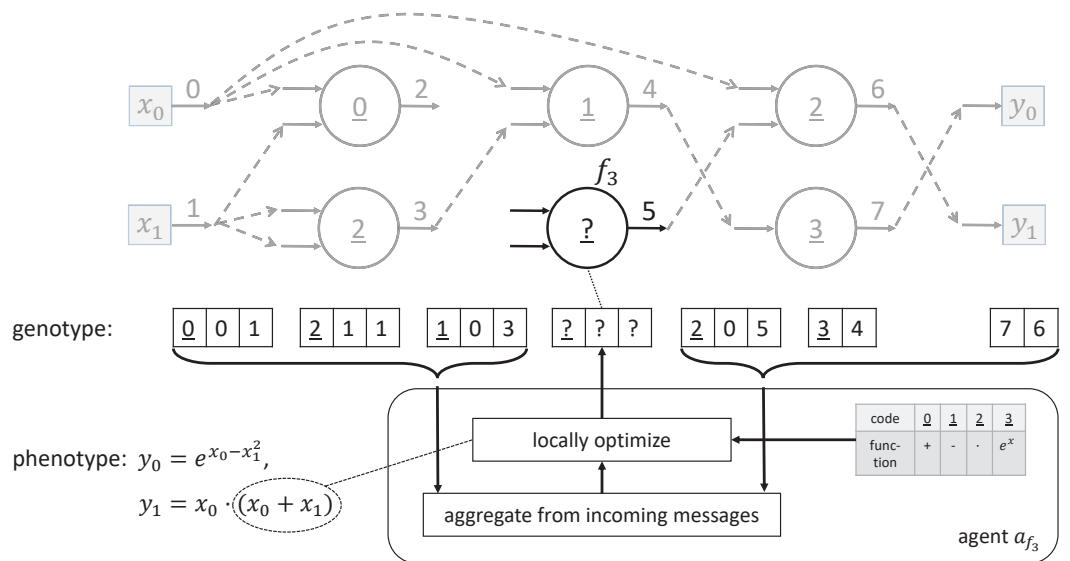


Figure 2. Single, local optimization step (intra-agent decision) during CGP program evolution (cf. [33]).

The optimization for finding the best local decision could, in general, be completed by any algorithm; e.g., by an evolution strategy. For the test cases in [33], the full enumeration of all solution candidates was sufficiently fast enough because each agent had just a rather limited set of choices. Nevertheless, for larger scenarios, the use of problem-specific heuristics has already been recommended in [33]. Constraints can easily be checked as the number of functions and the arity of each function are known to the agent. Currently, we set the number of rows in the graph to one (as in [33]). This convenient single-row representation has been shown to be no less effective [52] and has already been frequently used, e.g., in [53]. The levels-back parameter is set to the number of agents to allow input from all preceding nodes. Each agent knows its own index and may thus decide which other nodes (or program input) to choose as input for its own node.

As soon as the best local allele assignment has been found, the agent compares the solution quality with the quality of the previously found solution. If the new solution is better, messages are sent to neighboring agents. During our experiments, we found that it is advantageous to always send messages to the output agents in order to enable a more frequent update of the best output. The basic difference between the function and the output node agents is the local gene. The output agents just manage a single gene consisting of a single-integer allele that encodes the node that is connected to the respective output.

When no more agents can progress, no messages are sent, and the whole negotiation process finally ceases. Initially, COHDA was supposed to approach an often unknown optimum as closely as possible. For the CGP use case, on the other hand, it is also fine to drop out of the process if one agent, for the first time, finds a solution that fully satisfies a quality condition (i.e., the program does what it is supposed to do). Thus, we added an additional termination criterion. If an agent discovers a solution that constitutes a success, it sends a termination signal instead of a decision message and reports the found solution.

3. Evaluation

3.1. General Approach

For evaluation of the overall approach, The authors in [33] investigated three use cases: regression problems, the n -parity problem, and classification problems. We start with a recap of the results.

3.1.1. Regression

For comparison with the results achieved by the original CGP from [54], a symbolic regression of the following sixth order polynomial was used: $x^6 - 2x^4 + x^2$. The objective here is to evolve a program that produces the same output for the arbitrary input x . The function set that was used consisted of the four basic arithmetic operations: $\{+, -, \cdot, /\}$. For evaluation, 50 input values x were randomly chosen from the interval $[-1, 1]$, and x was the only input to the program. In the original approach, [54] gave also the constant 1.0 as additional input to the program. Previously, ephemeral constants had been used as well to support with solving the problem with GP [55]. Already in [33], it was observed that this additional help is not necessary, and so, we also refrained from using such auxiliary constructs.

For comparison, the following statistical measures as introduced by John Koza [56] were used. The cumulative probability of success for a budget of i objective evaluations is given by

$$P(M, i) = \frac{n_{\text{success}}(i)}{n_{\text{total}}}, \quad (1)$$

where n_{success} denotes the number of successful runs with i objective function calls each, and n_{total} denotes the total number of runs. M denotes the number of individuals. In our use case, M —although possibly interpretable as the number of agents—is of no use as the agent system works asynchronously and not in terms of generations with a constant number of evaluations per iteration. Instead, i was set to the total budget of the maximum number of objective function calls allowed by all of the agents together, and, thus, M was set to $M := 1$. This approach is consistent with the generalization in [57].

From the success rate, the mean number of independent runs required to achieve a minimum rate of success when the budget is fixed to a maximum of i evaluations per run can be derived. Let z denote the wanted success rate. Then,

$$R(z) = \left\lceil \frac{\log(1 - z)}{\log(1 - P(M, i))} \right\rceil \quad (2)$$

gives the number of necessary runs. The computational effort $I(M, i, z) = M \cdot i \cdot R(z)$ gives the number of individual function evaluations that must be performed to solve a problem

to a proportion of z [57]. As i is a matter of parametrization, Koza defines the minimum computational effort as

$$I_{\min}(M, z) = \min_i M \cdot R(z). \quad (3)$$

Table 1 shows the comparison of results yielded from distributed CGP with the original results from [54]. The distributed approach shows competitiveness compared with the original results achieved with a genetic algorithm with a population size of 10, a uniform crossover (100% rate), and a 2% mutation. In [54], the number of maximum generations was set to 8000. This is not meaningful in asynchronous agent systems. In [33], the total number of evaluations was restricted to 80,000 (for 10 agents in total) instead. The confidence level was set to $z = 0.99$.

Table 1. Comparison of the computational effort for symbolic regression between distributed and standard CGP (cf. [33]).

	COHDA	GA [54]
success rate (80,000)	0.97	0.61
minimum computational effort	75,000	90,060
independent runs (budget)	3 (25,000)	6 (15,000)

Table 2 shows some results for several other symbolic regression problems.

Table 2. Results (yielding the minimal computational effort CE) for several symbolic regression problems with 1-dimensional and 2-dimensional input. Modified after [33].

$f(x)$	No. of Agents	Budget	$P(i)$	$R(z)$	min. CE	Mean Effort
$x^2 + 2x + 1$	8	20,000	0.93	2	40,000	7768.4 ± 9944.6
$x^8 + x^5$	20	200,000	0.83	3	600,000	$136,833.2 \pm 129,852.2$
$\frac{x^2}{2x-1}$	30	220,000	0.67	5	1,100,000	$786,270.1 \pm 791,943.8$
$0.2x^2 + 0.5$	20	500,000	0.50	7	3,500,000	$538,625.2 \pm 436,231.5$
$2x_1^2 + x_1x_2$	15	180,000	0.81	3	540,000	$110,931.5 \pm 127,073.4$

Figure 3 explores the relation of the number of agents (and thus mainly the functional nodes) to the mean number of evaluations and to the length of the encoded phenotype solution (the number of active nodes). The experiment was conducted for the simple regression problem $-x^6$ with 100 runs for each different number of agents. Although the mean number of active nodes (calculated only for successful runs) stays rather constant in Figure 3b, an unnecessarily high number of agents leads obviously to some outliers with a bloated phenotype. The mean number of evaluations also grows. Future improvements should address these issues, maybe by starting with a rather low number of agents and adding more agents only in the case that no further improvement to a solution is detected.

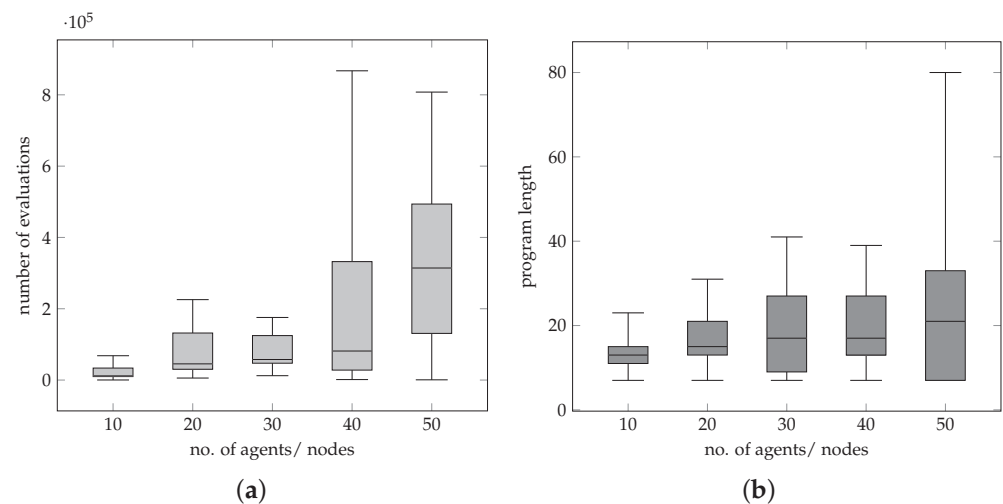


Figure 3. Relation of the number of agents and the mean number of evaluations (a) as well as the relation of the number of agents and the length of resulting phenotypes in (b) – measured by the number of active nodes. Modified after [33].

3.1.2. N-Parity

Evolving Boolean functions is a standard use case in evaluating genetic programming algorithms [58–60]. A special case is the even- n -parity problem [56]. The goal is to find a function that counts the number of ones of the given n bits using only Boolean expressions, returning TRUE if the number is even. In this way, a correct classification of arbitrary bit strings of a length n with an even number of ones is sought [61]. The input to the CGP program are the bits b_0, \dots, b_n . The used function set for program evolution consists of the four Boolean functions $\{AND, OR, NAND, and NOR\}$. The Boolean even- n -parity function is seen as the most difficult Boolean function to evolve [56,62]. With standard GP, results for problem sizes up to $n = 5$ can be obtained [62]. For solving larger instances, techniques such as automatically defined functions [63] or extended function sets are needed [59]. Some results for the CGP programs evolved by COHDA are listed in Table 3. The number of agents and the total budget (for all agents together) are a result of sampling for the minimal computational effort (Equation (3)). Sampling has been carried out by grid search.

Table 3. Results for several instances of the n -parity problem (correctly classifying an even number of 1 in a bit string of length n) for different numbers of agents (cf. [33]).

Size	# of Nodes	No. of Evaluations
even-2	10	522.77 ± 575.61
even-2	20	1640.52 ± 1900.74
even-2	30	3111.77 ± 3366.56
even-3	10	2309.05 ± 1819.84
even-3	20	7834.47 ± 8174.38
even-4	30	$64,772.04 \pm 54,501.49$
even-5	40	$242,264.33 \pm 192,328.68$

So far, distributed CGP had been trained for up to $n = 5$. For smaller instances of the problem, different numbers of agents (the computation nodes) were tested as well. Obviously, having larger chromosomes (the number of agents in our case), as stated in [64], is not always advantageous—at least in the distributed case.

3.1.3. Classification

Finally, a real-world problem from the energy domain was used. The distributed approach was tested on a classification problem known as flexibility modeling [65] and

adapted. The goal is to correctly classify the energy-generation profiles $x \in \mathbb{R}^d$ with x_i , denoting the generated amount of energy during the i th time interval for a given, specific energy resource. A generation profile x can either be feasible (meaning it can be operated by the energy resource without violating any technical constraint) or not. This problem is often modeled as a one-class classification problem [66].

Solutions using support vector machines (SVM) or support vector data description (SVDD) as a classifier can, for example, be found in [67,68]. The approach was compared with SVDD [69]. The classifiers are trained using a set of feasible generation profiles that is generated using an appropriate simulation model of the energy resource. The authors in [33] used the model of a co-generation plant (CHP). This model comprises a micro CHP with 4.7 kW of rated electrical power (12.6 kW thermal power) bundled with a thermal buffer store. Constraints restrict the power band, buffer charging, gradients, min. on and off times, and satisfaction of thermal demand. Thermal demand is a subject to simulate the losses of a detached house according to given weather profiles. For each agent, the CHP model is individually (randomly) configured with a state of charge, weather condition, temperature range, allowed operation gradients, and similar variables [47].

The goal was to evolve a program that obtains d values that represent the amount of energy $x = (x_1, \dots, x_d)$ as inputs and outputs of $y < 0$, if the profile is infeasible to operate for the energy resource, or $y \geq 0$, in the case that the profile is feasible and can thus be operated by the CHP. For evaluation, a training set of 1000 generation profiles (50% feasible) that was generated by the simulation model was used. The function set was: $\{+, -, \cdot, /, \text{AND}, \text{OR}, \text{NOT}, \text{XOR}, =, <, >, \text{IF THEN ELSE}, 0, 1, 2\}$, with 0, 1, and 2 denoting constants (nullary functions).

For evaluation and for comparing the performance of the the classifiers, we used the classification accuracy. Evaluation during CGP evolution was completed using 1000 training instances to calculate the confusion matrix and, finally, the achieved accuracy. The SVDD classifier was trained with 1000 feasible instances. After the classifier program has been evolved, we compared the CGP classifier with the SVDD classifier using 100 times a test set of 1000 newly generated, thus far unseen generation profile instances from identically parameterized simulation models (the same sets for both classifiers, respectively). Table 4 shows the results for the different dimensions d of the generation profile.

Table 4. Comparison of results for the flexibility modeling classification problem (using a model for co-generation plants). We compare distributively evolved CGP program and SVDD classifier (cf. [33]).

Dim.	Agents	Evaluations	Training Accuracy	Accuracy CGP	Accuracy SVDD
8	30	8,351,149	0.89	0.8616 ± 0.0110	0.8770 ± 0.0091
32	50	205,262,361	0.952	0.9569 ± 0.0085	0.9732 ± 0.0048
96	50	68,825,837	0.896	0.9461 ± 0.0109	0.9603 ± 0.0059

Although all CGP results are slightly worse than those of SVDD, the achieved accuracy is still estimable. The training accuracy denotes the best fitness that has been achieved during several test runs of program evolution. The best-found programs were then been applied to the different unseen test sets generated for the newly instantiated CHP models. This generalization ability is compared between CGP and SVDD by their respective mean accuracies. Another interesting observation can be seen in the following example phenotype for eight-dimensional generation profiles as input:

$$y = +(* (x_1, x_3), \text{IF}(x_7, -(/(2.0, -(\text{AND}(x_0, -(/(x_0, x_7), x_7)), *(x_1, x_3))), +(/(x_6, x_7), / (x_7, x_6))), x_0)). \quad (4)$$

Obviously, not all inputs are always of interest for classification as some are constantly omitted. These were always the same ones in different evolved programs. This fact is also reflected by the just marginally growing number of agents compared to the faster-growing

dimensionality of the problem. Such identification of the smaller intrinsic dimension is an extra for the CGP program not provided by classifiers such as SVDD.

3.2. CMA-ES for Optimizing Local Agent Decisions

The covariance matrix adaption evolution strategy [70,71] is well known as an evolution strategy for solving multi-modal black-box problems by using lessons that have been learned from previously seen successful evolution steps for the improvement of future operations. A new population of solution candidates is sampled from a multivariate normal distribution $\mathcal{N}(0, C)$ with the covariance matrix C . This covariance matrix is adapted at the end of each iteration such that it maximizes the generation of improving steps according to previously seen distributions for good steps. The method learns a second-order model of the objective function and exploits it for structure information and for reducing the calls of objective evaluations. Such behavior renders this algorithm especially useful for our purpose as it is used as an inner part (optimizing the local decision of an agent) of a bi-level approach. In this way, CMA-ES is used for calculating the objective of an outer optimization (the agent negotiations). Thus, using as few objective evaluations as possible is advantageous for our use case.

An a priori parametrization with structure knowledge of the problem by the user is not necessary as if the method is adapting itself without supervision. A good introduction to CMA-ES can be found, for example, in [72].

CMA-ES is used for solving the internal optimization problem that arises when an agent has to decide on the best allele combination for the local sub-chromosome that encodes the function controlled by that agent. Thus, the dimension of the problem that has to be solved by CMA-ES is determined by the number of genes in the chromosome, which is rather small compared to the overall number of genes of the global problem that is used for evaluation. Thus, CMA-ES is expected to still work rather quickly and not to suffer from performance degradation due to the huge matrix computations caused by high-dimensional problem instances. In the following, we follow [73,74] with our explanations.

We consider the aforementioned agent negotiation with a stage in which each agent has to search the individual allele configuration for a single node. Thus, the individual feasible set of indices for function encoding and wiring of other outputs to this input for the best option (according to the given objectives) must be found. This search constitutes a local optimization problem. As this smaller sub-problem is a local one, as seen from the agent's perspective, there is no need to harness a distributed solving strategy.

As this local optimization is part of the exterior agent negotiation process, and because it is therefore called many times, a heuristic that uses only a small number of objective evaluations is advantageous. CMA-ES is well known for this characteristic [72]. The constraints of this optimization problem are rather simple box constraints and, thus, easy to handle.

In each iteration g of CMA-ES, a multivariate distribution is sampled in order to generate a new offspring solution population in the σ -vicinity of good parent solutions with the mean m :

$$\mathbf{x}_k^{(g+1)} \sim \mathbf{m}^{(g)} + \sigma^{(g)} \mathcal{N}(0, \mathbf{C}^{(g)}), k = 1, \dots, \lambda. \quad (5)$$

This sampling is suitable for continuous problems. To be able to use CMA-ES for our discrete problem, we allowed continuous alleles and scaled them back to discrete values prior to their assignment to genes in the sub-chromosome. Additionally, we restricted the range to $[0, 1]$:

$$\mathbf{x}_k^{(g+1)} = \begin{cases} \mathbf{x}_k^{(g+1)} \bmod 1, & \mathbf{x}_k^{(g+1)} \geq 1 \\ (\mathbf{x}_k^{(g+1)} \bmod 1) + 1, & \mathbf{x}_k^{(g+1)} < 0 \end{cases}. \quad (6)$$

$\mathbf{C}^{(g)} \in \mathbb{R}^{n \times n}$ constitutes the covariance matrix of the search distribution at a generation (iteration) g with an overall standard deviation $\sigma^{(g)}$ that can also be interpreted in terms of an adaptive (multivariate) step size. The step size is adapted individually for each dimension to support and favor the direction in which fast improvement can be expected

according to the formerly seen results. The mean of the multivariate distribution is denoted by $\mathbf{m}^{(g)}$; $\lambda \geq 2$ denotes the population size.

The new mean $\mathbf{m}^{(g+1)}$ for generating a sample of the next generation in CMA-ES is calculated as the weighted average

$$\mathbf{m}^{(g+1)} = \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}^{(g+1)}, \sum w_i = 1, w_i > 0, \tag{7}$$

of the best (in terms of the objective function evaluation) individuals form the current sample $\mathbf{x}_i^{(g)}, \dots, \mathbf{x}_\lambda^{(g)}$.

In order to make the above relaxed continuous genotype solution candidates discrete again, we introduce a decoder mapping to the respective phenotype (example of agent a_{f_i}):

$$\gamma = \begin{cases} \lfloor \mathbf{x}_{k,i}^{(g+1)} \cdot |F| \rfloor & i = 0 \\ \lfloor \mathbf{x}_{k,i}^{(g+1)} \cdot \text{index}(a_{f_i}) \rfloor & i > 0 \end{cases}, \quad 0 \leq I < 1 + \text{max arity}(F) \tag{8}$$

where $|F|$ denotes the number of functions in the function set F , $\text{index}(a_{f_i})$ denotes the number of agents that are ahead of agent a_{f_i} (only from these may the input be taken), and $\text{max arity}(F)$ denotes the maximum arity of all functions in the set. In this way, the solution candidate is scaled back to a valid discrete genotype.

In our case, the genotype consists of a sub-chromosome with the zeroth gene encoding the function and genes 1 to $\text{max arity}(F) + 1$ encoding the wiring with the previous (in the calculation line). Functions with an arity lower than the maximum arity do not use all wirings. This approach is in line with [75].

Ranking is now carried out by

$$f(\gamma(\mathbf{x}_{1:\lambda}^{(g)}), \kappa), \dots, f(\gamma(\mathbf{x}_{\lambda:\lambda}^{(g)}), \kappa), \lambda \geq \mu, \tag{9}$$

to define $\mathbf{x}_{i:\lambda}^{(g)}$ as the i th-ranked best individual. Please note that for evaluation, the global objective is used. The evaluation of the global objective takes the individual sub-chromosomes of all agents and combines them. Thus, κ in Equation (9) denotes the current working memory of the agent. κ contains the temporarily fixed alleles of the other agents (so far known from the perception phase).

Finally, the covariance matrix is updated as usual and is also based on the decoder-based ranking Equation (9):

$$\mathbf{C}_\mu^{(g+1)} = \sum_{i=1}^{\mu} w_i (\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)}) (\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)})^\top. \tag{10}$$

CMA-ES has a set of parameters that can be tweaked to some degree for a problem-specific adaption. Nevertheless, the default values that are applicable for a wide range of problems are usually available. For our experiments, we used the following default settings for the CMA-ES part. The (external) strategy parameters are $\lambda, \mu, w_{i=1 \dots \mu}$, controlling selection and recombination; c_σ and d_σ controlling the step size; and c_c and μ_{cov} controlling the covariance matrix adaption. We have chosen to set these values after [72]:

$$\lambda = 4 + \lfloor 3 \ln n \rfloor, \quad \mu = \left\lceil \frac{\lambda}{2} \right\rceil, \tag{11}$$

$$w_i = \frac{\ln(\frac{\lambda}{2} + 0.5) - \ln i}{\sum_{j=1}^{\mu} \frac{\lambda}{2} + 0.5 - \ln j}, \quad i = 1, \dots, \mu \tag{12}$$

$$C_c = \frac{4}{n + 4}, \quad \mu_{\text{cov}} = \mu_{\text{eff}}, \tag{13}$$

$$C_{\text{cov}} = \frac{1}{\mu_{\text{cov}}} \frac{2}{(n + \sqrt{2})^2} + \left(1 - \frac{1}{\mu_{\text{cov}}}\right) \min\left(1, \frac{2\mu_{\text{cov}} - 1}{(n + 2)^2 + \mu_{\text{cov}}}\right). \quad (14)$$

An in-depth discussion of these parameters is also given in [76]. These settings are specific to the dimension N of the objective function. In our case, $N = 1 + \max \text{arity}(F)$ is related to the maximum arity of all functions in the function set plus one gene for encoding the function itself. Thus, the dimensionality stays rather small. Hence, CMA-ES will not suffer from large matrix calculations for updating the covariance matrix as in high-dimensional problems.

3.3. Results

For the evaluation of the CMA-ES approach, we used the classification use case described above, as this is the most practically relevant use case from [33]. In order to make the problem more severe, we added five NOP to the function set, causing no harm to the result but making the search space exponentially larger. In addition, we used more agents to further increase the search space (due to more wiring choices). This is also quite realistic for the use case of an agent swarm that is supposed to learn how to solve new, unseen problems. An agent may offer abilities and functions that are of no use for solving the problem but increase the search space.

As there is no known minimum in this optimization problem, we chose to stop the process as soon as the objective value of 0.18 has been found. This value has been empirically found. We observed that, once a CGP program with an objective value of less than or equal to 0.18 has been found, the program shows a sufficiently good performance in classification. The absolute number of objective evaluations of all agents was counted to calculate Koza's minimum effort statistics.

The results showed that full enumeration—except for very small problem instances—had to be stopped at some point (we chose 5 million evaluations) without a valid result. Table 5 shows the result for a small four-dimensional classification problem.

Table 5. Comparison of full enumeration and CMA-ES for a very small example (4-dimensional classification with 30 agents).

Budget	$P(i)$	$R(z)$	Computational Effort
full enumeration			
1,000,000	0.31	13	1.3×10^7
2,000,000	0.42	10	2.0×10^7
4,000,000	0.79	3	1.2×10^7
CMA-ES			
10,000	0.27	15	150,000
11,000	0.53	7	77,000
12,000	0.66	5	60,000
13,000	0.87	3	39,000

This example already shows the vast improvement of CMA-ES over full enumeration. The minimum computational effort that the agents (in total) needed with CMA-ES is more than 300 times smaller than with full enumeration. For larger examples, the full enumeration approach was not at all able to obtain a sufficiently good result with a reasonable budget for the objective evaluations. For this reason, we omitted full enumeration in the remaining results. Tables 6 and 7 finally show some results for larger classification problems of CMA-ES only.

Table 6. Results for CMA-ES only for a 32-dimensional classification problem with 100 agents.

Budget	$P(i)$	$R(z)$	Computational Effort
32,000	0.27	15	480,000
40,000	0.64	5	200,000
45,000	0.72	4	180,000
50,000	0.91	2	100,000

Table 7. Results for CMA-ES only for a 32-dimensional classification problem with 200 agents.

Budget	$P(i)$	$R(z)$	Computational Effort
15,000	0.08	53	795,000
17,000	0.33	12	204,000
19,000	0.75	4	76,000
20,000	0.92	2	40,000

3.4. Analysis

Finally, we analyzed the individual complexity that the agents face during different episodes of the negotiation. This complexity is not constant. It depends on the previous choices of all other agents. If it is an agent's turn to make a decision—i.e., to optimize its own function choice, including the wiring of the parameters to the other agent's functions—the fitness landscape is defined as follows

$$F = (S, f, d), \quad (15)$$

with the search space S , which always stays the same (the set of all combinations of the functions with the allowed parameter wirings) and the neighborhood definition d . For our research, we used the following neighborhood:

$$x_i^{k+1} = \begin{cases} x_i^k + 1 & \text{if } r \leq \frac{1}{3} \\ x_i^k & \text{if } \frac{1}{3} < r \leq \frac{2}{3}, \\ x_i^k - 1 & \text{otherwise} \end{cases}, \quad 1 \leq i \leq d. \quad (16)$$

Here, x^{k+1} denotes a neighboring solution that is generated from a solution candidate x^k . The random variable r is uniformly randomly sampled from the interval $[0, 1]$.

In this way, each allele is increased by one, decreased by one or stays the same with a likelihood of $1/3$ each. The element that changes in the landscape definition F is the objective function f . The objective f is still defined as described above for the classification use case. The classification accuracy (which is used for f) is calculated using the performance of a solution candidate on different so-far unseen classification instances. However, this performance highly depends on the decisions of all other agents. Thus, the fitness landscape varies with each call of the decision method of an agent.

The analysis then was performed as follows. Prior to each decision method call, the modality (as a measure of ruggedness) of the local problem of each agent was calculated. To complete this, we followed the method of [77]. First, we generated a series of fitness values by using a random path of the solution candidates on the fitness landscape. To generate the random path, the neighborhood relation d from Equation (16) was used. For each solution candidate along the generated path, the fitness can be calculated using the objective function with the fixed parts of the other agents. We can then tokenize the series of all subsequent fitness values to a sequence of tokens encoding uphill, downhill, or flat episodes. The relation of the length of the shortest possible token string (containing up, down, and flat tokens) to the length of the original path or series of fitness values then gives a measure for the modality $\rho \in [0, 1]$. $\rho = 0$ denotes a completely flat fitness landscape; $\rho = 1$, on the other hand, denotes the maximum number of local optima that fit into a path of the given length.

This modality is calculated for each local decision of each agent. Figure 4 shows an example result for a first, simple instance of the classification problem. Here, we looked at a four-dimensional classification problem and used 20 agents for evolving a program that does the classification of CHP schedule feasibility. Time-steps (as a unit for time), in this case, are an artificial measure. The multi-agent systems work asynchronously, but, we can maintain an artificial clock tick that could be interpreted as a unit of time (second, millisecond, ...) but, in fact, has no practical meaning here except for ordering the events.

The experiment has been repeated four times, as depicted in Figure 4a–d. We see that the modality varies over time and becomes lower toward the end of the evolution process. The latter fact is not immediately clear. Although toward the end more and more agents fix their local result because they are unable to find any further improvement, it is not clear why this should make the search for the remaining agents easier.

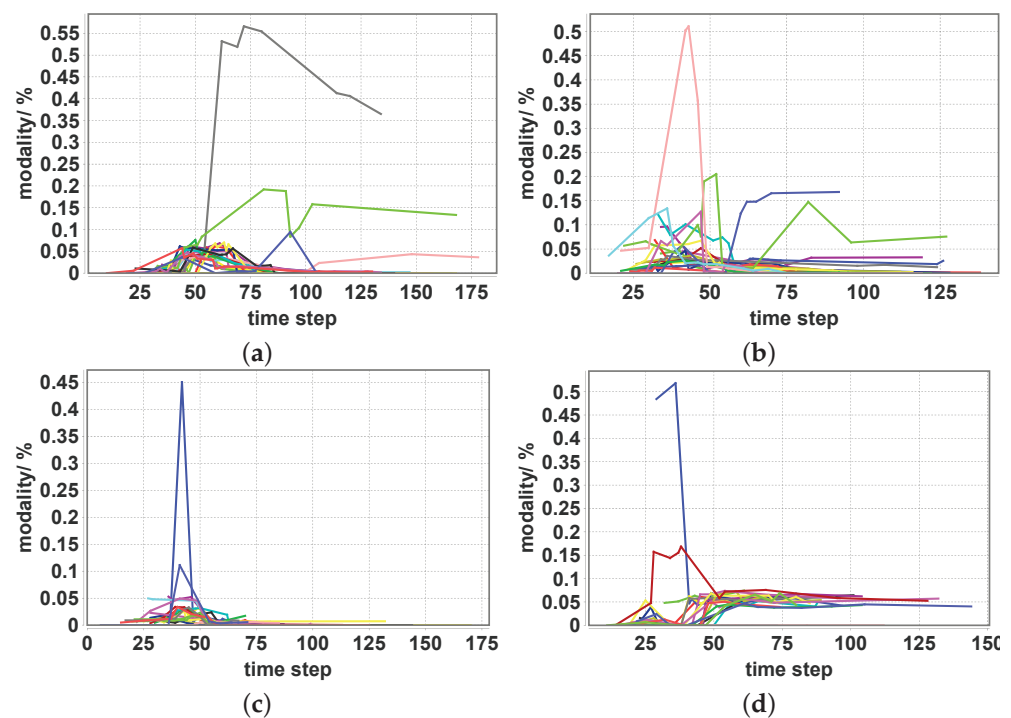


Figure 4. Variation of the modality of the different agents during the program evolution process for 4 instances of a 4-dimensional classification problem solved with 20 agents each. (a–d) display one instance each with the traces of the individual modality (one line per agent).

Nevertheless, this seems to be a general pattern. Figures 5–7 show some more examples of 96-dimensional classification problem instances. These figures show the same general pattern. Moreover, it seems that if the number of agents is increased, the phase of high modality in local optimization moves to earlier stages. The number of agents varies from 50 (used for the result in Figure 5) to 200 (which is already way too many; see Figure 7).

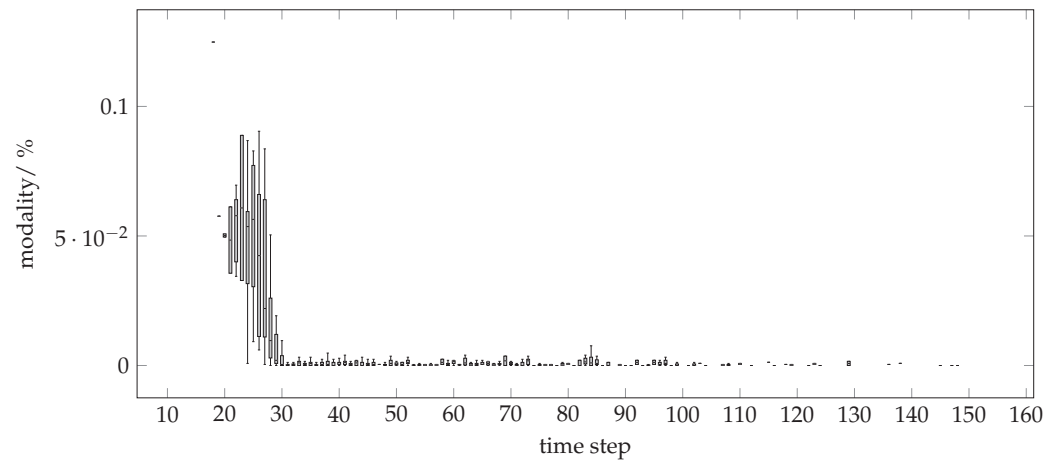


Figure 5. Variation of the modality for an example instance of a 96-dimensional classification problem solved by 50 agents.

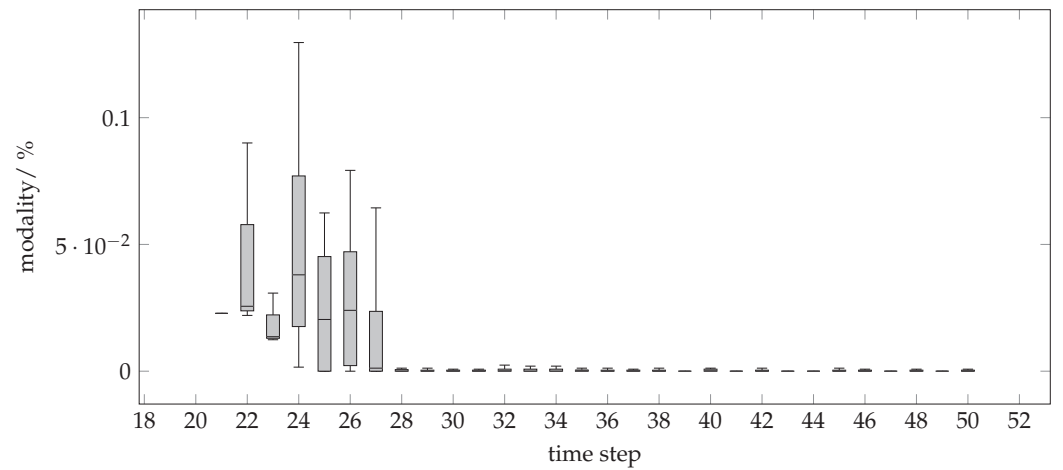


Figure 6. Variation of the modality for an example instance of a 96-dimensional classification problem solved by 100 agents.

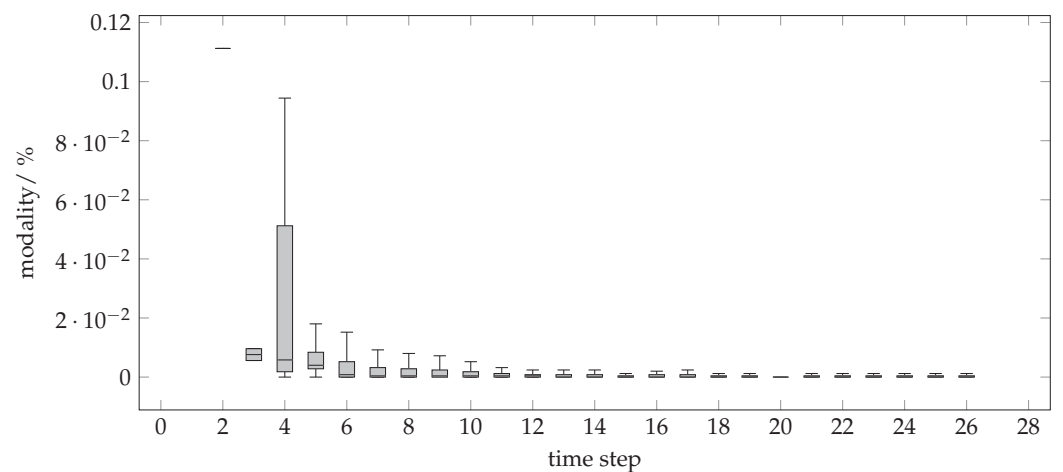


Figure 7. Variation of the modality for an example instance of a 96-dimensional classification problem solved by 200 agents.

We can conclude two things:

1. The choice of CMA-ES as an algorithm that adapts without supervision to different problem instances with different characteristics was good because the modality of the

local optimization problems that the agents have to solve comprises a wide range of modalities.

2. It seems worthwhile to conduct a larger and more thorough fitness landscape analysis in order to develop situation-aware and adaptive local decision support for the agents.

Thus, future work will address a fitness landscape-aware selection of different local optimization techniques for the agents.

4. Conclusions

We presented the adaption of a distributed optimization heuristic protocol for Cartesian genetic programming and an extension using CMA-ES for improving local agent decisions. By decomposing the evolution on an algorithmic level, it becomes possible to distribute the nodes and regard the evolution process as a parallel, asynchronous execution of an individual coordinate's descent.

The results show that the distributed approach is competitive with regard to the evolved programs. This holds true for the solution quality as well as for the computational effort and for smaller tasks even with the full enumeration approach. A speed-up by parallel execution becomes possible and has been increased significantly here. With the extension via CMA-ES for agent-internal optimization during decision-making, the computational effort has dropped significantly, even for larger problem instances. Another advantage is the ability to distribute the computational burden. Moreover, the distributed evolution of programs enables seamless integration into distributed applications, in addition to using the example of the smart grid.

Agent-based Cartesian genetic programming constitutes a universal means to execute cooperative planning among individually acting entities. Future work will now lean toward distributed use cases. Another advantage is that different nodes may have different function sets in case they represent real-world nodes with different capabilities.

So far, we considered only the original standard CGP. Extensions such as recurrent CGP can be integrated right away now. These extensions affect basic interpretation (some also affect the execution of the phenotype) and can thus be evolved by the same distributed approach. Only an adaption of the possible choices of other agents' output as input for its own node is necessary. In the same way, different levels-back parameterizations can be easily handled.

Further improvements are expected when agents are equipped with intelligent rules for choosing from different decision methods; i.e., optimization methods (CMA-ES, full enumeration, others) should be chosen ad hoc from a set of different methods according to the current individual's situation.

However, even with this initial setting that has been scrutinized in this contribution for making a swarm of individually acting agents learn problem-solving via distributed CGP, the results are already very promising. Agents capable of combining individual skills to solve so-far unseen problems without any central algorithm are a major building block for large-scale future autonomous cyber-physical systems.

Author Contributions: Conceptualization, J.B. and S.L.; methodology, J.B.; software, J.B. and S.L.; validation, J.B. and S.L.; formal analysis, J.B.; investigation, J.B. and S.L.; resources, S.L.; data curation, J.B. and S.L.; writing—original draft preparation, J.B.; writing—review and editing, S.L.; visualization, J.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: An implementation of the used COHDA algorithm can be found at <https://gitlab.com/mango-agents/> (accessed on 19 February 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Miller, J.F. An empirical study of the efficiency of learning boolean functions using a cartesian genetic programming approach. In Proceedings of the Genetic and Evolutionary Computation Conference, Orlando, FL, USA, 13–17 July 1999; Volume 2, pp. 1135–1142.
2. Miller, J.F. Cartesian genetic programming: Its status and future. *Genet. Program. Evolvable Mach.* **2020**, *21*, 129–168. [[CrossRef](#)]
3. Manazir, A.; Raza, K. Recent developments in cartesian genetic programming and its variants. *ACM Comput. Surv. (CSUR)* **2019**, *51*, 1–29. [[CrossRef](#)]
4. Khan, M.M.; Ahmad, A.M.; Khan, G.M.; Miller, J.F. Fast learning neural networks using cartesian genetic programming. *Neurocomputing* **2013**, *121*, 274–289. [[CrossRef](#)]
5. Miller, J.F.; Mohid, M. Function Optimization Using Cartesian Genetic Programming. In Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation, Amsterdam, The Netherlands, 6–10 July 2013; Association for Computing Machinery: New York, NY, USA, 2013; GECCO '13 Companion, pp. 147–148. [[CrossRef](#)]
6. Clegg, J.; Walker, J.A.; Miller, J.F. A new crossover technique for cartesian genetic programming. In Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, London, UK, 7–11 July 2007; pp. 1580–1587.
7. Calvaresi, D.; Appoggetti, K.; Lustrissimini, L.; Marinoni, M.; Sernani, P.; Dragoni, A.F.; Schumacher, M. Multi-Agent Systems' Negotiation Protocols for Cyber-Physical Systems: Results from a Systematic Literature Review. In Proceedings of the ICAART (1), Funchal, Portugal, 16–18 January 2018; pp. 224–235.
8. Zhu, Q.; Bushnell, L.; Başar, T. Resilient distributed control of multi-agent cyber-physical systems. In *Control of Cyber-Physical Systems*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 301–316.
9. Nieße, A.; Lehnhoff, S.; Tröschel, M.; Uslar, M.; Wissing, C.; Appelrath, H.J.; Sonnenschein, M. Market-based self-organized provision of active power and ancillary services: An agent-based approach for Smart Distribution Grids. In Proceedings of the 2012 Complexity in Engineering (COMPENG), Aachen, Germany, 11–13 June 2012; pp. 1–5. [[CrossRef](#)]
10. Bremer, J. Learning to Optimize. In *Recent Advances in Computational Optimization: Results of the Workshop on Computational Optimization WCO 2021*; Fidanova, S., Ed.; Springer International Publishing: Cham, Switzerland, 2022; pp. 1–19. [[CrossRef](#)]
11. Jipp, M.; Ackerman, P.L. The Impact of Higher Levels of Automation on Performance and Situation Awareness. *J. Cogn. Eng. Decis. Mak.* **2016**, *10*, 138–166. [[CrossRef](#)]
12. Sheridan, T.B.; Parasuraman, R. Human-Automation Interaction. *Rev. Hum. Factors Ergon.* **2016**, *1*, 89–129. [[CrossRef](#)]
13. European Commission. *Draft Ethics Guidelines for Trustworthy AI*; Technical Report; European Commission: Brussels, Belgium, 2018.
14. Rapp, B.; Solsbach, A.; Mahmoud, T.; Memari, A.; Bremer, J. IT-for-Green: Next Generation CEMIS for Environmental, Energy and Resource Management. In Proceedings of the EnviroInfo 2011–Innovations in Sharing Environmental Observation and Information, Proceedings of the 25th EnviroInfo Conference 'Environmental Informatics', Ispra, Italy, 5–7 October 2011; Pillmann, W., Schade, S., Smits, P., Eds.; Shaker Verlag: Herzogenrath, Germany, 2011; pp. 573–581.
15. Gholami, M.; Pilloni, A.; Pisano, A.; Sanai Dashti, Z.A.; Usai, E. Robust Consensus-Based Secondary Voltage Restoration of Inverter-Based Islanded Microgrids with Delayed Communications. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, USA, 17–19 December 2018; IEEE Press: Manhattan, NY, USA, 2018; pp. 811–816. [[CrossRef](#)]
16. Gordon, M.A.; Vargas, F.J.; Peters, A.A.; Maass, A.I. Platoon Stability Conditions Under Inter-vehicle Additive Noisy Communication Channels. *IFAC-PapersOnLine* **2020**, *53*, 3150–3155. [[CrossRef](#)]
17. Coppola, A.; Lui, D.G.; Petrillo, A.; Santini, S. Cooperative driving of heterogeneous uncertain nonlinear connected and autonomous vehicles via distributed switching robust PID-like control. *Inf. Sci.* **2023**, *625*, 277–298. [[CrossRef](#)]
18. Platzer, A. The logical path to autonomous cyber-physical systems. In Proceedings of the International Conference on Quantitative Evaluation of Systems, Glasgow, UK, 9–12 September 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 25–33.
19. McKee, D.W.; Clement, S.J.; Almutairi, J.; Xu, J. Survey of advances and challenges in intelligent autonomy for distributed cyber-physical systems. *CAAI Trans. Intell. Technol.* **2018**, *3*, 75–82. [[CrossRef](#)]
20. Collier, J. Fundamental properties of self-organization. *Causality Emerg. Self-Organ.* **2003**, 287–302.
21. Parzyjegla, H.; Schröter, A.; Seib, E.; Holzapfel, S.; Wander, M.; Richling, J.; Wacker, A.; Heiß, H.U.; Mühl, G.; Weis, T. Model-driven development of self-organising control applications. In *Organic Computing—A Paradigm Shift for Complex Systems*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 131–144.
22. Bremer, J.; Lehnhoff, S. Towards Evolutionary Emergence. *Ann. Comput. Sci. Inf. Syst.* **2021**, *26*, 55–60.
23. Turner, A.J.; Miller, J.F. Recurrent Cartesian Genetic Programming. In Proceedings of the Parallel Problem Solving from Nature—PPSN XIII, Ljubljana, Slovenia, 13–17 September 2014; Bartz-Beielstein, T., Branke, J., Filipič, B., Smith, J., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 476–486.
24. Goldman, B.W.; Punch, W.F. Reducing wasted evaluations in cartesian genetic programming. In Proceedings of the European Conference on Genetic Programming, 2013, Vienna, Austria, 3–5 April 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 61–72.
25. Walker, J.A.; Völk, K.; Smith, S.L.; Miller, J.F. Parallel evolution using multi-chromosome cartesian genetic programming. *Genet. Program. Evolvable Mach.* **2009**, *10*, 417. [[CrossRef](#)]
26. Poli, R. *Parallel Distributed Genetic Programming*; University of Birmingham, Cognitive Science Research Centre: Birmingham, UK, 1996.

27. Liu, Z.; Zhan, X.; Han, T.; Yan, H. Distributed Adaptive Finite-Time Bipartite Containment Control of Linear Multi-Agent Systems. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *69*, 4354–4358. [[CrossRef](#)]
28. Liu, G.; Liang, H.; Pan, Y.; Ahn, C.K. Antagonistic Interaction-Based Bipartite Consensus Control for Heterogeneous Networked Systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 71–81. [[CrossRef](#)]
29. Lui, D.G.; Petrillo, A.; Santini, S. Bipartite Tracking Consensus for High-Order Heterogeneous Uncertain Nonlinear Multi-Agent Systems With Unknown Leader Dynamics via Adaptive Fully-Distributed PID Control. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 1131–1142. [[CrossRef](#)]
30. Yang, L.; Li, X.; Sun, M.; Sun, C. Hybrid Policy-based Reinforcement Learning of Adaptive Energy Management for the Energy Transmission-constrained Island Group. *IEEE Trans. Ind. Inform.* **2023**, 1–12. [[CrossRef](#)]
31. Zhang, N.; Sun, Q.; Yang, L.; Li, Y. Event-Triggered Distributed Hybrid Control Scheme for the Integrated Energy System. *IEEE Trans. Ind. Inform.* **2022**, *18*, 835–846. [[CrossRef](#)]
32. Cerquides, J.; Farinelli, A.; Meseguer, P.; Ramchurn, S.D. A Tutorial on Optimization for Multi-Agent Systems. *Comput. J.* **2013**, *57*, 799–824.
33. Bremer, J.; Lehnhoff, S. Fully Distributed Cartesian Genetic Programming. In Proceedings of the Advances in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation, The PAAMS Collection, L'Aquila, Italy, 13–15 July 2022; Dignum, F., Mathieu, P., Corchado, J.M., De La Prieta, F., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 36–49.
34. Talbi, E.G. *Metaheuristics: From Design to Implementation*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 74.
35. Hinrichs, C.; Vogel, U.; Sonnenschein, M. Approaching Decentralized Demand Side Management via Self-Organizing Agents. In Proceedings of the ATEs Workshop, Proceedings of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, 2–6 May 2011.
36. Hinrichs, C.; Sonnenschein, M.; Lehnhoff, S. Evaluation of a Self-Organizing Heuristic for Interdependent Distributed Search Spaces. In Proceedings of the International Conference on Agents and Artificial Intelligence (ICAART 2013), Barcelona, Spain, 15–18 February 2013; Filipe, J., Fred, A.L.N., Eds.; SciTePress: Setúbal, Portugal 2013; Volume 1, pp. 25–34. [[CrossRef](#)]
37. Potter, M.A.; Jong, K.A.D. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evol. Comput.* **2000**, *8*, 1–29. [[CrossRef](#)]
38. Bremer, J.; Lehnhoff, S. The Effect of Laziness on Agents for Large Scale Global Optimization. In Proceedings of the Agents and Artificial Intelligence, Prague, Czech Republic, 19–21 February 2019; van den Herik, J., Rocha, A.P., Steels, L., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 317–337.
39. Sotito, L.F.D.P.; Kaufmann, P.; Atkinson, T.; Kalkreuth, R.; Basgalupp, M.P. A Study on Graph Representations for Genetic Programming. In Proceedings of the 2020 Genetic and Evolutionary Computation Conference, Cancún, Mexico, 8–12 July 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 931–939. [[CrossRef](#)]
40. Miller, J. *Cartesian Genetic Programming*; Springer: Berlin/Heidelberg, Germany, 2003; Volume 4. [[CrossRef](#)]
41. Miller, J.F.; Thomson, P.; Fogarty, T. Designing electronic circuits using evolutionary algorithms. arithmetic circuits: A case study. In *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*; John Wiley & Sons: Hoboken, NJ, USA, 1997; pp. 105–131.
42. Harding, S.; Leitner, J.; Schmidhuber, J. Cartesian genetic programming for image processing. In *Genetic Programming Theory and Practice X*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 31–44.
43. Harding, S.; Banzhaf, W.; Miller, J.F. A survey of self modifying cartesian genetic programming. In *Genetic Programming Theory and Practice VIII*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 91–107.
44. Hinrichs, C.; Sonnenschein, M. A distributed combinatorial optimisation heuristic for the scheduling of energy resources represented by self-interested agents. *Int. J. Bio-Inspired Comput.* **2017**, *10*, 69–78. [[CrossRef](#)]
45. Bremer, J.; Lehnhoff, S. Decentralized Coalition Formation with Agent-based Combinatorial Heuristics. *ADCAIJ Adv. Distrib. Comput. Artif. Intell. J.* **2017**, *6*, 29–44. [[CrossRef](#)]
46. Tong, B.; Liu, Q.; Dai, C.; Jia, Z. A Decentralized Multiple MAV Collision Avoidance Trajectory Planning Method. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 1545–1552. [[CrossRef](#)]
47. Bremer, J.; Lehnhoff, S. Decentralized Surplus Distribution Estimation with Weighted k-Majority Voting Games. In *Highlights of Practical Applications of Cyber-Physical Multi-Agent Systems, Proceedings of the International Workshops of PAAMS 2017, Porto, Portugal, 21–23 June 2017*; Bajo, J., Vale, Z., Hallenborg, K., Rocha, A.P., Mathieu, P., Pawlewski, P., Del Val, E., Novais, P., Lopes, F., Duque Méndez, N.D., et al., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 327–339. [[CrossRef](#)]
48. Sonnenschein, M.; Lünsdorf, O.; Bremer, J.; Tröschel, M. Decentralized Control of Units in Smart Grids for the Support of Renewable Energy Supply. *Environ. Impact Assess. Rev.* **2015**, *52*, 40–52. [[CrossRef](#)]
49. Watts, D.; Strogatz, S. Collective dynamics of ‘small-world’ networks. *Nature* **1998**, *440*, 440–442. [[CrossRef](#)] [[PubMed](#)]
50. Nieße, A.; Beer, S.; Bremer, J.; Hinrichs, C.; Lünsdorf, O.; Sonnenschein, M. Conjoint Dynamic Aggregation and Scheduling Methods for Dynamic Virtual Power Plants. In Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, Warsaw, Poland, 7–10 September 2014; Ganzha, M., Maciaszek, L.A., Paprzycki, M., Eds.; IEEE: Piscataway, NJ, USA, 2014; Volume 2, pp. 1505–1514. [[CrossRef](#)]
51. Bremer, J.; Lehnhoff, S. Lazy Agents for Large Scale Global Optimization. In Proceedings of the ICAART, Prague, Czech Republic, 19–21 February 2019; pp. 72–79.

52. Oranchak, D. Cartesian Genetic Programming for the Java Evolutionary Computing Toolkit (CGP for ECJ). 2010. Available online: <http://www.oranchak.com/cgp/doc/> (accessed on 19 February 2023).
53. Inácio, T.; Miragaia, R.; Reis, G.; Grilo, C.; Fernández, F. Cartesian genetic programming applied to pitch estimation of piano notes. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016; pp. 1–7.
54. Miller, J.F.; Thomson, P. Cartesian Genetic Programming. In Proceedings of the EuroGP, Scotland, UK, 15–16 April 2000; Springer: Berlin/Heidelberg, Germany, 2000; Volume 1802, pp. 121–132.
55. Koza, J.R. *Genetic Programming II: Automatic Discovery of Reusable Programs*; MIT Press: Cambridge, MA, USA, 1994.
56. Koza, J.R.; Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press: Cambridge, MA, USA, 1992; Volume 1.
57. Christensen, S.; Oppacher, F. An Analysis of Koza’s Computational Effort Statistic for Genetic Programming. In *Genetic Programming, Proceedings of the 5th European Conference, EuroGP 2002, Kinsale, Ireland, 3–5 April 2002*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 182–191. [[CrossRef](#)]
58. Gathercole, C.; Ross, P. Tackling the boolean even N parity problem with genetic programming and limited-error fitness. *Genet. Program.* **1997**, *97*, 119–127.
59. Poli, R.; Page, J. Solving high-order boolean parity problems with smooth uniform crossover, sub-machine code GP and demes. *Genet. Program. Evolvable Mach.* **2000**, *1*, 37–56. [[CrossRef](#)]
60. Mambrini, A.; Oliveto, P.S. On the Analysis of Simple Genetic Programming for Evolving Boolean Functions. In Proceedings of the Genetic Programming, 2016, Porto, Portugal, 30 March–1 April 2016; Heywood, M.I., McDermott, J., Castelli, M., Costa, E., Sim, K., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 99–114.
61. Parent, J.; Nowé, A.; Defaweux, A. Addressing the Even-n-parity problem using Compressed Linear Genetic Programming. In Proceedings of the Late Breaking Paper at Genetic and Evolutionary Computation Conference (GECCO’2005), Washington, DC, USA, 25–29 June 2005; pp. 25–29.
62. Muntean, O.; Diosan, L.; Oltean, M. Solving the even-n-parity problems using Best SubTree Genetic Programming. In Proceedings of the Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007), Edinburgh, UK, 5–8 August 2007; pp. 511–518.
63. Koza, J.R.; Andre, D.; Bennett III, F.H.; Keane, M.A. Use of automatically defined functions and architecture-altering operations in automated circuit synthesis with genetic programming. In Proceedings of the First Annual Conference on Genetic Programming, Cambridge, MA, USA, 28–31 July 1996; Stanford University MIT Press: Cambridge, MA, USA, 1996; pp. 132–140.
64. Miller, J.F.; Smith, S.L. Redundancy and computational efficiency in cartesian genetic programming. *IEEE Trans. Evol. Comput.* **2006**, *10*, 167–174. [[CrossRef](#)]
65. Chakraborty, I.; Nandanoori, S.P.; Kundu, S.; Kalsi, K., Data-Driven Predictive Flexibility Modeling of Distributed Energy Resources. In *Artificial Intelligence Techniques for a Scalable Energy Transition: Advanced Methods, Digital Technologies, Decision Support Tools, and Applications*; Sayed-Mouchaweh, M., Ed.; Springer International Publishing: Cham, Switzerland, 2020; pp. 311–343. [[CrossRef](#)]
66. Bremer, J.; Sonnenschein, M. Model-based integration of constrained search spaces into distributed planning of active power provision. *Comput. Sci. Inf. Syst.* **2013**, *10*, 1823–1854. [[CrossRef](#)]
67. Pinto, R.; Matos, M.A.; Bessa, R.J.; Gouveia, J.; Gouveia, C. Multi-period modeling of behind-the-meter flexibility. In Proceedings of the 2017 IEEE Manchester PowerTech, Manchester, UK, 18–22 June 2017; pp. 1–6. [[CrossRef](#)]
68. Bremer, J.; Rapp, B.; Sonnenschein, M. Encoding distributed search spaces for virtual power plants. In Proceedings of the 2011 IEEE Symposium Series on Computational Intelligence (SSCI), Computational Intelligence Applications in Smart Grid (CIASG), Paris, France, 11–15 April 2011. [[CrossRef](#)]
69. Tax, D.M.J.; Duin, R.P.W. Support Vector Data Description. *Mach. Learn.* **2004**, *54*, 45–66. .:MACH.0000008084.60811.49. [[CrossRef](#)]
70. Ostermeier, A.; Gawelczyk, A.; Hansen, N. A Derandomized Approach to Self-Adaptation of Evolution Strategies. *Evol. Comput.* **1994**, *2*, 369–380. [[CrossRef](#)]
71. Hansen, N. The CMA evolution strategy: A comparing review. In *Towards a New Evolutionary Computation. Advances on Estimation of Distribution Algorithms*; Lozano, J., Larranaga, P., Inza, I., Bengoetxea, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 75–102.
72. Hansen, N. The CMA Evolution Strategy: A Tutorial. *arXiv* **2016**, arXiv:1604.00772. <https://arxiv.org/abs/1604.00772>
73. Hansen, N.; Auger, A. CMA-ES: Evolution strategies and covariance matrix adaptation. In Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation, Dublin, Ireland, 12–16 July 2011; pp. 991–1010.
74. Bremer, J.; Lehnhoff, S. Hybrid Multi-ensemble Scheduling. In *Applications of Evolutionary Computation, Proceedings of the 20th European Conference, EvoApplications 2017, Amsterdam, The Netherlands, 19–21 April 2017*; Squillero, G., Sim, K., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 342–358.
75. Miller, J.; Series, N.C. Resources for cartesian genetic programming. In *Cartesian Genetic Programming*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 337–339.

76. Hansen, N.; Ostermeier, A. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evol. Comput.* **2001**, *9*, 159–195. [[CrossRef](#)]
77. Vassilev, V.K.; Fogarty, T.C.; Miller, J.F. Information Characteristics and the Structure of Landscapes. *Evol. Comput.* **2000**, *8*, 31–60. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

It's All about Reward: Contrasting Joint Rewards and Individual Reward in Centralized Learning Decentralized Execution Algorithms

Peter Atrazhev and Petr Musilek *

Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada

* Correspondence: pmusilek@ualberta.ca

Abstract: This paper addresses the issue of choosing an appropriate reward function in multi-agent reinforcement learning. The traditional approach of using joint rewards for team performance is questioned due to a lack of theoretical backing. The authors explore the impact of changing the reward function from joint to individual on learning centralized decentralized execution algorithms in a Level-Based Foraging environment. Empirical results reveal that individual rewards contain more variance, but may have less bias compared to joint rewards. The findings show that different algorithms are affected differently, with value factorization methods and PPO-based methods taking advantage of the increased variance to achieve better performance. This study sheds light on the importance of considering the choice of a reward function and its impact on multi-agent reinforcement learning systems.

Keywords: agent coordination; multi-agent reinforcement learning; centralized learning decentralized execution

Citation: Atrazhev, P.; Musilek, P. It's All about Reward: Contrasting Joint Rewards and Individual Reward in Centralized Learning Decentralized Execution Algorithms. *Systems* **2023**, *11*, 180. <https://doi.org/10.3390/systems11040180>

Academic Editors: Philippe Mathieu, Juan M. Corchado, Alfonso González-Briones and Fernando De la Prieta Pintado

Received: 1 February 2023

Revised: 22 March 2023

Accepted: 23 March 2023

Published: 30 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Multi-agent reinforcement learning (MARL) is a promising field of artificial intelligence (AI) research, and over the last couple of years, has seen increasingly more pushes to tackle less “toy” problems (full game environments such as ATARI and the Starcraft Multi-Agent Environment (SMAC)) and instead try to solve complex “real-world” problems [1–3]. Coordination of agents across a large state space is a challenging and multifaceted problem, with many approaches that can be used to increase coordination. These include communication between agents, both learned and established, parameter sharing and other methods of imparting additional information to function approximators, and increasing levels of centralization.

One paradigm of MARL that aims to increase coordination is called centralized Learning decentralized Execution (CLDE) [4]. CLDE algorithms train their agents’ policies with additional global information using a centralized mechanism. During execution, the centralized element is removed, and the agent’s policy is based on conditions only on local observations. This has been shown to increase the coordination of agents [5]. CLDE algorithms separate into two major categories: centralized policy gradient methods [6–8] and value decomposition methods [9,10]. Recently, however, there has been work that has put into question the assumption that centralized mechanisms do indeed increase coordination. Lyu et al. [11] found that in actor–critic systems, the use of a centralized critic led to an increase in variance seen in the final policy learned; however, they noted more coordinated agent behaviour while training and concluded that the use of a centralized critic should be thought of as a choice that carries with it a bias variance trade-off.

One aspect of agent coordination that is similarly often taken at face value is the use of a joint reward in cooperative systems that use centralization. The assumption is that joint rewards are necessary for the coordination of systems that rely on centralization. We

have not been able to find a theoretical basis for this claim. The closest works addressing team rewards in cooperative settings that we could find include works on difference rewards which try to measure the impact of an individual agent's actions on the full system reward [12]. The high learnability, among other nice properties, makes difference rewards attractive but impractical, due to the required knowledge of the total system state [13–15].

We investigate the effects of changing the reward from a joint reward to an individual reward in the Level-Based Foraging (LBF) environment. We investigate how different CLDE algorithm performances change as a result of this change and discuss this performance change. In this work, we study the effect of varying reward functions from joint rewards to individual rewards on Independent Q Learning (IQL) [16], Independent Proximal Policy Optimization (IPPO) [17], independent synchronous actor-critic (IA2C) [6], multi-agent proximal policy optimization (MAPPO) [7], multi agent synchronous actor-critic (MAA2C) [5,6], value decomposition networks (VDN) [10], and QMIX [9] when evaluated on the LBF environment [18]. This environment was chosen as it is a gridworld environment, and therefore simpler to understand when compared to other MARL environments such as those based on the StarCraft environment; however, it is a very challenging environment that requires cooperation to solve and has the ability to include the forcing of cooperative policies and partial observability for study.

We show empirically that using an individual reward in the LBF environment causes an increase in the variance in the reward term in the Temporal Difference (TD) error signal and any derivative of this term. We study the effects that this increase in variance has on the selected algorithms and discuss whether this variance is helpful for the learning of better joint policies in the LBF environment. Our results show that PPO-based algorithms, with and without centralized systems and QMIX, perform better with individual rewards, while actor-critic models based on A2C suffer when using individual rewards.

This work is comprised of multiple sections, starting with the background in Section 2. Section 3 outlines our experimental method, and we report our results in Section 4. We discuss the results and compare them to the previous results in Section 5. All supplementary information pertaining to this work can be found in the Appendices A–C.

2. Background

2.1. Dec-POMDPs

We define a fully cooperative task as a decentralized partially observable Markov decision process (Dec-POMDP) which consists of the tuple $M = \langle D, S, A, T, O, o, R, h, b_0 \rangle$ [4]. Where D is the set of agents, S is the set that describes the true state of the environment, A is the joint action set over all agents, and T is the transition probability function, mapping the joint actions to state. O is the joint observation set, o represents the observation probability function, and R is the reward function which describes the set of all individual rewards for each agent $R = R_i^j$. The problem horizon, h , is equivalent to the discount factor γ in the RL literature. The initial state distribution is given by b_0 . M describes a partially observable scenario in which agents interact with the environment through observations, without ever knowing the true state of the environment. When agents have full access to the state information, the tuple becomes $\langle D, S, A, T, R, h, b_0 \rangle$ and is defined as *Multi-agent Markov Decision Process (MMDP)* [4].

2.2. Reward Functions

2.2.1. Joint Reward

The entire team receives a joint reward value at each time step taken as the sum of all individual agent rewards $R = R^i = \dots = R^N = \sum_{i=1}^N R_i^i$. The joint reward has an interesting property that is usually left aside: by being the summation of all agents' rewards, if an agent is not participating in a reward event, they still receive a reward. This creates a small but nonzero probability for all agents to receive a reward in any state and for any action. In addition, in partially observable tasks, these reward events can occur with no context for some of the agents. The advantage of the joint reward is a salient signal

across all that can be learned from, as well as additional information about the performance of team members that may or may not be observable.

2.2.2. Individual Reward

Mixed tasks differ from the fully cooperative case only in terms of the reward received by the agents. Mixed tasks attribute individual rewards to each agent rather than a joint reward, making the term R in the tuple $M, R = R_i^i$ for each agent i . During reward events, a reward is only given to agents who participate in reward events. This reduces the saliency of the reward signal during a reward event, and can cause increased variance in the reward signal when different agents achieve a reward.

2.3. Level-Based Foraging

Level-Based Foraging (LBF) is a challenging exploration problem in which multiple agents must work together to collect food items scattered randomly on a gridworld [18]. The environment is highly configurable, allowing for partial observability and the use of cooperative policies only. In LBF, agents and food are assigned random levels, with the maximum food level always being the sum of all agent levels. Agents can take discrete actions, such as moving in a certain direction, loading food, or not taking any action. Agents receive rewards when they successfully load a food item, which is possible only if the sum of all agent levels around the food is equal to or greater than the level of the food item. Agent observations are discrete and include the location and level of all food and agents on the board, including themselves.

The LBF environment is highly configurable, starting with gridworld size, number of agents, and number of food items. The scenarios in the LBF are described using the following nomenclature: $N \times M - A p - B f$, where N and M define the size of the gridworld, A indicates the number of agents, and B indicates the number of food objectives in the world. A 10 by 10 grid world with three agents and three food would be described as $10 \times 10 - 3 p - 3 f$. Additionally, partial observability can be configured by adding C_s - before the grid size. C defines the radius size that agents can observe. For all objects outside the radius, the agent will receive a constant value of -1 in that observation. Finally, the addition of the $-coop$ tag after the number of food causes the game to enforce that all agents must be present to collect food, thereby forcing cooperative policies to be the only policies that can be learned. As an example, an eight-by-eight gridworld with two players and two food that forces cooperative policies while subjecting the agents to partial observability with a radius of two would be specified as $2s-8x8-2p-2f-coop$. An example of the LBF gridworld is shown in Figure 1.

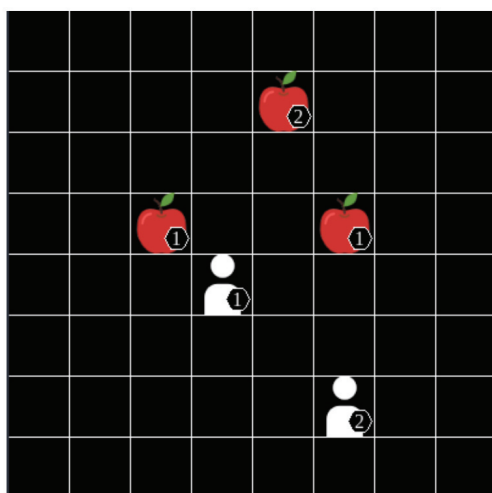


Figure 1. LBF Foraging-8x8-2p-3f example gridworld taken from Papoudakis et al. [5]

3. Method

To compare our results with those of previous publications, we made sure that the scenarios and scenario parameters matched those of Papoudakis et al. [5] and Atrazhev et al. [19], and the results were compared to the results of those previous works.

To remain consistent with previous publications, the LBF scenarios selected for this study are *8x8-2p-2f-coop*, *2s-8x8-2p-2f-coop*, *10x10-3p-3f*, and *2s-10x10-3p-3f*. Algorithms are also selected based on these criteria: IQL [16], IA2C [6], IPPO [17], MAA2C [5], MAPPO [7], VDN [10] and QMIX [9] were selected as they are studied in both Papoudakis et al. [5] and in Atrazhev et al. [19] and represent an acceptable assortment of independent algorithms, centralized critic CLDE algorithms, and value factorization CLDE algorithms.

To evaluate the performance of the algorithm, we calculate the average returns and maximum returns achieved throughout all evaluation windows during training, and the 95% confidence interval across ten seeds.

Our investigation consists of varying two variables, the reward function, and episode length. The length of the episode was varied between the reported value of 25 used by Papoudakis et al. [5] and 50, which is the default length of the episode in the environment. We perform two separate hyperparameter tunings, one for each reward type, adhering to the hyperparameter tuning protocol included in Papoudakis et al. [5].

All other experimental parameters are taken from Papoudakis et al. [5], and we encourage readers to look into this work for further details.

4. Results

We compare IQL, IA2C, IPPO, MAA2C, MAPPO, VDN, and QMIX and report the mean returns and max returns achieved by algorithms using individual rewards in Tables 1 and 2, respectively. The mean returns and maximum returns of algorithms using joint rewards are reported in Tables 3 and 4, respectively. We include tables for the increased episode length (50 timesteps) in the Appendix C.

Table 1. Maximum returns and 95% confidence interval of algorithms using individual rewards in selected scenarios over 10 seeds, after a hyperparameter search was completed. Bolded values indicate the best result in a scenario.

Scenario	IQL	IA2C	IPPO	MAA2C	MAPPO	VDN	QMIX
8x8-2p-2f-c	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.11	1.00 ± 0.00
8x8-2p-2f-2s-c	0.97 ± 0.0	0.94 ± 0.01	0.95 ± 0.01	0.93 ± 0.01	0.93 ± 0.01	0.97 ± 0.0	0.98 ± 0.00
10x10-3p-3f	0.94 ± 0.02	0.86 ± 0.01	0.88 ± 0.04	0.85 ± 0.03	0.86 ± 0.02	0.93 ± 0.04	0.95 ± 0.02
10x10-3p-3f-2s	0.75 ± 0.01	0.71 ± 0.02	0.76 ± 0.02	0.7 ± 0.02	0.73 ± 0.07	0.74 ± 0.01	0.77 ± 0.01

Table 2. Mean return values and 95% confidence interval of algorithms using individual rewards in selected scenarios over 10 seeds, after a hyperparameter search was completed. Bolded values indicate the best result in a scenario.

Scenario	IQL	IA2C	IPPO	MAA2C	MAPPO	VDN	QMIX
8x8-2p-2f-c	0.78 ± 0.08	0.82 ± 0.02	0.84 ± 0.07	0.78 ± 0.05	0.77 ± 0.07	0.7 ± 0.08	0.75 ± 0.04
8x8-2p-2f-2s-c	0.83 ± 0.01	0.71 ± 0.01	0.77 ± 0.01	0.68 ± 0.01	0.69 ± 0.03	0.81 ± 0.01	0.86 ± 0.01
10x10-3p-3f	0.68 ± 0.02	0.7 ± 0.02	0.72 ± 0.03	0.66 ± 0.03	0.69 ± 0.02	0.55 ± 0.04	0.58 ± 0.03
10x10-3p-3f-2s	0.62 ± 0.0	0.55 ± 0.02	0.58 ± 0.02	0.51 ± 0.02	0.53 ± 0.06	0.57 ± 0.01	0.62 ± 0.01

Table 3. Maximum returns and 95% confidence interval of algorithms using joint rewards in selected scenarios over 10 seeds, after a hyperparameter search was completed. Bolded values indicate the best result in a scenario.

Scenario	IQL	IA2C	IPPO	MAA2C	MAPPO	VDN	QMIX
8x8-2p-2f-c	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
8x8-2p-2f-2s-c	0.97 ± 0.01	1.0 ± 0.0	0.63 ± 0.02	1.0 ± 0.0	0.56 ± 0.02	0.98 ± 0.0	0.97 ± 0.0
10x10-3p-3f	0.89 ± 0.08	0.99 ± 0.01	0.89 ± 0.02	0.98 ± 0.01	0.9 ± 0.24	0.9 ± 0.03	0.91 ± 0.02
10x10-3p-3f-2s	0.7 ± 0.01	0.84 ± 0.04	0.56 ± 0.01	0.85 ± 0.01	0.58 ± 0.01	0.77 ± 0.01	0.76 ± 0.04

Table 4. Mean return values and 95% confidence interval of algorithms using joint rewards in selected scenarios over 10 seeds, after a hyperparameter search was completed. Bolded values indicate the best result in a scenario.

Scenario	IQL	IA2C	IPPO	MAA2C	MAPPO	VDN	QMIX
8x8-2p-2f-c	0.77 ± 0.08	0.96 ± 0.01	0.96 ± 0.01	0.97 ± 0.01	0.96 ± 0.02	0.78 ± 0.04	0.69 ± 0.04
8x8-2p-2f-2s-c	0.82 ± 0.01	0.94 ± 0.01	0.39 ± 0.02	0.94 ± 0.0	0.45 ± 0.02	0.84 ± 0.0	0.79 ± 0.01
10x10-3p-3f	0.47 ± 0.07	0.88 ± 0.02	0.71 ± 0.03	0.87 ± 0.02	0.59 ± 0.21	0.56 ± 0.03	0.46 ± 0.04
10x10-3p-3f-2s	0.56 ± 0.01	0.67 ± 0.05	0.44 ± 0.0	0.69 ± 0.02	0.46 ± 0.0	0.6 ± 0.01	0.56 ± 0.05

We generally observe that in the individual reward case, QMIX is able to consistently achieve the highest maximal return value in all scenarios. In terms of the highest mean returns, QMIX is able to outperform IPPO in the partially observable scenarios. In the joint reward case, the majority of the results are in line with those reported in [5]; however, we note that the average return results for QMIX are much higher with our hyperparameters. We go into more detail regarding these results in Appendix A.

When comparing joint reward performance with individual reward performance, we note that the effects of reward are not easily predictable. Centralized critic algorithms are evenly split in performance, with MAPPO performing better with individual reward, while MAA2C's performance suffers. This is paralleled by the independent versions of MAPPO and MAA2C. The value factorization algorithms are also divided, with QMIX performance becoming the top-performing algorithm across the tested scenarios. VDN, however, sees an incredible drop in performance when using joint rewards. Finally, IQL performance when using individual reward is relatively unaffected in the simpler 8x8 scenarios but decreases in the larger scenarios.

5. Discussion

5.1. Independent Algorithms

5.1.1. IQL

Our results show that IQL achieves increased mean return values and maximum return values when using individual rewards. Our results also show that IQL experienced a reduction in loss variance when using individual rewards. Since IQL is an independent algorithm, the joint reward is the only source of information from other agents. Seeing that IQL does not observe the other agents specifically, our results suggest that the joint reward seems to increase the variance in the loss function by the nonzero probability of agents receiving the reward at any timestep, as discussed earlier. The reduction in variance in the loss function allows for better policies to be learned by each individual agent, and this is further evidenced by the reduction in variance and simultaneous increase in the mean of the absolute TD error that agents have in the CLBF experiments.

5.1.2. IPPO

IPPO is able to use the individual reward signal to achieve higher mean returns and maximum returns in all scenarios except for the 8x8-2p-2f-coop. We believe that this is in large part due to the decrease in variance that is observed in the maximum policy values that are learned. Our results show that the TD error that is generated from multiple different individual rewards appears to be higher and more varied than the TD error that is generated from a joint reward. This variance seems to permeate through the loss function, allowing the algorithm to continue discovering new higher policies through training. It seems that joint rewards cause the TD error to start out strong, and quickly the algorithm finds a policy (or set of policies) that has the maximal chances of achieving rewards at all timesteps. This is a local minimum, but the error is too small for policies to escape the minima.

5.1.3. IA2C

IA2C suffers from the increase in variance in individual rewards. We note evidence of divergent policy behaviour in a number of metrics, most notably the critic and policy gradient loss. The critic is still able to converge; however, the policy gradient loss diverges quite a bit more in the individual reward case. It seems that a joint reward is necessary to help coordinate the agent's behaviour.

5.2. Value Factorization Algorithms

5.2.1. VDN

VDN with individual rewards has a very rapid reduction in loss values. Our data suggest that when using individual rewards, VDN converges prematurely on suboptimal policies, causing the observed reduction in mean and max return. This may be due to the fact that VDN does not incorporate any state information into the creation of the joint value function. The authors seem to have relied on the information contained in the joint rewards to help guide the coordination of agents through the learned joint value function. With individual rewards, the joint action value function simply optimizes for the first policy that serves to maximize returns without regard for agent coordination or guiding agents to find optimal policies.

5.2.2. QMIX

Our results show that when individual rewards are used with qmix, return mean and maximum return values are increased. When comparing joint rewards to independent rewards, independent rewards show signs of faster convergence in loss and gradient norms. Qmix's combination of monotonicity constraints and global state information in its hyper-network seems to be able to find coordinated policies when using individual rewards that achieve higher returns than those found when using joint rewards. By leveraging the global state information during training, the improvement shows significantly higher in the partially observable scenarios where the increased information builds stronger coordination between agents.

5.3. Centralized Critic Algorithms

Performance in centralized critics is varied and seems to depend on the underlying algorithm used.

5.3.1. MAA2C

The increase in information that is imparted by MAA2C's centralized critic seems to not be enough to counter the increase in variance that is caused by individual rewards. When using joint rewards, the critic is able to converge and is able to guide the actor policies to find optimal values relatively quickly, and is best demonstrated by the convergence of the TD error. When using individual rewards, there seems to be too much variance for the critic to be able to converge quickly. It has been shown that simply adding a centralized critic to an actor-critic MARL algorithm with the hopes of decreasing variance in the agent learning is not necessarily true and will actually increase the variance seen by actors [11]. It seems that in MAA2C, using the joint reward to decrease the variance seen by the critic is a good way to increase performance. We do, however, note that when we increased the episode length, the individual reward mean and max returns continued to increase; however, they do not show any evidence of rapid convergence. It seems that more research is required on the effects of increasing the episode length to determine if the joint reward has a bias component.

5.3.2. MAPPO

Similarly to IPPO, MAPPO performs better when using individual rewards than when using joint rewards. MAPPO's centralized critic does not seem to be able to prevent the critic from converging prematurely. Centralized critics have been shown to increase

variance [11]; however, our results show that the increase in variance in the critic loss is not enough. Just as in IPPO, the critic converges within 100 episodes when using joint rewards. This corresponds to the majority of the gains in return, which seems to indicate that some local minima are found by the algorithm.

6. Conclusions and Future Work

In summary, our results show that different CLDE algorithms respond in different ways when the reward is changed from joint to individual in the LBF environment. MAPPO and QMIX show that they are able to leverage the additional variance present in the individual reward to find improved policies, while VDN and MAA2C suffer from the increase and perform worse. Of the centralized critic algorithms, it seems that it is crucial that the centralized algorithm critic be able to converge slowly enough to find the optimal joint policy, but not fast enough to find a local minima. In addition, if the critic is too sensitive to the increase in variance, it may diverge as in MAA2C and be unable to find the optimal policy. Value decomposition methods also seem to need additional state information to condition the coordination of agents to learn optimal policies. Since much of the emergent behaviour sought in MARL systems is a function of how agents work together, we feel that the choice of reward function may be of even more importance in MARL environments than in a single-agent environment. Our results hint that there may be some greater bias variance-type trade-off between joint and individual rewards; however, more research will need to be performed to confirm this.

As we have outlined in several sections of this work, there are still many questions that need answering before we can definitively say that the choice of using a joint reward or an individual reward when training MARL algorithms comes down to a bias variance trade-off. First, this theory of increased variance would need to be studied in simpler scenarios that can be solved analytically in order to confirm that individual rewards do increase variance. This simpler scenario would need to have the same sparse positive reward as seen in the LBF. Following the establishment of this theoretical underpinning, the next step would be to either relax the sparse constraint or the positive reward constraint and still see if the theory holds true. Once that is performed, a definitive conclusion could be presented about the effects of varying reward functions between joint and individual rewards in cooperative MARL systems.

Author Contributions: Conceptualization, P.A.; methodology, P.A. and P.M.; investigation, P.A.; software, P.A.; resources, P.M.; writing—original draft preparation, P.A.; writing—review and editing, P.M.; supervision, P.M.; funding acquisition, P.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data for the reproduction of the figures as well as validation of this research can be found at: <https://github.com/at-peter/System-all-about-rewards-data>.

Acknowledgments: The authors gratefully acknowledge the indirect support provided by the Mitacs Accelerate Entrepreneur program and by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Hyperparameter Optimization

CLBF Hyperparameter Optimisation

The appendix of [5] contains the hyperparameter search protocol that they used in order to perform their hyperparameter search. In order to keep the comparison to [5], we propose following the same hyperparameter search protocol, which is outlined in Table A1.

Table A1. Hyperparameter search protocol taken from [5].

Hyperparameters	Values
Hidden dimension	64/128
Learning rate	0.0001/0.0003/0.0005
Reward Standardization	True/False
Network Type	FC/GRU
Evaluation Epsilon	0.0/0.05
Epsilon Anneal	50,000/200,000
Target Update	200 (hard)/0.01 (soft)
Entropy Coefficient	0.01/0.001
n-step	5/10

The hyperparameter search was performed as follows. A search with three seeds was performed on the 10x10-3p-3f scenario to narrow down a short list of candidate hyperparameter configurations. Priority was given to hyperparameter sets that repeat.

Table A2 Shows the difference between previously tested hyperparameters and the hyperparameters that were discovered during the hyperparameter search on the CLBF environment.

Table A2. IPPO selected hyperparameters.

Hyperparameters	Papoudakis et al. [5]	Our Hyperparameter Search
Hidden dimension	128	64
Learning rate	0.0003	0.0003
Reward Standardization	False	True
Network Type	FC	GRU
Target Update	200 (hard)	200 (hard)
Entropy Coefficient	0.001	0.01
n-step	5	10

Appendix B. Validation of Papoudakis et al. [5] Results

As part of our work on the analysis of algorithmic performance, we replicated the work that was performed as part of [5] on the LBF environment. This section includes the data that were collected from our repeated experiments. We used the hyperparameters that were reported in the appendix section of [5] and ran 10 runs for each hyperparameter configuration. The selected hyperparameters were those for parameter sharing, and parameter sharing was used for the data collection to keep in line with the results in [5].

We found discrepancies between the reported data in [5] for VDN and QMIX, and these discrepancies also seem to explain some of the results we reported in [19]. Notably, we found that the convergence of the value factorization methods was not reported properly in [5], and these convergence values are in line with the increase in convergence rates that we found in [19].

Table A3. Maximum returns and 95% confidence interval of hyperparameter configurations taken from [5]. Bolded values are those that differ significantly from [5].

Tasks/Algs	IQL	IA2C	IPPO	MAA2C	MAPPO	VDN	QMIX
8x8-2p-2f-c	1.0 ± 0.0	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.00	1.0 ± 0.0	1.0 ± 0.00	1.0 ± 0.00
8x8-2p-2f-2s-c	0.97 ± 0.01	1.0 ± 0.0	0.63 ± 0.02	1.0 ± 0.0	0.56 ± 0.02	0.98 ± 0.00	0.97 ± 0.0
10x10-3p-3f	0.89 ± 0.08	0.99 ± 0.01	0.89 ± 0.02	0.98 ± 0.01	0.9 ± 0.24	0.9 ± 0.03	0.91 ± 0.02
10x10-3p-3f-2s	0.7 ± 0.01	0.84 ± 0.04	0.56 ± 0.01	0.85 ± 0.01	0.58 ± 0.01	0.77 ± 0.01	0.76 ± 0.04

Table A4. Average returns and 95% confidence interval of hyperparameter configurations taken from [5]. Bolded values are those that differ significantly from [5].

Tasks/Algs	IQL	IA2C	IPPO	MAA2C	MAPPO	VDN	QMIX
8x8-2p-2f-c	0.77 ± 0.08	0.96 ± 0.01	0.96 ± 0.01	0.97 ± 0.01	0.96 ± 0.02	0.78 ± 0.04	0.69 ± 0.04
8x8-2p-2f-2s-c	0.82 ± 0.01	0.94 ± 0.01	0.39 ± 0.02	0.94 ± 0.0	0.45 ± 0.02	0.84 ± 0.0	0.79 ± 0.01
10x10-3p-3f	0.47 ± 0.07	0.88 ± 0.02	0.71 ± 0.03	0.87 ± 0.02	0.59 ± 0.21	0.56 ± 0.03	0.46 ± 0.04
10x10-3p-3f-2s	0.56 ± 0.01	0.67 ± 0.05	0.44 ± 0.0	0.69 ± 0.02	0.46 ± 0.0	0.6 ± 0.01	0.56 ± 0.05

Appendix C. Variance Analysis Data

This section of the appendix contains all the statistical data analysis that was used during the empirical variance analysis in Section 4. The statistical analysis used Bartlett's test in order to determine if the variance in two means is the same. The α value used to determine statistical significance is $\alpha = 0.05$. Bartlett's test tests the null hypothesis h_0 that the variances of each data distribution tested are identical. If the p -value is below that of the selected α , then the null hypothesis is rejected, and the variances of the data tested are not the same. In our analysis, the data collected for each run were averaged over, and then the set of 10 replicates was used in Bartlett's test. The nan value indicates that there was no variation at all because the algorithm was able to solve the scenario perfectly in the 25 timestep scenarios for both individual and joint rewards.

Appendix C.1. IQL

Below are the statistics that were gathered on the IQL algorithm. The result aspects of the algorithm that were compared include the following: loss, grad norm, mean of selected q values, means of return, max of returns, and target network mean q values for the selected action. Variances are evaluated between joint reward and independent reward. Bolded p -values reject the null hypothesis, indicating that the variances between the 25-step and 50-step runs are different.

Table A5. p -values of Bartlett's test for homogeneity of variances for gradient norm values of IQL between 25 timesteps and 50 timesteps grouped by scenario.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.88	0.15	0.048	0.35
50	0.066	0.63	0.18	0.044

Table A6. p -values of Bartlett's test for homogeneity of variances for loss values of IQL between 25 timesteps and 50 timesteps.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.014	0.095	1.45×10^{-3}	0.71
50	0.21	0.069	6.42×10^{-4}	0.67

Table A7. p -values of Bartlett's test for homogeneity of variances for the mean q value of selected actions of IQL between 25 timesteps and 50 timesteps grouped by scenario.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.68	0.50	0.074	0.002
50	0.56	0.49	0.059	0.64

Table A8. p -values of Bartlett's test for homogeneity of variances for the target value of selected actions of IQL between 25 timesteps and 50 timesteps grouped by scenario.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.70	0.47	0.080	2.08×10^{-3}
50	0.56	0.47	0.061	0.63

Table A9. *p*-values of Bartlett’s test for homogeneity of variances for the mean return values of IQL between 25 timesteps and 50 timesteps grouped by scenario.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.99	0.36	2.80×10^{-3}	7.73×10^{-3}
50	0.73	0.48	0.023	0.57

Table A10. *p*-values of Bartlett’s test for homogeneity of variances for the max return values of IQL between 25 timesteps and 50 timesteps grouped by scenario.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	1.84×10^{-6}	0.77	5.28×10^{-5}	0.41
50	nan	9.22×10^{-3}	1.60×10^{-3}	0.47

Appendix C.2. IPPO

Below are the statistics that were gathered on the IPPO algorithm. The statistics that were tested include the following: mean return, max return, agent grad norms, critic grad norms, critic loss, policy gradient loss, maximum Pi values of the actor, and advantage means. Variances are evaluated between joint reward and independent reward. Bolded *p*-values reject the null hypothesis, indicating that the variances between the 25-step and 50-step runs are different.

Table A11. *p*-values of Bartlett’s test for homogeneity of variance for mean returns of IPPO varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	4.52×10^{-5}	0.074	0.54	1.14×10^{-5}
50	0.16	3.39×10^{-6}	0.75	1.37×10^{-4}

Table A12. *p*-values of Bartlett’s test for homogeneity of variance for max returns of IPPO varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.0	0.013	0.049	9.04×10^{-4}
50	0.0	8.66×10^{-7}	8.26×10^{-6}	0.34

Table A13. *p*-values of Bartlett’s test for homogeneity of variance for agent grad norms returns of IPPO varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	8.27×10^{-17}	9.97×10^{-4}	1.17×10^{-12}	3.23×10^{-14}
50	9.14×10^{-16}	5.11×10^{-8}	8.82×10^{-13}	5.99×10^{-18}

Table A14. *p*-values of Bartlett’s test for homogeneity of variance for critic grad norms returns of IPPO varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	1.98×10^{-21}	1.15×10^{-13}	2.98×10^{-22}	2.27×10^{-15}
50	4.39×10^{-23}	2.60×10^{-15}	9.49×10^{-20}	5.75×10^{-18}

Table A15. *p*-values of Bartlett’s test for homogeneity of variance for critic loss of IPPO varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	1.51×10^{-25}	2.56×10^{-10}	2.33×10^{-22}	2.43×10^{-17}
50	1.69×10^{-28}	1.79×10^{-11}	1.01×10^{-18}	2.90×10^{-18}

Table A16. *p*-values of Bartlett’s test for homogeneity of variance for policy gradient loss of IPPO varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	1.86×10^{-17}	2.06×10^{-9}	5.25×10^{-14}	4.11×10^{-12}
50	1.22×10^{-22}	9.84×10^{-17}	7.55×10^{-14}	4.60×10^{-14}

Table A17. *p*-values of Bartlett’s test for homogeneity of variance for maximum policy values of IPPO varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	1.39×10^{-5}	1.28×10^{-3}	0.19	0.18
50	0.64	6.63×10^{-4}	1.88×10^{-4}	0.011

Table A18. *p*-values of Bartlett’s test for homogeneity of variance for advantage means of IPPO varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	2.05×10^{-17}	3.98×10^{-10}	9.86×10^{-15}	2.59×10^{-12}
50	1.12×10^{-21}	1.38×10^{-16}	5.35×10^{-14}	8.63×10^{-14}

Appendix C.3. IA2C

Below are the statistics that were gathered on the IA2C algorithm. The statistics that were tested include the following: mean return, max return, agent grad norms, critic grad norms, critic loss, policy gradient loss, maximum Pi values of the actor, and advantage means. Variances are evaluated between joint reward and independent reward. Bolded *p*-values reject the null hypothesis, indicating that the variances between the 25-step and 50-step runs are different.

Table A19. *p*-values of Bartlett’s test for homogeneity of variances for mean return values of IA2C between 25 timesteps and 50 timesteps.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.016	0.35	0.63	6.69×10^{-3}
50	0.003	0.38	0.12	0.063

Table A20. *p*-values of Bartlett’s test for homogeneity of variances for max return values of IA2C between 25 timesteps and 50 timesteps.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.0	4.27×10^{-4}	0.071	0.29
50	0.0	0.0	2.07×10^{-9}	0.016

Table A21. *p*-values of Bartlett’s test for homogeneity of variances for critic grad norm of IA2C between 25 timesteps and 50 timesteps.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.25	0.24	0.33	0.005
50	0.13	0.31	0.019	0.010

Table A22. *p*-values of Bartlett’s test for homogeneity of variances for critic loss of IA2C between 25 timesteps and 50 timesteps.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.60	0.19	0.011	4.12×10^{-4}
50	0.81	0.33	0.045	0.17

Table A23. *p*-values of Bartlett’s test for homogeneity of variances for PG loss of IA2C between 25 timesteps and 50 timesteps.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.25	0.24	0.33	4.89×10^{-3}
50	0.13	0.31	0.019	9.87×10^{-3}

Table A24. *p*-values of Bartlett’s test for homogeneity of variances for advantage mean of IA2C between 25 timesteps and 50 timesteps.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.20	0.17	3.13×10^{-3}	0.033
50	0.029	0.13	0.15	3.15×10^{-3}

Appendix C.4. VDN

Below are the statistics that were gathered on the VDN algorithm. The results aspects of the algorithm that were compared include the following: loss, grad norm, mean of selected q values, means of return, max of returns, and target network mean q values for the selected action. Variances are evaluated between joint reward and independent reward. Bolded *p*-values reject the null hypothesis, indicating that the variances between the 25-step and 50-step runs are different.

Table A25. *p*-values of Bartlett’s test for homogeneity of variances for gradient norm values of VDN between 25 timesteps and 50 timesteps grouped by scenario.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.30	0.41	1.00	0.13
50	0.45	0.011	0.55	0.005

Table A26. *p*-values of Bartlett’s test for homogeneity of variances for loss values of VDN between 25 timesteps and 50 timesteps.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.17	0.40	0.016	0.10
50	0.87	0.58	0.33	0.076

Table A27. *p*-values of Bartlett’s test for homogeneity of variances for the mean q value of selected actions of VDN between 25 timesteps and 50 timesteps grouped by scenario.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.034	0.099	0.77	0.052
50	0.021	0.87	0.83	0.20

Table A28. *p*-values of Bartlett’s test for homogeneity of variances for the target network mean q values of selected actions of VDN between 25 timesteps and 50 timesteps grouped by scenario.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.034	0.11	0.78	0.038
50	0.020	0.86	0.81	0.18

Table A29. *p*-values of Bartlett’s test for homogeneity of variances for the mean return values VDN between 25 timesteps and 50 timesteps grouped by scenario.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.038	0.002	0.27	0.11
50	0.36	0.75	0.71	0.37

Table A30. *p*-values of Bartlett’s test for homogeneity of variances for the max return values VDN between 25 timesteps and 50 timesteps grouped by scenario.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.0	0.50	0.31	0.089
50	0.0	0.26	0.26	0.003

Appendix C.5. QMIX

Below are the statistics that were gathered on the QMIX algorithm. The results aspects of the algorithm that were compared include the following: loss, grad norm, mean of selected q values, means of return, max of returns, and target network mean q values for the selected action. Variances are evaluated between joint reward and independent reward. Bolded *p*-values reject the null hypothesis, indicating that the variances between the 25-step and 50-step runs are different.

Table A31. *p*-values of Bartlett’s test for homogeneity of variances for loss values of Qmix between 25 timesteps and 50 timesteps.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	8.18×10^{-6}	0.13	7.12×10^{-5}	5.66×10^{-8}
50	9.17×10^{-32}	1.84×10^{-10}	0.25	3.55×10^{-4}

Table A32. *p*-values of Bartlett’s test for homogeneity of variances for gradient norm values of Qmix between 25 timesteps and 50 timesteps grouped by scenario.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	2.92×10^{-8}	0.20	6.36×10^{-7}	7.19×10^{-10}
50	1.09×10^{-17}	1.53×10^{-11}	1.06×10^{-3}	9.73×10^{-7}

Table A33. *p*-values of Bartlett’s test for homogeneity of variances for the mean q value of selected actions of Qmix between 25 timesteps and 50 timesteps grouped by scenario.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.04	0.55	0.01	1.65×10^{-8}
50	0.04	2.86×10^{-8}	0.03	0.08

Table A34. *p*-values of Bartlett’s test for homogeneity of variances for the target network mean q values of selected actions of Qmix between 25 timesteps and 50 timesteps grouped by scenario.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.04	0.52	6.18×10^{-3}	1.21×10^{-8}
50	0.03	2.66×10^{-8}	0.03	0.07

Table A35. *p*-values of Bartlett’s test for homogeneity of variances for the max return values Qmix between 25 timesteps and 50 timesteps grouped by scenario.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	<i>Nan</i>	0.90	0.76	2.73×10^{-4}
50	4.65×10^{-6}	0.83	3.90×10^{-4}	0.21

Table A36. *p*-values of Bartlett’s test for homogeneity of variances for the mean return values Qmix between 25 timesteps and 50 timesteps grouped by scenario.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.87	0.61	0.40	1.24×10^{-5}
50	7.55×10^{-3}	2.10×10^{-2}	0.51	0.25

Appendix C.6. MAA2C

Below are the statistics that were gathered on the MAA2C algorithm. The statistics that were tested include the following: mean return, max return, agent grad norms, critic grad norms, critic loss, policy gradient loss, maximum Pi values of the actor, and advantage means. Bolded p -values reject the null hypothesis, indicating that the variances between the 25-step and 50-step runs are different.

Table A37. p -values of Bartlett's test for homogeneity of variance for mean returns of MAA2C varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	1.46×10^{-7}	9.93×10^{-5}	0.12	0.22
50	9.03×10^{-8}	7.12×10^{-12}	1.59×10^{-7}	0.89

Table A38. p -values of Bartlett's test for homogeneity of variance for Max Returns of MAA2C varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.0	9.83×10^{-10}	1.09×10^{-4}	0.011
50	0.0	0.0	9.83×10^{-2}	0.28

Table A39. p -values of Bartlett's test for homogeneity of variance for agent grad norms of MAA2C varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.90	7.12×10^{-2}	0.43	4.62×10^{-3}
50	1.76×10^{-2}	0.50	0.80	4.54×10^{-2}

Table A40. p -values of Bartlett's test for homogeneity of variance for critic grad norms of MAA2C varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.33	0.020	2.81×10^{-5}	0.74
50	0.13	0.005	0.13	0.012

Table A41. p -values of Bartlett's test for homogeneity of variance for critic loss of MAA2C varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	0.027	2.52×10^{-4}	0.004	0.69
50	0.029	0.11	1.20×10^{-5}	0.59

Table A42. p -values of Bartlett's test for homogeneity of variance for pg loss of MAA2C varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	3.20×10^{-5}	1.32×10^{-5}	2.22×10^{-3}	0.029
50	0.034	1.17×10^{-6}	0.014	0.025

Table A43. p -values of Bartlett's test for homogeneity of variance for max policy values of MAA2C varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	3.28×10^{-7}	6.16×10^{-4}	0.18	0.019
50	3.31×10^{-4}	1.64×10^{-3}	1.94×10^{-8}	0.061

Table A44. *p*-values of Bartlett’s test for homogeneity of variance for advantage mean values of MAA2C varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	1.51×10^{-3}	0.025	1.11×10^{-3}	0.071
50	0.38	3.53×10^{-4}	0.90	1.07×10^{-4}

Appendix C.7. MAPPO

Below are the statistics that were gathered on the MAPPO algorithm. The statistics that were tested include the following: mean return, max return, agent grad norms, critic grad norms, critic loss, policy gradient loss, maximum Pi values of the actor, and advantage means. Bolded *p*-values reject the null hypothesis, indicating that the variances between the 25-step and 50-step runs are different.

Table A45. *p*-values of Bartlett’s test for homogeneity of variance for return means of MAPPO varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	3.21×10^{-4}	0.12	1.08×10^{-6}	2.85×10^{-6}
50	2.76×10^{-6}	0.84	0.92	4.05×10^{-6}

Table A46. *p*-values of Bartlett’s test for homogeneity of variance for return maxes of MAPPO varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	3.28×10^{-5}	0.90	3.71×10^{-8}	3.70×10^{-4}
50	0.00	1.86×10^{-5}	1.66×10^{-6}	1.74×10^{-3}

Table A47. *p*-values of Bartlett’s test for homogeneity of variance for agent grad norms of MAPPO varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	1.73×10^{-12}	1.85×10^{-5}	2.49×10^{-10}	3.24×10^{-9}
50	6.17×10^{-16}	2.60×10^{-8}	3.29×10^{-13}	4.52×10^{-18}

Table A48. *p*-values of Bartlett’s test for homogeneity of variance for critic grad norm of MAPPO varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	1.29×10^{-17}	2.35×10^{-11}	6.32×10^{-11}	4.66×10^{-12}
50	1.53×10^{-20}	1.51×10^{-19}	9.26×10^{-22}	8.30×10^{-20}

Table A49. *p*-values of Bartlett’s test for homogeneity of variance for policy gradient loss of MAPPO varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	3.46×10^{-18}	6.95×10^{-7}	1.15×10^{-7}	6.13×10^{-8}
50	3.70×10^{-22}	1.11×10^{-17}	1.14×10^{-14}	3.87×10^{-16}

Table A50. *p*-values of Bartlett’s test for homogeneity of variance for max policy values of MAPPO varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	6.89×10^{-3}	0.83	1.44×10^{-6}	0.22
50	0.015	0.15	0.40	0.10

Table A51. *p*-values of Bartlett’s test for homogeneity of variance for advantage mean values of MAPPO varying episode length between 25 timesteps and 50 timesteps and comparing reward functions.

	Foraging-8x8-2p-2f-coop/	Foraging-2s-8x8-2p-2f-coop/	Foraging-10x10-3p-3f/	Foraging-2s-10x10-3p-3f/
25	2.81×10^{-18}	4.19×10^{-7}	3.13×10^{-7}	5.83×10^{-8}
50	6.95×10^{-22}	1.27×10^{-17}	1.62×10^{-14}	4.02×10^{-16}

References

- Samvelyan, M.; Rashid, T.; de Witt, C.S.; Farquhar, G.; Nardelli, N.; Rudner, T.G.J.; Hung, C.M.; Torr, P.H.S.; Foerster, J.; Whiteson, S. The StarCraft Multi-Agent Challenge. In Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montréal, Canada, 13–17 May 2019.
- Bellemare, M.G.; Naddaf, Y.; Veness, J.; Bowling, M. The arcade learning environment: An evaluation platform for general agents. *J. Artif. Intell. Res.* **2013**, *47*, 253–279. [\[CrossRef\]](#)
- Ellis, B.; Moalla, S.; Samvelyan, M.; Sun, M.; Mahajan, A.; Foerster, J.N.; Whiteson, S. SMACv2: An Improved Benchmark for Cooperative Multi-Agent Reinforcement Learning. *arXiv* **2022**, arXiv:2212.07489.
- Oliehoek, F.A.; Amato, C. The Decentralized POMDP Framework. In *A Concise Introduction to Decentralized POMDPs*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 11–32.
- Papoudakis, G.; Christianos, F.; Schäfer, L.; Albrecht, S.V. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks. In Proceedings of the RLEM’20: Proceedings of the 1st International Workshop on Reinforcement Learning for Energy Management in Buildings & Cities, Virtual, 17 November 2020.
- Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 22 June 2016; Balcan, M.F., Weinberger, K.Q., Eds.; PMLR: New York, NY, USA, 2016; Volume 48, pp. 1928–1937.
- Yu, C.; Velu, A.; Vinitisky, E.; Wang, Y.; Bayen, A.M.; Wu, Y. The Surprising Effectiveness of MAPPO in Cooperative, Multi-Agent Games. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24611–24624.
- Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
- Rashid, T.; Samvelyan, M.; de Witt, C.S.; Farquhar, G.; Foerster, J.N.; Whiteson, S. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018.
- Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W.M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J.Z.; Tuyls, K.; et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv* **2017**, arXiv:1706.05296.
- Lyu, X.; Xiao, Y.; Daley, B.; Amato, C. Contrasting Centralized and Decentralized Critics in Multi-Agent Reinforcement Learning. In Proceedings of the AAMAS ’21: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, Virtual, 3–7 May 2021.
- Tumer, K.; Wolpert, D. A survey of collectives. *Collectives and the Design of Complex Systems*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 1–42.
- Colby, M.; Duchow-Pressley, T.; Chung, J.J.; Tumer, K. Local approximation of difference evaluation functions. In Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, 9–13 May 2016; pp. 521–529.
- Agogino, A.K.; Tumer, K. Analyzing and visualizing multiagent rewards in dynamic and stochastic domains. *Auton. Agents Multi-Agent Syst.* **2008**, *17*, 320–338. [\[CrossRef\]](#)
- Proper, S.; Tumer, K. Modeling difference rewards for multiagent learning. In Proceedings of the AAMAS, Valencia, Spain, 4–8 June 2012; pp. 1397–1398.
- Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [\[CrossRef\]](#) [\[PubMed\]](#)
- Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
- Christianos, F.; Schäfer, L.; Albrecht, S.V. Shared Experience Actor-Critic for Multi-Agent Reinforcement Learning. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020.
- Atrazhev, P.; Musilek, P. Investigating Effects of Centralized Learning Decentralized Execution on Team Coordination in the Level Based Foraging Environment as a Sequential Social Dilemma. In Proceedings of the International Conference on Practical Applications of Agents and Multi-Agent Systems, L’Aquila, Italy, 13–15 July 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 15–23.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

An Infrastructure Cost and Benefits Evaluation Framework for Blockchain-Based Applications

Miguel Pincheira ^{1,*}, Elena Donini ², Massimo Vecchio ¹ and Raffaele Giaffreda ¹¹ Digital Industry Center, Fondazione Bruno Kessler, 38123 Trento, Italy² Digital Society Center, Fondazione Bruno Kessler, 38123 Trento, Italy

* Correspondence: mpincheiracararo@fbk.eu

Abstract: Blockchain is currently a core technology for developing new types of decentralized applications. With the unique properties of blockchain, unique challenges and characteristics are introduced to the system. Among these characteristics, the infrastructure costs and benefits of the system are critical to evaluate the feasibility of any system and have yet to be addressed in the current literature. This work presents a framework for evaluating blockchain applications' infrastructure costs and benefits. The framework includes a taxonomy to classify the related transactions, a model to evaluate the infrastructure costs and benefits in applications using public or private blockchains, and a methodology to guide the use of the model. The model is based on simple parameters that describe the systems, and the methodology helps to identify and estimate these parameters at any stage of the application life cycle. We quantitatively analyze three real use cases to demonstrate the framework's merit. The analyses highlight the model's accuracy by achieving the same results presented in the use cases. Furthermore, the use-case analyses emphasize the framework's potential to evaluate different scenarios across the entire life cycle of blockchain-based applications.

Keywords: blockchain; software; infrastructure; costs; benefits; evaluation

Citation: Pincheira, M.; Donini, E.; Vecchio, M.; Giaffreda, R. An Infrastructure Cost and Benefits Evaluation Framework for Blockchain-Based Applications. *Systems* **2023**, *11*, 184. <https://doi.org/10.3390/systems11040184>

Academic Editors: Philippe Mathieu, Juan M. Corchado, Alfonso González-Briones and Fernando De la Prieta Pintado

Received: 31 January 2023

Revised: 16 March 2023

Accepted: 31 March 2023

Published: 5 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Blockchain is recognized as one of the most important technology disruptions in the latest years [1]. Researchers and practitioners have shown increasing interest in leveraging blockchain technology's unique benefits and properties to empower new software systems [2]. Applications in several domains have adopted private [3] and public blockchain networks [4,5] to provide a software platform where interactions between actors can occur without intermediaries. Thus, from a software perspective, blockchain has been mainly characterized as an architecture component that provides immutable storage and computational capabilities in a decentralized way. The unique properties of this software component also raise new challenges across the systems development life cycle while introducing unexplored characteristics that need to be identified and evaluated [1,6]. Much of the current literature has focused on non-functional characteristics of blockchain-based systems, such as scalability, security, and performance [6,7].

However, blockchain is still in the early stages of adoption [8], and more is needed to know about other characteristics supporting the system development, deployment, and evaluation. Among these characteristics, the infrastructure to support the application is critical to evaluate the potential monetary value in terms of costs and benefits across the entire application life cycle. Costs and benefits can greatly increase the adoption of blockchain on software systems [1], as blockchain technology is a major target of investments for companies [7]. Although there has been some research on evaluating the infrastructure costs of blockchain applications [1,9], these findings are focused on particular use cases and are not extendable to other scenarios. Each application and use case may have unique

requirements and characteristics that must be considered to evaluate its blockchain infrastructure [1]. Therefore, further research is necessary to fully understand blockchain applications' potential costs and benefits across various use cases and contexts.

This work presents an infrastructure cost framework for blockchain-based software systems, extending a first version of the model presented in [10]. We propose a framework to evaluate application costs and benefits from the early stages of development to the exploitation phase, based on the blockchain software component of the system. Our goal is to provide a tool matching the existing literature on the cost and benefits of blockchain-based applications while also providing the tools to analyze more complex cost scenarios.

The framework comprises a transaction taxonomy, a cost and benefits model for public and private networks, and a methodology to apply the model. First, the taxonomy aims to classify and generalize typical transactions in the life cycle of blockchain applications. Then, the infrastructure cost and benefits model aims to create evaluation scenarios over the entire application's life cycle based on simple application parameters. The selection of these parameters aims to simplify the characterization of the system with minimal application domain knowledge. Finally, we propose a methodology for identifying and estimating the model parameters. To illustrate the proposed framework's usability, we quantitatively analyze the monetary costs and benefits of three blockchain-based applications from the current literature. These evaluations highlight the flexibility of the model to work for public and private blockchains, the simplicity of identifying the model parameters using the proposed methodology, and the benefits of the framework to analyze different scenarios for the entire application life cycle.

The contribution of this paper is three-fold. (i) We propose a transaction taxonomy for blockchain-based applications. (ii) We define an infrastructure cost model for private and public blockchains. (iii) We propose a methodology that identifies and estimates the model parameters to create and evaluated scenarios over the application's life cycle.

The rest of this paper is structured as follows: Section 2 describes similar works and highlights the gap in the literature; Section 3 formalizes the problem our framework addresses; Section 4 describes the proposed framework by detailing the transaction taxonomy, cost model, and methodology; Section 5 uses the methodology to apply the model and evaluate three real use cases. We finalize the paper with our conclusion and plans for future work.

2. Related Works

In recent years, there has been an increasing amount of literature on blockchain as a core component for developing new software systems. From a system and software-engineering perspective, blockchain applications present inherent challenges, such as scalability, security, and performance [6] given by a specific life cycle that introduces unique constraints and characteristics in the system [1]. Much of the current literature on blockchain-based systems pays particular attention to high-level characteristics such as the levels of permissions or types of actors to evaluate which software components can benefit from blockchain [11] or if other software components are sufficient [12]. However, monetary costs are still marginally addressed.

In this context, one of the first studies addressing the topic of costs for a blockchain-based system is presented in [13]. The authors aimed to quantify the current scalability limits of Bitcoin, and from that goal, they performed a small exploratory analysis of estimating monetary costs. However, since they focus on the scalability aspects of Bitcoin, they do not provide an approach compatible with a software-system perspective. Similarly, the authors of [14] address the cost of a blockchain-based system in the context of a blockchain-based digital payment. The authors rely on private Ethereum infrastructure and present a brief analysis of the costs of managing the private architecture. Their model focuses on the rewarding costs for miners and the costs of network resources, based on the same scalability metrics described in [13].

A different approach to cost is presented in [15] by describing an evaluation framework that identifies factors influencing a blockchain application from a financial perspective (i.e., cost savings and benefits). However, the proposal has a high-level approach that divides the framework into five focus areas: the purpose of the blockchain, the features of the blockchain, the cost reductions derived from using blockchain, environmental and motivational factors for using blockchain, and the actual implementation and operations costs. Therefore, infrastructure costs are linked to only one focus area and are addressed with a narrow perspective. Similarly, the authors of [16] present a cost comparison between the Ethereum blockchain and Amazon under the context of business process execution for supply chain applications. In this work, the authors focus on operation costs under different architecture choices and provide a good model for the Ethereum network. However, they focus on the comparison against Amazon Simple Workflow Service but need more generalization for other application scenarios or blockchain architectural choices.

Recently, authors of [17] presented an overview of blockchain-based applications with a focus on smart contracts in the Ethereum network. On the one hand, the work emphasized how Ethereum is becoming the preferred platform for developing blockchain-based applications, a fact also highlighted by surveys such as [18,19]. On the other hand, the study presented several metrics regarding the application, such as the level of open-source and the usage of patterns in smart contracts. Here, the authors focused only on analyzing existing applications, using gas usage as a metric to evaluate the costs of blockchain-based systems on public networks. Nonetheless, they do not provide a model or structure framework for this evaluation.

The importance of gas usage as a cost metric is also underlined by the authors of [20] in their study about developing cost-effective blockchain-powered applications. The authors emphasize that developers of these applications need to understand the gas of their smart contract through the entire application lifecycle (deployment and usage). Furthermore, the authors state that transactions with high gas usage will frequently have the same priority as transactions with low gas usage when using the same gas price, despite the difference in transaction fees. To leverage this topic, the authors propose a gas usage prediction models to help developers make informed decisions regarding gas prices. However, the authors do not address the cost of deploying or issuing the transaction in the context of a software system; they only focus on the gas price. The authors of [9] conducted preliminary work on infrastructure costs for blockchain-based systems based on gas usage. They provided a simple monetary cost model for the required infrastructure in a farm-to-fork case study but did not offer enough information to generalize to other architectures or case studies.

Summarizing, the infrastructure cost of blockchain-based applications has been marginally analyzed in current literature, which leans to focus on metrics such as scalability and performance or high-level concerns such as motivation or other environmental factors. Furthermore, only a few works provide frameworks or models for these cost analyses. There is a need for an approach that covers the entire lifecycle of blockchain-based applications or needs more details to provide enough generalization to evaluate different scenarios, particularly on public and private infrastructures.

3. Problem Definition

Our proposal aims to provide a model to allow actors and stakeholders to evaluate the economic feasibility of a blockchain-based application. The model seeks to estimate costs and monetary benefits during the entire life cycle of the application (i.e., from the early stages of development to a more advanced stage of exploitation). Furthermore, the model is general enough to evaluate a system using a private or a public blockchain network.

First, we consider a blockchain-based application as the software that supports the interactions of a group of actors identified only by their private/public keys. These actors have limited types of interactions to create and transfer information and value among them. We consider that the entire application logic is in the blockchain so that all information and interactions in the application are immutable, auditable, and accessible by anybody.

Further, we consider that the application has a two-phase life cycle: bootstrap and operation, similarly to the phases deployment and operation presented in [16]. It is important to notice that off-chain computations and side-chains are beyond the scope of this paper.

Using this definition of a blockchain-based application, we defined the cost of the application $C(m)$ as the infrastructure needed to process and store the interactions I between the actors A . These interactions generate a value unit K , which provides the monetary benefits $B(m)$ for a group of stakeholders S .

Therefore, we estimate $C(m)$ and $B(m)$ based on the characteristics of the application and the blockchain network that supports it, i.e., the nodes running the network. The number of nodes and configurations depends on the blockchains' technical implementation (e.g., consensus model). Furthermore, the configuration follows the application requirements (e.g., transaction life-span, latency, and throughput) and the existing trust between stakeholders [21]. However, the type of blockchain used for the application (i.e., public or private) makes a great difference in estimating the costs and benefits. On public blockchains, transactions that create new information (i.e., modify the state of the blockchain) have a monetary cost. Conversely, the transaction number does not directly affect the infrastructure cost in private blockchains. Therefore, we divided the model into two parts: the public blockchain cost model and the private blockchain cost model, described in the following section.

4. Proposed Cost and Benefit Model

In this section, we describe our proposed model to evaluate a blockchain-based application's infrastructure costs and benefits.

First, we define a transaction taxonomy since our model considers interactions among actors and stakeholders (i.e., transactions) as the functional units for an application's life-cycle assessment (LCA). The taxonomy is called *CRIV* and classifies the interactions between the actors in the framework of blockchain applications during all the development and exploitation phases.

Then, we define the costs and benefits model using the proposed taxonomy and a few simple system parameters. We model the costs of public and private blockchains separately, given that the transaction fees depend on the network type. Similarly, we model the monetary benefits for the stakeholders associated with a generic blockchain that can be public or private.

Finally, we proposed a methodology that guides the model's use by helping identify and estimate the model parameters.

Table 1 lists and describes the mathematical symbols used as the parameters for the model.

Table 1. Parameters for the proposed cost model.

Parameter	Description	Parameter	Description
A_0	Number of initial actors	S	Number of stakeholders
K	Value unit	L_k	K lifespan
$P_C(m)$	Cryptocurrency price	P_K	Value unit price
P_{node}	Node price	F_I	Interaction factor
μ	Time factor for transactions	F_g	Growth factor
O_C	Computational cost of T_C	F_V	Value factor
O_R	Computational cost of T_R	F_t	Trust factor
O_I	Computational cost of T_I	F_o	Operation factor
O_V	Computational cost of T_V	F_k	Benefit factor

4.1. Proposed Transaction Taxonomy for Blockchain Applications

As described in the previous section, the interactions among actors in a blockchain-based application are transactions. From the set \mathcal{I} of all possible transactions, we focus only on a subset $\mathcal{T} = \{T_i, i = [C, R, I, V]\} \subset \mathcal{I}$ of transactions that creates new information for the application and the actors. Considering that these transactions vary greatly from one application to another, we propose the CRIV taxonomy to easily identify core transactions and link them to the application’s life cycle. CRIV categorizes the interactions into four types of transactions: creation T_C , registration T_R , interaction T_I , and value T_V . Figure 1 shows the four transactions in our taxonomy along with the general life cycle of the application. Each type of transaction is defined as follows:

- Creation transaction T_C deploys the application into the blockchain. T_C may include one or more transactions happening during the bootstrap phase.
- Registration transaction T_R is required at the first interaction of an actor with the system to make the actor part of the application.
- Interaction transaction T_I are the most common interactions between actors. They produce information to be stored in the blockchain, without including any value transfer.
- Value transaction T_V is the most important transaction as it includes the value transfer between unknown actors.

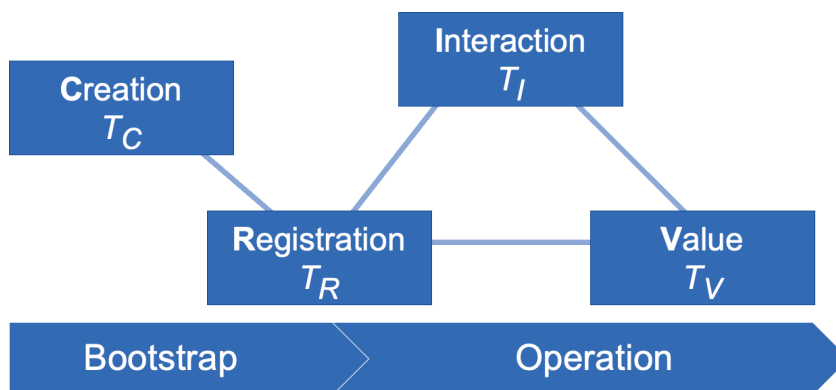


Figure 1. Life cycle of a blockchain-based application.

4.2. Proposed Public Blockchain Cost Model

Given the life cycle of an application shown in Figure 1, we divide the cost model into two components, i.e., the bootstrap C_B and operation C_O costs. C_B considers the transactions needed to deploy the application logic (T_C) and the transactions to register the initial actors (T_R). C_O considers the transactions for the registration of new actors (T_R), the interaction transactions between actors (T_I), and the transactions that transfer value (T_V). Here, considering the general life cycle of an application, C_B and C_O are evaluated in a given month m , defined as the minimal time window for assessing the systems. This window makes it easier to make comparisons with other types of monetary evaluations (e.g., budget planning). However, the cost model can easily adapt to shorter and longer windows. Hence, the initial month ($m = 0$) corresponds to the bootstrap phase, and any other month ($m > 0$) indicates the operation phase. We define the costs of a public blockchain infrastructure $C_{pub}(m)$ as:

$$C_{pub}(m) = \begin{cases} C_B(m) = CT_C(m) + CT_R(m), & \text{if } m = 0 \\ C_O(m) = CT_R(m) + CT_I(m) + CT_V(m) & m > 0 \end{cases} \quad (1)$$

where $CT_i(m)$ is the total monetary costs paid in a month m , for all the transactions of type i , where $i = \{C, R, I, V\}$. The monetary cost of a single transaction of type i links the computational cost O_i of the transaction with the price of the cryptocurrency $P_C(m)$ and a processing time factor μ . Finally, the total number of transactions of type i in a given month

m is given by $Q_i(m)$. The price of the cryptocurrency $P_C(m)$ is a function that the user must define by considering the high volatility of the cryptocurrency price and the scenario to evaluate. For instance, the user can use historic cryptocurrency prices to define $P_C(m)$ with a fixed value for any month (i.e., an average for all months). Similarly, the user can define $P_C(m)$ with a different monthly value (i.e., an average for each month). The factor μ is used to scale the price paid for each transaction (i.e., a transaction fee). Transactions with higher prices are more attractive for the node operators and typically are processed faster since the node that processes the transaction will receive the fee as a reward [20]. The total cost $CT_i(m)$ for each transaction i is given by:

$$CT_i(m) = O_i P_C(m) \mu Q_i(m), \quad \text{with } i \in \{C, R, I, V\} \tag{2}$$

In the operational phase, the total number of transactions of each type i in a given month m is directly related to the number of actors $A(m)$ in the system defined as:

$$A(m) = \begin{cases} A_0, & \text{if } m = 1 \\ A(m-1) F_g & m > 1 \end{cases} \tag{3}$$

where A_0 is the initial number of actors in the system and F_g is the actor growth factor that describes the growth of the system in terms of actors related to the time unit m such that:

$$F_g = (A(m) / A(m-1)) - 1 \tag{4}$$

For instance, a $F_g = 0.5$ means that if $A(m-1) = 100$, then at $A(m) = 150$. Finally, the number of each type of transaction $Q_i(m)$ in the operation phase ($m > 0$) with respect to the actor number is defined as:

$$Q_i(m) = \begin{cases} 0 & \text{for } i = T_C, m = 0 \\ A(m) - A(m-1) & \text{for } i = T_R, m \geq 0 \\ A(m) F_I & \text{for } i = T_I, m > 0 \\ A(m) F_V & \text{for } i = T_V, m > 0 \end{cases} \tag{5}$$

where $Q_C(m)$ is equal to 0 after the bootstrap phase ($m > 0$). $Q_R(m)$ is given by the number of new actors in that month. $Q_I(m)$ links the total number of actors in the system $A(m)$ using an interaction factor F_I . F_I relates to the expected number of interaction transactions T_I of each actor. For example, if each actor is expected to have at least two T_I in the time unit m (e.g., T_I per month), the factor is set to $F_I = 2$. Lastly, $Q_V(m)$ links the total number of actors with factor F_V , which represents the value transfer transactions T_V of each actor. For instance, when actors are expected to have at least one T_V every two months, the factor is set to $F_V = 0.5$. The user can estimate the values of F_I and F_V at an early stage of development. In a more advanced stage, the factors can be estimated from the current activity in the system.

4.3. Proposed Private Blockchain Cost Model

For applications based on a private blockchain, we define the infrastructure cost $C_{Pri}(m)$ in a given month m as divided into two components C_B and C_O , indicating the bootstrap and the operation phase, respectively. C_B is the initial investment to acquire N nodes (i.e., computers) with a price P_{node} to create the network infrastructure. C_O is the expense of running and operating the nodes. Similar to the traditional software systems, we estimate the operating costs C_O as a percentage of P_{node} using a scale factor F_0 . F_0 is estimated by considering the system characteristics in terms of the hardware and software required to run the nodes. The infrastructure cost $C_{Pri}(m)$ in a private blockchain is defined as:

$$C_{Pri}(m) = \begin{cases} C_B(m) = N P_{node} & \text{if } m = 0 \\ C_O(m) = N F_0 P_{node} & m > 0, \end{cases} \tag{6}$$

In our model, the node number N is related to the number of stakeholders S by a trust factor F_t :

$$N = S (1 - F_t). \quad (7)$$

where F_t is the relation between the N and the total number of stakeholders. For instance, if 100 stakeholders agree that only 30 different nodes are required to support the infrastructure, there is a trust factor of 70%. Similarly, a 100% trust factor will translate into a centralized system. Here, for simplicity, one node represents one stakeholder.

4.4. Proposed Model for Monetary Benefits

For applications based on both public and private blockchains, the monetary benefits $B(m)$ for the stakeholders are derived from the value units K transacted in the application and are given by:

$$B(m) = F_k Q_V(m) P_K \quad (8)$$

where F_k is the benefit factor that indicates the expected value units for each value transfer transaction T_V , $Q_V(m)$ is the number of value transfer transactions in the month m , and P_K is the price (i.e., total monetary value) of the value unit K . P_K is the sum of all the monetary values assigned to each stakeholder S (i.e., the benefit for each stakeholder). In a public network, this value may also be linked to the price of the cryptocurrency, such as $P_K(m) = 0.4 P_C(m)$.

4.5. Proposed Methodology

From a software system perspective, a methodology is a procedure to help understand the steps needed to perform a task with such a system [22]. Here, we propose a methodology of five steps to guide the people behind the blockchain-based application (i.e., the user) to use our model to perform a monetary evaluation of the application. The methodology groups the model parameters into four categories, using the relations between the parameters. These four groups translate into four steps (S1–S4), providing a simplified incremental approach to identifying them. The last step of the methodology (S5) is the actual monetary evaluation of the application and includes a series of proposed analyses. The five steps of the methodology are:

- (S1) Define the blockchain setup: The first step aims at selecting the blockchain network on which the application will be implemented (e.g., Ethereum, Hyperledger). Assign the values for the network's node price P_{node} , the operation factor F_o , the cryptocurrency price $P_C(m)$, and the time factor μ .
- (S2) Identify the actors and stakeholders: Once the blockchain network is selected, the second step aims at identifying the actors A_0 (i.e., who uses the application), stakeholders S (i.e., who is interested in the application), the value unit K , and its lifespan L_k (i.e., what is transacted in the application and how long it lasts). Estimate how the number of actors will change F_g and how much trust is between the stakeholders F_t .
- (S3) Estimate the computational cost of interactions: The third step strives to apply the proposed CRIV taxonomy to identify the different types of transactions and estimate their computation costs O_c , O_r , O_i , and O_v . Estimate the interaction factor F_i and the value factor F_v .
- (S4) Identify the benefits: The fourth step aims to give a monetary value to the value unit P_k and estimate the benefits factor F_k . If these parameters can not be identified, the model can still be used for estimating the costs, but the benefits will not be available. At the end of this step, all the model parameters have been identified, even if some values could not be estimated.
- (S5) Evaluate scenarios: Once all parameters have been identified, the values assigned provide a scenario to evaluate the blockchain-based application's cost and benefits. Changing the model parameters value can provide different scenarios to compare (different

F_g , different $P_C(m)$. Some of the most common evaluations include: using Equation (1) to estimate the bootstrap and operation costs on a public network or using Equation (6) for a private network. Another example is using Equation (8) for estimating the benefits. Furthermore, equations and parameters can be combined to obtain other evaluations. For instance, dividing Equation (1) by the number of actors on a given month $C(m)/A(m)$ can provide an estimate of how much each actor will pay for the system operation. For each evaluation, varying the model parameters value can provide different scenarios to compare (different F_g , different $P_C(m)$). These are just a few examples of the model's usability.

5. Evaluation of the Proposed Model

To evaluate the correctness and goodness of our proposal, we evaluated a series of blockchain-based applications in the current literature [4,23–25]. We selected applications using Ethereum, as it is a reference implementation for smart contracts and can be used in both public and private scenarios [18]. However, our proposed model can be used with any other blockchain implementation.

We present three use cases: a water management system using a public network, a medical image system that can be used on public or private networks, and a manufacturing traceability application using a private network. For each use case, we show how to apply the proposed model following our methodology for defining the model parameters. Given that all applications run on the Ethereum network, the first step of the methodology is common for all use cases. Then, for each use case, we provide analyses highlighting our model's potential when evaluating the costs and benefits of blockchain-based applications.

5.1. (S1) Define the Blockchain Setup

We define Ethereum as the blockchain network for all use cases. The cryptocurrency price $P_C(m)$ is the price of Ethereum, using historical values that are available online (Etherscan¹). The computational cost O_i of the transactions is equal to the gas required for their execution, and μ is the gas price on Ethereum, expressed in gwei. For the cost of the node P_{node} , we consider the minimum hardware requirements for an Ethereum node². At the time of writing, this translates into a computer of USD 300. We consider the operation factor for the node as 40% of the cost of the node, this $F_o = 0.4$.

5.2. Use Case: Water Management System

The authors of [4] present an architecture for a blockchain-based IoT water management system. The authors implement a prototype using Ethereum as a public blockchain and constrained IoT devices as data sources. The authors evaluate focused on the IoT devices and provide implementation details of the smart contracts. Some parameters of our model are clearly expressed (i.e., A, S, K, L_k), while others require a brief analysis to be estimated (i.e., P_k). Thus, following our methodology, the steps are:

5.2.1. (S2) Identify Actors and Stakeholders

The group of actors A comprises farmers using IoT devices to measure water consumption (i.e., a valve). The stakeholders S are three organizations interested in encouraging water savings (i.e., an energy company, NGO, and certification authorities), thus $S = 3$. The value unit K is a cubic meter of the saved water, and the lifespan unit is a day, as water usage is reported daily. We estimated the initial number of actors is $A_0 = 100$ with a monthly growth of 5%, making $F_g = 0.5$, based on the information described in the paper and the references within it.

5.2.2. (S3) Estimate the Computation Cost of Interactions

The application is based on two types of smart contracts with four types of transactions. These transactions can be directly mapped to our taxonomy C_C, C_R, C_I , and C_V . The computation cost is calculated based on the gas usage reported for the transactions. The

farmers report their water consumption once a week, which translates into four $F_I = 4$ (four transactions T_I per month), and they receive their rewards once a month, thus, $F_V = 1$ (one transaction T_V per month).

5.2.3. (S4) Identify the benefits

Each actor saves on average 4 m^3 in 100 ha farm as described in similar studies [26]. Hence, the benefit factor is set to $F_k = 4$. The authors do not provide a monetary value for a m^3 of saved water (P_K), so it must be estimated based on the document and its references. We estimated that the energy company offers a discount of USD 0.2 for each saved m^3 . We considered NGO assigns to the savings a value equal to the cost of irrigation m^3 at USD 0.8 (according to [26]). Finally, we assume a “eco-friendly” label will translate into USD 10 additional monthly benefits. Thus, the total monetary value of K is USD 11. Table 2 summarizes the value of the parameters for the application, obtained following the proposed methodology, and that will be used for evaluating scenarios.

Table 2. Parameters for the cost model of the water-management system from [4].

Parameter	Value	Parameter	Value
A_0	100	S	3
K	Saved m^3 of water	L_k	1 day
$P_C(m)$	205	P_K	USD 11
P_{node}	USD 300	F_I	4
μ	2, 5, 10 gwei	F_g	0.05
O_C	3,343,572	F_V	1
O_R	143,947	F_t	0.01
O_I	26,821	F_o	0.4
O_V	156,580	F_k	1

5.2.4. (S5) Evaluate Scenarios

In their work, the authors present a brief evaluation of the transaction costs using three different gas prices (i.e., 2, 5, and 10 gwei) with a cryptocurrency price equal to USD 205 (based on the yearly average for 2019), as shown in Table 3. Our model uses Equation (2) and the proposed taxonomy to obtain these values.

Table 3. Transaction costs for the water management use case [4].

	Gas	2 Gwei	5 Gwei	10 Gwei
Valve Creation	143.947	USD 0.059	USD 0.148	USD 0.295
setValue()	26.821	USD 0.011	USD 0.027	USD 0.055
App Creation	3.343.572	USD 1.371	USD 3.427	USD 6.854
Reward	156.580	USD 0.064	USD 0.160	USD 0.321

Furthermore, our model can extend the author’s evaluation to different scenarios. Considering the monthly average price for 2019, we can use Equations (1), (6), and (8) to evaluate the costs and benefits of the application for a year. Figure 2 shows the benefits ($Benef$), the total monthly cost of a private network ($CPri$), and the total monthly for a public network using 2, 5, and 10 gwei ($CPub_2$, $CPub_5$, $CPub_{10}$, respectively), evaluated for the year 2019.

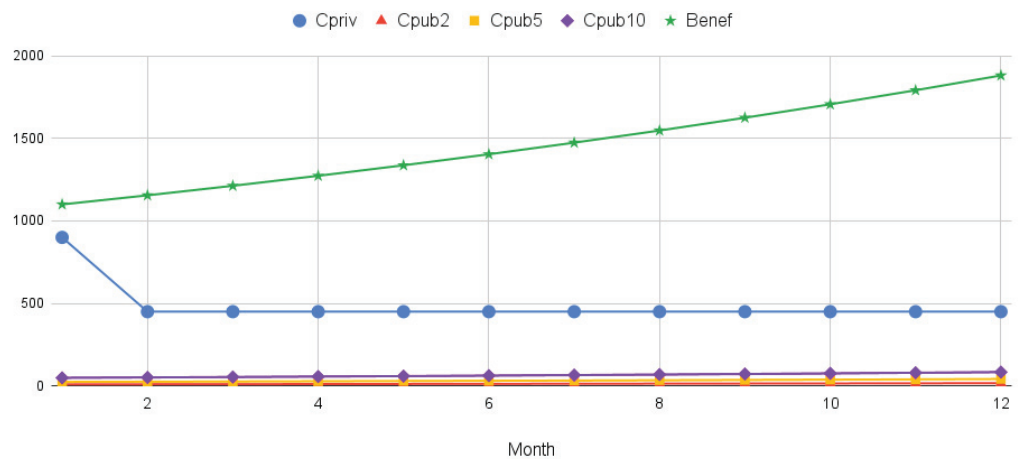


Figure 2. Monthly costs and benefits of the water management system.

With this use case, we highlight the correctness of our model to match the existing literature on cost. Furthermore, we highlight its potential for analyzing more complex cost scenarios with minimal additional information. For instance, those developing the blockchain-based application should easily identify the values we have estimated (i.e., K , P_k).

5.3. Use Case: Patient-Centric Image Management System

Jabarulla and Lee propose a blockchain-based patient-centric image management system [24]. They developed a proof-of-concept using the Ethereum blockchain and a distributed storage system. The authors validate their proposal with experiments and evaluate gas usage as a metric for executing functions. Furthermore, they assigned a monetary value to the gas to provide a price reference for the system.

5.3.1. (S2) Identify Actors and Stakeholders

The actors A are patients, doctors, and practitioners involved. The value unit K is a medical image with a lifespan L_K of 3 months. As presented in the paper, we define $A_0 = 4$ with a growth factor $F_g = 0.75$. However, more information is needed to identify the stakeholders S .

5.3.2. (S3) Estimate the Computation Cost of Interactions

Based on the source code provided by the authors, we can obtain the value for O_C . The authors provide the gas usage for the contract functions, and using our taxonomy, we can obtain the values for O_R and O_V . There needs to be more detail to estimate O_I based on the three functions described, so we average the values. Then, we define F_I and F_V as 1, meaning we consider sharing one image (F_I) and accessing one image (F_V).

5.3.3. (S4) Identify the Benefits

The authors state that an average transaction price of USD 0.11 is lower than existing solutions for managing patients' images. Therefore, we consider P_K as USD 0.11 and set F_K as 1. Table 4 summarizes the model's parameters.

Table 4. Parameters for the cost model of patient-centric image management system [24].

Parameter	Value	Parameter	Value
A_0	4	S	-
K	medical image	L_k	3 months
$P_C(m)$	USD 187	P_K	USD 0.11
P_{node}	USD 300	F_I	1
μ	2 gwei	F_g	0.75
O_C	1,611,435	F_V	1
O_R	67,397	F_t	-
O_I	113,510	F_o	0.4
O_V	170,412	F_k	1

5.3.4. (S5) Evaluate Scenarios

The authors use a cryptocurrency price of USD 187 and a gas price of 2 gwei to provide an average transaction price of USD 0.11. With our model, we can obtain the average transaction price by dividing the total costs Equation (2) by the number of actors Equation (3). This operation renders a value of USD 0.12, where the minimal difference is due to the estimation of the computation cost of interactions. Then, we extend the author’s evaluation by analyzing the impact of adding more images per user (changing the parameter F_I) impact the costs. Figure 3 shows this cost with a baseline of USD 0.12.

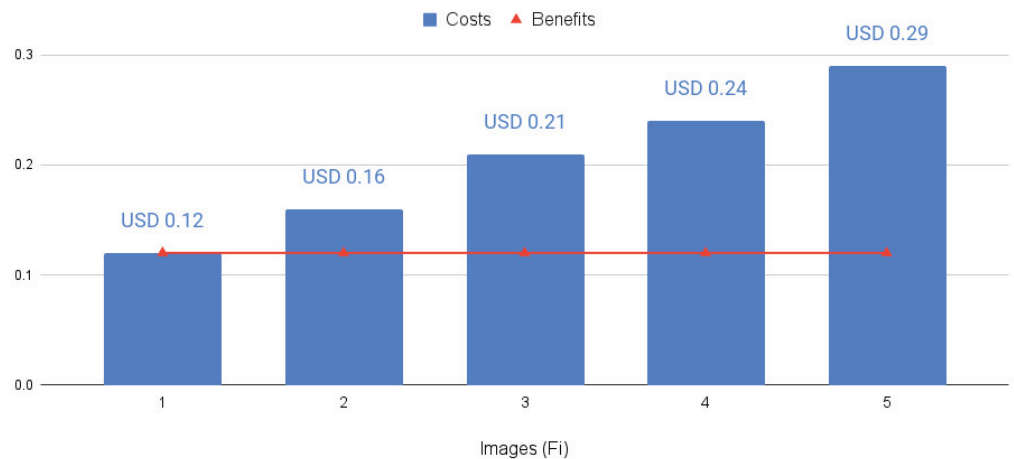


Figure 3. Comparison of average transaction price using different values for F_V and F_I .

This evaluation highlights the correctness of our model to match existing approaches to evaluate costs. Furthermore, the model offers additional value by providing the tools to evaluate different scenarios even if a few parameters can not be estimated.

5.4. Use Case: Automotive Manufacturing Traceability

Kuhn et al. [23] propose a blockchain-based traceability architecture to process manufacturing data. The authors present an evaluation of gas used per transaction as a metric for scalability without a monetary evaluation. Compared with the other use cases, this application does not provide enough information to estimate several model parameters. However, the model can still be used as follows.

5.4.1. (S2) Identify Actors and Stakeholders

The value unit is manufactured (i.e., electrical contacts) with a lifespan of L_k of one day. The stakeholders S are the companies involved in the manufacturing process, each providing a node for the blockchain network with $N \in [10, 50]$. Since each stakeholder provides a node, the trust factor is F_t is 0. The actors A are the machines and devices in the

production process with $A \in [10, 100]$. Unfortunately, there is not enough information to estimate a growth factor F_g .

5.4.2. (S3) Estimate the Computation Cost of Interactions

All the interactions are managed through a single contract based on the ERC1155 token standard and deployed on a private Ethereum network. The paper provides results regarding gas usage for a single stakeholder, processing a batch of 3000 items, which means 3000 transactions. However, more details are needed for applying the taxonomy or estimating F_v and F_i .

5.4.3. (S4) Identify the Benefits

Based on the paper [23] and the references within, we can define P_K as USD 0.5 for each processed unit using a Benefit factor $F_k = 1$. Table 5 summarizes the value of the parameters for the use case.

Table 5. Parameters for the cost model of the architecture for automotive traceability [23].

Parameter	Value	Parameter	Value
A_0	50	S	25
K	electric contact	L_k	1 day
$P_C(m)$	-	P_K	USD 0.5
P_{node}	USD 300	F_I	-
μ	-	F_g	0
O_C	-	F_V	-
O_R	-	F_t	-
O_I	-	F_o	0.4
O_V	-	F_k	1

5.4.4. (S5) Evaluate scenarios

Although if this use case only provides enough information for estimating 6 of the 18 parameters, it can be used to calculate additional cost information. For instance, using Equation (6), the bootstrap cost is USD 7.500 for acquiring 25 nodes, and the operation cost is fixed at USD 3500. Then, making the monthly costs equal to the benefits described by Equation (8), the total number of transactions $Qv(m)$ should be 7000 to reach an equilibrium value.

This use case highlights the usability of the model, even when not all parameters can be defined or estimated. With only 6 of the 18 parameters, the model provided the tools to find a monetary balance point for a private Ethereum network.

5.5. Discussion

In the previous sections, we evaluated our proposed model and methodology with three use cases from the current literature. In the first use case, we demonstrated that our model could match the existing literature on cost and has the potential to analyze more complex cost scenarios. In the second use case, we further demonstrated the correctness of our model in matching existing cost evaluation approaches, even if a few parameters cannot be estimated. In the last use case, we highlighted the usability of our model with minimal available information. The results and rationale of using our framework in these use cases highlighted the usability of the selected parameters and the methodology to identify them. On the one hand, our selection of static parameters, even if it can be considered a limitation, strikes simplicity and effectiveness and proved useful, particularly with little application domain knowledge. On the other hand, the proposed methodology to guide the users was also demonstrated to be effective in maintaining a streamlined and straightforward approach while still being widely applicable. Finally, the quantitative information showcased by the example cost and benefits analysis performed on each use

case provides an empirical reference that can further enrich the growing field of studying blockchain-based systems.

6. Conclusions and Future Works

In this paper, we presented a framework for evaluating the costs and benefits of the blockchain-based system across its entire life cycle. The proposed framework includes a transaction taxonomy, a cost and benefit model, and a methodology to use the model. We used the methodology to apply the proposed model and quantitatively evaluate the cost and benefits of three use cases found in the current literature. The analyses highlight the model's accuracy and usability in evaluating different types of blockchain-based applications. In particular, the proposed methodology emphasizes the simplicity of identifying and estimating the model parameters. Once the parameters have been identified, the evaluation shows how to assess different scenarios by simply varying the values of some parameters. Furthermore, the diverse use cases provided different application details, showcasing the model's potential even when some parameters could not be identified or estimated. All these features make our proposed methodology a valuable tool for organizations that want to estimate the costs associated with implementing blockchain-based systems in various domains. By leveraging our model's usability and flexibility, they can make informed decisions about such systems' feasibility and expected returns on investment. Additionally, the empirical reference provided by our quantitative information can serve as a benchmark for future research in this field, enabling researchers to explore new areas of study more effectively. Overall, our model and methodology offer a powerful combination of simplicity, effectiveness, and versatility that can benefit industry practitioners and academic researchers.

Future works include assessing other use cases to improve the methodology and provide reference values for the model parameters. Similarly, studying which parameters can benefit dynamic values is an interesting research path. Finally, studying a possible hybrid model combining public and private blockchain networks could enhance our proposed framework.

Author Contributions: Conceptualization, M.P. and R.G.; Methodology, M.P. and E.D.; Validation, M.P. and E.D.; Formal analysis, M.P. and E.D.; Investigation, M.P. and E.D.; Data curation, M.P. and E.D.; Writing—original draft, M.P. and E.D.; Supervision, M.V. and R.G.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the project "AI@TN" funded by the Autonomous Province of Trento.

Data Availability Statement: Data available on request due to restrictions. The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Notes

¹ <https://etherscan.io/chart/etherprice>(accessed on 20th January 2023).

² <https://docs.ethhub.io/using-ethereum/ethereum-clients/geth/> (accessed on 20th January 2023).

References

1. Vacca, A.; Di Sorbo, A.; Visaggio, C.A.; Canfora, G. A systematic literature review of blockchain and smart contract development: Techniques, tools, and open challenges. *J. Syst. Softw.* **2021**, *174*, 110891. [[CrossRef](#)]
2. Fahmideh, M.; Grundy, J.; Ahmad, A.; Shen, J.; Yan, J.; Mougouei, D.; Wang, P.; Ghose, A.; Gunawardana, A.; Aickelin, U.; et al. Engineering Blockchain-Based Software Systems: Foundations, Survey, and Future Directions. *ACM Comput. Surv.* **2022**, *55*, 1–44. [[CrossRef](#)]
3. Mistry, I.; Tanwar, S.; Tyagi, S.; Kumar, N. Blockchain for 5G-enabled IoT for industrial automation: A systematic review, solutions, and challenges. *Mech. Syst. Signal Process.* **2020**, *135*, 106382. [[CrossRef](#)]
4. Pincheira, M.; Vecchio, M.; Giaffreda, R.; Kanhere, S.S. Cost-effective IoT devices as trustworthy data sources for a blockchain-based water management system in precision agriculture. *Comput. Electron. Agric.* **2021**, *180*, 105889. [[CrossRef](#)]

5. Pincheira, M.; Donini, E.; Giaffreda, R.; Vecchio, M. A Blockchain-Based Approach to Enable Remote Sensing Trusted Data. In Proceedings of the 2020 IEEE Latin American GRSS ISPRS Remote Sensing Conference (LAGIRS), Santiago, Chile, 22–26 March 2020; pp. 652–657. [[CrossRef](#)]
6. Destefanis, G.; Marchesi, M.; Ortu, M.; Tonelli, R.; Bracciali, A.; Hierons, R. Smart contracts vulnerabilities: A call for blockchain software engineering? In Proceedings of the 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE), Campobasso, Italy, 20 March 2018; pp. 19–25. [[CrossRef](#)]
7. Xu, X.; Lu, Q.; Liu, Y.; Zhu, L.; Yao, H.; Vasilakos, A.V. Designing blockchain-based applications a case study for imported product traceability. *Future Gener. Comput. Syst.* **2019**, *92*, 399–406. [[CrossRef](#)]
8. Wöhrer, M.; Zdun, U. Architectural Design Decisions for Blockchain-Based Applications. In Proceedings of the The 3rd IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Sydney, Australia, 3–6 May 2021.
9. Pincheira, M.; Vecchio, M.; Giaffreda, R. Characterization and Costs of Integrating Blockchain and IoT for Agri-Food Traceability Systems. *Systems* **2022**, *10*, 57. [[CrossRef](#)]
10. Pincheira, M.; Donini, E.; Vecchio, M.; Giaffreda, R. Towards an Infrastructure Cost Model for Blockchain-Based Applications. In *Blockchain and Applications, Proceedings of the 4th International Congress, Lille, France, 11–13 July 2023*; Prieto, J., Benítez Martínez, F.L., Ferretti, S., Arroyo Guardado, D., Tomás Nevado-Batalla, P., Eds.; Springer International Publishing: Cham, Switzerland, 2023; pp. 345–355.
11. Wessling, F.; Ehmke, C.; Hesenius, M.; Gruhn, V. How much blockchain do you need? towards a concept for building hybrid dapp architectures. In Proceedings of the 2018 IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), Gothenburg, Sweden, 27 May 2018; IEEE: New York, NY, USA, 2018; pp. 44–47.
12. Wüst, K.; Gervais, A. Do you need a blockchain? In Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), Zug, Switzerland, 20–22 June 2018; IEEE: New York, NY, USA, 2018; pp. 45–54.
13. Croman, K.; Decker, C.; Eyal, I.; Gencer, A.E.; Juels, A.; Kosba, A.; Miller, A.; Saxena, P.; Shi, E.; Sirer, E.G.; et al. On Scaling Decentralized Blockchains. In *Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 106–125.
14. Hu, Y.; Manzoor, A.; Ekparinya, P.; Liyanage, M.; Thilakarathna, K.; Jourjon, G.; Seneviratne, A. A Delay-Tolerant Payment Scheme Based on the Ethereum Blockchain. *IEEE Access* **2019**, *7*, 33159–33172. [[CrossRef](#)]
15. Demir, M.; Turetken, O.; Ferworn, A. A Financial Evaluation Framework for Blockchain Implementations. In Proceedings of the 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 17–19 October 2019; pp. 715–722. [[CrossRef](#)]
16. Rimba, P.; Tran, A.B.; Weber, L.; Staples, M.; Ponomarev, A.; Xu, X. Quantifying the Cost of Distrust: Comparing Blockchain and Cloud Services for Business Process Execution. *Inf. Syst. Front.* **2020**, *22*, 489–507. [[CrossRef](#)]
17. Wu, K.; Ma, Y.; Huang, G.; Liu, X. A first look at blockchain-based decentralized applications. *Softw. Pract. Exp.* **2021**, *51*, 2033–2050. [[CrossRef](#)]
18. Kondo, M.; Oliva, G.A.; Jiang, Z.M.; Hassan, A.E.; Mizuno, O. Code cloning in smart contracts: A case study on verified contracts from the Ethereum blockchain platform. *Empir. Softw. Eng.* **2020**, *25*, 4617–4675. [[CrossRef](#)]
19. Oliva, G.A.; Hassan, A.E.; Jiang, Z.M.J. An exploratory study of smart contracts in the Ethereum blockchain platform. *Empir. Softw. Eng.* **2020**, *25*, 1864–1904. [[CrossRef](#)]
20. Zarir, A.A.; Oliva, G.A.; Jiang, Z.M.; Hassan, A.E. Developing cost-effective blockchain-powered applications: A case study of the gas usage of smart contract transactions in the ethereum blockchain platform. *ACM Trans. Softw. Eng. Methodol. (TOSEM)* **2021**, *30*, 1–38. [[CrossRef](#)]
21. Schäffer, M.; Di Angelo, M.; Salzer, G. Performance and scalability of private Ethereum blockchains. In Proceedings of the International Conference on Business Process Management, Rome, Italy, 6–10 September 2019; Springer: Berlin, Germany, 2019; pp. 103–118.
22. Karat, J. Software Evaluation Methodologies. In *Handbook of Human-Computer Interaction*; Helander, M., Ed.; Elsevier: Amsterdam, The Netherlands, 1988; pp. 891–903. [[CrossRef](#)]
23. Kuhn, M.; Funk, F.; Franke, J. Blockchain architecture for automotive traceability. *Procedia Cirp* **2021**, *97*, 390–395. [[CrossRef](#)]
24. Jabarulla, M.Y.; Lee, H.N. Blockchain-based distributed patient-centric image management system. *Appl. Sci.* **2020**, *11*, 196. [[CrossRef](#)]
25. Kudva, S.; Norderhaug, R.; Badsha, S.; Sengupta, S.; Kayes, A. PEBERS: Practical Ethereum Blockchain based Efficient Ride Hailing Service. In Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2–5 February 2020; pp. 422–428. [[CrossRef](#)]
26. Giannakis, E.; Bruggeman, A.; Djuma, H.; Kozyra, J.; Hammer, J. Water pricing and irrigation across Europe: Opportunities and constraints for adopting irrigation scheduling decision support systems. *Water Supply* **2015**, *16*, 245–252. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Agent-Based Collaborative Random Search for Hyperparameter Tuning and Global Function Optimization [†]

Ahmad Esmaeili ^{*}, Zahra Ghorrati and Eric T. Matson

Department of Computer and Information Technology, Purdue University, West Lafayette, IN 47907, USA

^{*} Correspondence: aesmaei@purdue.edu[†] This paper is an extended version of our paper published in the proceedings of the 20th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2022), L'Aquila, Italy, 13–15 July 2022.

Abstract: Hyperparameter optimization is one of the most tedious yet crucial steps in training machine learning models. There are numerous methods for this vital model-building stage, ranging from domain-specific manual tuning guidelines suggested by the oracles to the utilization of general purpose black-box optimization techniques. This paper proposes an agent-based collaborative technique for finding near-optimal values for any arbitrary set of hyperparameters (or decision variables) in a machine learning model (or a black-box function optimization problem). The developed method forms a hierarchical agent-based architecture for the distribution of the searching operations at different dimensions and employs a cooperative searching procedure based on an adaptive width-based random sampling technique to locate the optima. The behavior of the presented model, specifically against changes in its design parameters, is investigated in both machine learning and global function optimization applications, and its performance is compared with that of two randomized tuning strategies that are commonly used in practice. Moreover, we have compared the performance of the proposed approach against particle swarm optimization (PSO) and simulated annealing (SA) methods in function optimization to provide additional insights into its exploration in the search space. According to the empirical results, the proposed model outperformed the compared random-based methods in almost all tasks conducted, notably in a higher number of dimensions and in the presence of limited on-device computational resources.

Keywords: multi-agent systems; distributed machine learning; hyperparameter tuning; agent-based optimization; random search

Citation: Esmaeili, A.; Ghorrati, Z.; Matson, E.T. Agent-Based Collaborative Random Search for Hyperparameter Tuning and Global Function Optimization. *Systems* **2023**, *11*, 228. <https://doi.org/10.3390/systems11050228>

Academic Editors: Philippe Mathieu, Juan M. Corchado, Alfonso González-Briones and Fernando De la Prieta Pintado

Received: 2 March 2023

Revised: 1 May 2023

Accepted: 3 May 2023

Published: 5 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Almost all machine learning (ML) algorithms comprise a set of hyperparameters that control their learning process and the quality of their resulting models. The number of hidden units, the learning rate, the mini-batch sizes, etc., in neural networks, the kernel parameters and regularization penalty amount in support vector machines, and maximum depth, sample split criteria, and the number of used features in decision trees are a few common hyperparameter examples that need to be configured for the corresponding learning algorithms. Assuming a specific ML algorithm and a dataset, one can build a countless number of models each with a potentially different performance and/or learning speeds, by assigning different values to the algorithm's hyperparameters. While they provide ultimate flexibility in using ML algorithms in different scenarios, they also account for most failures and tedious development procedures. Unsurprisingly, there are numerous studies and practices in the machine learning community devoted to the optimization of hyperparameters. The most straightforward yet difficult approach utilizes expert knowledge to identify potentially better candidates in hyperparameter search spaces to evaluate and use. The availability of expert knowledge and generating reproducible results are among

the primary limitations of such a manual searching technique [1], particularly due to the fact that using any learning algorithm on different datasets likely requires different sets of hyperparameter values [2].

Formally speaking, let $\Lambda = \{\lambda\}$ denote the set of all possible hyperparameter value vectors and $\mathcal{X} = \{\mathcal{X}^{(train)}, \mathcal{X}^{(valid)}\}$ be the dataset split into training and validation sets. The learning algorithm with hyperparameter values vector λ is a function that maps training dataset $\mathcal{X}^{(train)}$ to model \mathcal{M} , i.e., $\mathcal{M} = \mathcal{A}_\lambda(\mathcal{X}^{(train)})$, and the hyperparameter optimization problem can be formally written as [1]:

$$\lambda^{(*)} = \arg \min_{\lambda \in \Lambda} \mathbb{E}_{x \sim \mathcal{G}_x} \left[\mathcal{L}(x; \mathcal{A}_\lambda(\mathcal{X}^{(train)})) \right] \quad (1)$$

where \mathcal{G}_x and $\mathcal{L}(x; \mathcal{M})$ are, respectively, the grand truth distribution and the expected loss of applying learning model \mathcal{M} over i.i.d. samples x ; and $\mathbb{E}_{x \sim \mathcal{G}_x} \left[\mathcal{L}(x; \mathcal{A}_\lambda(\mathcal{X}^{(train)})) \right]$ gives the generalization error for algorithm \mathcal{A}_λ . To cope with the inaccessibility of the grand truth in real-world problems, the generalization error is commonly estimated using the cross-validation technique [3], leading to the following approximation of the above-mentioned optimization problem:

$$\lambda^{(*)} \approx \arg \min_{\lambda \in \Lambda} \text{mean}_{x \in \mathcal{X}^{(valid)}} \mathcal{L}(x; \mathcal{A}_\lambda(\mathcal{X}^{(train)})) \equiv \arg \min_{\lambda \in \Lambda} \Psi(\lambda) \quad (2)$$

where $\Psi(\lambda)$ is called the hyperparameter response function [1].

Putting the manual tuning approaches aside, there is a wide range of techniques that use black-box optimization methods to address the ML hyperparameter tuning problem. Grid search [4,5], random search [1], Bayesian optimization [6–8], and evolutionary and population-based optimizations [9,10] are some common tuning methodologies that are studied and used extensively by the community. In grid search for instance, every combination of a predetermined set of values in each hyperparameter is evaluated, and the hyperparameter value vector that minimizes the loss function is selected. For k number of configurable hyperparameters, if we denote the set of candidate values for the j -th hyperparameter $\lambda_j^{(i)} \in \Lambda^{(i)}$ by \mathcal{V}_j , the grid search would evaluate $T = \prod_{j=1}^k |\mathcal{V}_j|$ number of trials that can grow exponentially with the increase in the number of configurable hyperparameters and the quantity of the candidate values for each dimension. This issue is referred to as the curse of dimensionality [11] and is the primary reason for making grid search an uninteresting methodology in large-scale real-world scenarios. Moreover, in the standard random search, a set of b uniformly distributed random points in the hyperparameter search space, $\{\lambda^{(1)}, \dots, \lambda^{(b)}\} \in \Lambda$ are evaluated to select the best candidate. As the number of evaluations only depends on the budget value b , a random search does not suffer from the curse of dimensionality, is shown to be more effective than grid search [1], and is often used as a baseline method. Bayesian optimization, as a global black-box expensive function optimization technique, iteratively fits a surrogate model to the available observations $(\lambda^{(i)}, \Psi_{\lambda^{(i)}})$, and then uses an acquisition function to determine the next hyperparameter values to evaluate and use in the next iteration [8,12]. Unlike grid and random search methods, in which the searching operations can be easily parallelized, the Bayesian method is originally sequential, though various distributed versions have been proposed in the literature [13,14]. Nevertheless, thanks to its sample efficiency and robustness to noisy evaluations, Bayesian optimization is a popular method in the hyperparameter tuning of deep-learning models, particularly when the number of configurable hyperparameters is less than 20 [15]. Evolution and population-based global optimization methods, such as genetic algorithms and swarm-based optimization techniques, form the other class of common tuning approaches in which the hyperparameter configurations are improved over multiple generations generated by local and global perturbations [10,16]. Population-based methods are embarrassingly parallel [17] and, similar to grid and random search approaches, the evaluations can be distributed over multiple machines.

Multi-agent systems (MAS) and agent-based technologies, when applied to machine learning and data mining, bring about scalability and autonomy and facilitate the decentralization of learning resources and the utilization of strategic and collaborative learning models [18–20]. Neither agent-based machine learning nor collaborative hyperparameter tuning are novelties of this paper, as they have been previously studied in the literature. The research reported in [21] is among the noteworthy contributions, which proposes a surrogate-based collaborative tuning technique incorporating the experience achieved from previous experiments. To put it simply, this model performs simultaneous configurations of the same hyperparameters over multiple datasets and employs the gained information in all subsequent tuning problems. Auto-tuned models (ATM) [22] is a distributed and collaborative system that automates hyperparameter tuning and classification model selection procedures. At its core, ATM utilizes the conditional parameter tree (CPT), in which a learning method is placed at the root, and its children are the method's hyperparameters to represent the hyperparameter search space. Different tunable subsets of hyperparameter nodes in the CPT are selected during model selection and assigned to a cluster of workers to be configured. Koch et al. [23] introduced autotune as a derivative-free hyperparameter optimization framework. Composed of a hybrid and extendable set of solvers, this framework concurrently runs various searching methods, potentially distributed over a set of workers, to evaluate objective functions and provide feedback to the solvers. Autotune employs an iterative process during which all of the points that have already been evaluated are exchanged with the solvers to generate new sets of points to evaluate. In learning-based settings, the work reported in [24] used multi-agent reinforcement learning (MARL) to optimize the hyperparameters of deep convolutional neural networks (CNN). The suggested model splits the design space into sub-spaces and devotes each agent to tuning the hyperparameters of a single network layer using Q-learning. Parker-Holder et al. [25] presented the population-based bandit (PB2) algorithm, which efficiently directs the searching operation of hyperparameters in reinforcement learning using a probabilistic model. In PB2, a population of agents is trained in parallel, and their performance is monitored on a regular basis. An underperforming agent's network weights are replaced with those of a better-performing agent, and its hyperparameters are tuned using Bayesian optimization.

In continuation of our recent generic collaborative optimization model [20], this paper delves into the design of a multi-level agent-based distributed random search technique that can be used for both hyperparameter tuning and general purpose black-box function optimization. The proposed method, at its core, forms a tree-like structure comprising a set of interacting agents that, depending on their position in the hierarchy, focus on tuning/optimizing a single hyperparameter/decision variable using a biased hyper-cube random sampling technique or aggregating the results and facilitating collaborations based on the gained experience of other agents. The rationales behind choosing random search as the core tuning strategy of the agents include, but are not limited to, its intrinsic distributability, acceptable performance in practice, and it does not require differentiable objective functions. Although the parent model in [20] does not impose any restrictions on the state and capabilities of the agents, this paper assumes homogeneity in the sense that the tuner/optimizer agents use the same mechanism for their assigned job. With that said, the proposed method is analyzed in terms of its design parameters, and the empirical results from the conducted ML classification and regression tasks, as well as various multi-dimensional function optimization problems, demonstrate that the suggested approach not only outperforms the underlying random search methodologies under the same deployment conditions, but also provides a better-distributed solution in the presence of limited computational resources.

The remainder of this paper is organized as follows: Section 2 dissects the proposed agent-based random search method; Section 3 presents the details of used experimental ML and function optimization settings and discusses the performance of the proposed model

under different scenarios; and finally, Section 4 concludes the paper and provides future work suggestions.

2. Methodology

This section dissects the proposed agent-based hyperparameter tuning and black-box function optimization approaches. To help with a clear understanding of the proposed algorithms, this section begins by providing the preliminaries and introducing the key concepts, then presents the details of the agent-based randomized search algorithms accompanied by hands-on examples whenever needed.

2.1. Preliminaries

An agent, as the first-class entity in the proposed approach, might play different roles depending on its position in the system. As stated before, this paper uses hierarchical structures to coordinate the agents in the system, and hence, it defines two agent types: (1) *internals*, which play the role of result aggregators and collaboration facilitators in connection with their subordinates; and (2) *terminals*, which, implementing a single-variable randomized searching algorithm, are the actual searchers/optimizers positioned at the bottom-most level of the hierarchy. Assuming G to be the set of all agents in the system and the root of the hierarchy to be at level 0, this paper uses $g_{\lambda_i}^l$ ($G_{\lambda_j}^l$) to denote the agent (the set of agents) at level l of the hierarchy that are specialized in tuning hyperparameter λ_i (hyperparameter set λ_j), respectively, where $\lambda_j \subseteq \lambda$ and $g_{\lambda_i}^{l+1} \in G_{\lambda_j}^l$ iff. $\lambda_i \in \lambda_j$.

As denoted above, the hyperparameters that the agents represent determine their position in the hierarchy. Let $\mathcal{A}_{\lambda=\{\lambda_1, \lambda_2, \dots, \lambda_n\}}$ be the ML algorithms for which we intend to tune the hyperparameters. As the tuning process might not target the entire hyperparameter set of the algorithm, the proposed method divides the set into two *objective* and *fixed* disjoint subsets which, respectively denoted by λ_o and λ_f refer to the hyperparameter sets that we intend to tune and the ones we need to keep fixed. Formally, that is $\lambda = \lambda_o \cup \lambda_f$ and $\lambda_o \cap \lambda_f = \emptyset$. The paper further assumes two types of objective hyperparameters: (1) *primary* hyperparameters denoted by $\hat{\lambda}_o$, which comprise the main targets of the corresponding tuners (agents); and (2) *subsidiary* hyperparameters denoted by $\hat{\lambda}'_o$, which include the ones whose values are set by the other agents to help limit the searching space. These two sets are complements of each other, i.e., $\hat{\lambda}'_o = \lambda_o - \hat{\lambda}_o$, and the skill of an agent is determined by the primary objective set $\hat{\lambda}_o$, that it represents. With that said, for all terminal agents in the hierarchy, we have $|\hat{\lambda}_o| = 1$, where $|\dots|$ denotes the set cardinality.

The agents of a realistic MAS are susceptible to various limitations that are imposed by their environment and/or computational resources. This paper, due to its focus on the decentralization of the searching process, foresees two limitations for the agents: (1) the maximum number of concurrent connections, denoted by c that an agent can manage; (2) the number of concurrent processes, called *budget* and denoted by b that the agent can execute and handle. In the proposed method, $c > 1$ determines the maximum number of subordinates (children) that an internal agent can have. However, the budget $b \geq 1$ puts a restriction on the maximum number of parallel evaluations that an agent can perform in the step of searching for the optima.

Communications play a critical role in all MAS, including the agent-based method proposed in this paper. For all intra-systems, i.e., between any two agents, and inter-systems, i.e., between an agent and a user's interactions, the suggested method uses the following tuple-based structure for the queries:

$$\langle \mathcal{A}_\lambda, \{\hat{\lambda}_o, \hat{\lambda}'_o, \lambda_f\}, \mathcal{V}, \{\mathcal{X}^{(train)}, \mathcal{X}^{(valid)}\}, \mathcal{L} \rangle \tag{3}$$

where $\mathcal{V} = \{(\lambda_i, v_i)\}_{1 \leq i \leq n}$ denotes the set containing the candidate values for all hyperparameters, and the remaining notations are as defined in Equation (1). Based on what was discussed before, it is clear that $|\lambda_f| \leq |\mathcal{V}| \leq \lambda$.

2.2. Agent-Based Randomized Searching Algorithm

The high-level abstract view of the proposed approach is composed of two major steps: (1) distributedly building the hierarchical MAS; and (2) performing the collaborative searching process through vertical communications in the hierarchy. The sections that follow go into greater detail about these two stages.

2.2.1. Distributed Hierarchy Formation

As for the first step, each agent t divides the primary objective hyperparameter set of the query it receives, i.e., $\hat{\lambda}_o$, into a $c_t > 1$ number of subsets, for each of which the system initiates a new agent to handle. This process continues recursively until there is only one hyperparameter in the primary objective set, i.e., $|\hat{\lambda}_o| = 1$, which is assigned to a terminal agent. Figure 1 provides an example hierarchy resulting from the recursive division of the primary objective set $\lambda_o = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6\}$. For the sake of clarity, we have used the indexes of the hyperparameters as the labels of the nodes in the hierarchy, and the green and orange colors are employed to highlight the members of the $\hat{\lambda}_o$ and $\hat{\lambda}'_o$ sets, respectively. Regarding the maximum number of concurrent connections that the agents can handle in this example, it is assumed for all agents that $c = 2$, except for the rightmost agent in the second level of the hierarchy, for which $c = 3$. It is worth emphasizing that at the beginning of the process, when the tuning query is received from the user, we have $\hat{\lambda}_o = \lambda_o$ and $\hat{\lambda}'_o = \emptyset$, which is the reason for the all-green node of the root node in this example.

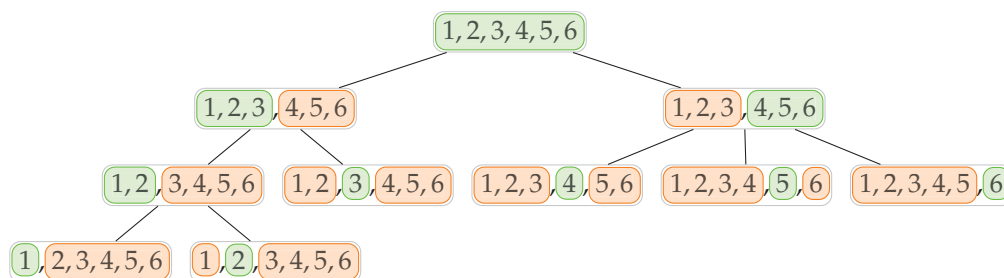


Figure 1. Hierarchical structure built for $\lambda_o = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6\}$, where the primary and complementary hyperparameters of each node are, respectively, highlighted in green and orange, and the labels are the indexes of λ_j .

Algorithm 1 presents the details of the process. We have chosen self-explanatory names for the functions and variables and provided comments wherever they are required to improve clarity. In this algorithm, the function PREPARERESOURCES in line 3 prepares the data and computational resources for the newly built/assigned terminal agent. Such resources are used for training, validation, and tuning processes. The function SPAWNORCONNECT in line 8 creates a subordinate agent that represents the ML algorithm \mathcal{A}_λ and expected loss function \mathcal{L} . This is achieved by either creating a new agent or connecting to an existing idle one if the resources are reused. Two functions, PREPAREFEEDBACK and TUNE in lines 17 and 18, respectively, are called when the structure formation process is over and the root agent initiates the tuning process in the hierarchy. Later, these two functions are discussed in more detail.

Algorithm 1: Distributed formation of the hierarchical agent-based hyperparameter tuning structure.

```

1 Function START( $\langle \mathcal{A}_\lambda, \{\hat{\lambda}_o, \hat{\lambda}'_o, \lambda_f\}, \mathcal{V}, \{\mathcal{X}^{(train)}, \mathcal{X}^{(valid)}\}, \mathcal{L} \rangle$ ):
2   if  $|\hat{\lambda}_o| = 1$  then ▷ agent is terminal
3      $\mathcal{R} \leftarrow \text{PREPARERESOURCES}(\langle \{\hat{\lambda}_o, \hat{\lambda}'_o, \lambda_f\}, \{\mathcal{X}^{(train)}, \mathcal{X}^{(valid)}\} \rangle)$ 
4      $\text{INFORM}(\text{Parent}, \mathcal{R})$  ▷ informs the parent agent
5   else ▷ agent is internal ( $|\hat{\lambda}_o| > 1$ )
6      $k \leftarrow \min(c_{my}, |\hat{\lambda}_o|)$  ▷ the number of children
7     for  $i \leftarrow 1$  to  $k$  do
8        $G_i \leftarrow \text{SPAWNORCONNECT}(\mathcal{A}_\lambda, \mathcal{L})$ 
9        $\hat{\lambda}_{o_i} \leftarrow \text{DIVIDE}(\hat{\lambda}_o, i, k)$  ▷ the  $i^{\text{th}}$  unique division
10       $\hat{\lambda}'_{o_i} \leftarrow (\hat{\lambda}_o - \hat{\lambda}_{o_i}) \cup \hat{\lambda}'_o$ 
11       $\mathcal{R}_i \leftarrow \text{ASK}(G_i, \text{START},$ 
12         $\langle \mathcal{A}_\lambda, \{\hat{\lambda}_{o_i}, \hat{\lambda}'_{o_i}, \lambda_f\}, \mathcal{V}, \{\mathcal{X}^{(train)}, \mathcal{X}^{(valid)}\}, \mathcal{L} \rangle)$ 
13      end
14       $\mathcal{R} \leftarrow \text{AGGREGATE}(\{\mathcal{R}_i\}_i)$  ▷ combines children's answers
15      if  $\text{Parent} \neq \emptyset$  then
16         $\text{INFORM}(\text{Parent}, \mathcal{R})$ 
17      else
18         $\mathcal{F} \leftarrow \text{PREPAREFEEDBACK}(\mathcal{R}, \mathcal{V})$ 
19         $\text{TUNE}(\mathcal{F})$  ▷ initiates the tuning process
20      end
21 end

```

2.2.2. Collaborative Tuning Process

The collaborative tuning process is conducted through a series of vertical communications in the built hierarchy. Initiated by the root agent, as explained in the previous section, the TUNE request is propagated to all of the agents in the hierarchy. As for the internal agents, the request will be simply passed down to the subordinates as they arrive. As for the terminal agents, moreover, the request launches the searching process in the sub-space specified by the parent. The flow of the results will be in an upward direction with a slightly different mechanism. As soon as a local optimum is found by a terminal agent, it will be sent up to the parent agent. Having waited for the results to be collected from all of its subordinates, the parent aggregates them together and passes the combined result to its own parent. This process continues until it reaches the root agent, where the new search guidelines are composed for the next search round.

Algorithm 2 presents the details of the iterated collaborative tuning process, which might be called by both terminal and internal agents. When it is called by a terminal agent, it initiates the searching operation for the optima of the hyperparameter that the agent represents and informs the result to its parent. Let $g_{\lambda_j}^l$ be the terminal agent concentrating on tuning hyperparameter λ_j . As it can be seen in line 3, the result of the search will be a single-item set composed of the identifier of the hyperparameter, i.e., λ_j , the set $\mathcal{V}_j^{(*)}$ containing the coordinates of the best candidate agent $g_{\lambda_j}^l$ has been found, and the response function value for that best candidate is, i.e., $\Psi_j^{(*)}$. An internal agent running this procedure merely passes the tuning request to the subordinates and waits for their search results (line 7 of the algorithm). Please note that this asking operation comprises a filtering operation on set \mathcal{F} . That is, a subordinate will receive a subset $\mathcal{F}_i \subset \mathcal{F}$ that only includes the starting coordinates for the terminal agents that are reachable through that agent. Having collected all of the results from its subordinates, the internal agent aggregates them by

simply joining the result sets and informing its own parent, in case it is not the root agent. This process is executed recursively until the aggregated results reach the root agent of the hierarchy. Depending on whether the stopping criteria of the algorithm are reached, the root prepares feedback to initiate the next tuning iteration or a report detailing the results. The collaboration between the agents is conducted implicitly through the feedback that the root agent provides to each terminal agent based on the results it has gathered in the previous iteration. As presented in line 17 of the algorithm, this feedback basically determines the coordinates of the position where the terminal agents should start their searching operation. It should be noted that the `argmin` function in this operation is due to employing the loss function \mathcal{L} as a metric to evaluate the performance of an ML model. For performance measures in which maximization is preferred, such as in *accuracy*, this operation needs to be replaced by `argmax` accordingly.

Algorithm 2: Iterated collaborative tuning procedure.

```

1 Function TUNE( $\mathcal{F}$ ):
2   if  $Children = \emptyset$  then ▷ terminal agent agent
3      $\{(\lambda_j, \mathcal{V}_j^{(*)}, \Psi_j^{(*)})\} \leftarrow \text{RUNTUNINGALGORITHM}(\mathcal{F} = \mathcal{V})$ 
4      $\text{INFORM}(\text{Parent}, \{(\lambda_j, \mathcal{V}_j^{(*)}, \Psi_j^{(*)})\})$ 
5   else
6     foreach  $G_i^{l+1} \in Children$  do
7        $\mathcal{R}_i^{(*)} \leftarrow \text{ASK}(G_i^{l+1}, \text{TUNE}, \mathcal{F}_i \subset \mathcal{F})$  ▷  $\mathcal{R}_i^{(*)} = \{(\lambda_k, \mathcal{V}_k^{(*)}, \Psi_k^{(*)})\}_k$ 
8     end
9      $\mathcal{R}^{(*)} \leftarrow \bigcup_{G_i^{l+1} \in Children} \mathcal{R}_i^{(*)}$  ▷ aggregates results
10    if  $\text{Parent} \neq \emptyset$  then ▷ non-root internal agent
11       $\text{INFORM}(\text{Parent}, \mathcal{R}^{(*)})$ 
12    else
13      if  $\text{SHOULDSTOP}(\text{StopCriteria}) \neq \text{True}$  then
14         $\mathcal{F} \leftarrow \left\{ (\lambda_i, \mathcal{V}_j); j = \arg \min_{1 \leq k \leq n} \Psi_k^{(*)} \right\}_{1 \leq i \leq n}$  ▷ prepares feedback
15         $\text{TUNE}(\mathcal{F})$  ▷ initiates next tuning iteration
16      else
17         $\text{PREPAREREPORT}(\mathcal{R}^{(*)})$  ▷ reports final result
18      end
19    end
20  end
21 end

```

The details of the tuning function that each terminal agent runs in line 3 of Algorithm 2 to tune a single hyperparameter are presented in Algorithm 3. As its input, this function receives a coordinate that agent $g_{\lambda_i}^l$ will use as its starting point in the searching process. The received argument, together with b additional coordinates that the agent generates randomly, are stored in the set of candidate \mathcal{C} . Accordingly, $\mathcal{C}[c]$ and $\mathcal{C}[c](\lambda_i)$ refer to the c -th coordinate in the set and the value assigned to the hyperparameter λ_i of that coordinate, respectively. Moreover, please recall from Section 2.1 that b denotes the evaluation budget of a terminal agent. The terminal agents in the proposed method employ slot-based uniform random sampling to explore the search space. Formally, let $\mathcal{E} = \{\epsilon_{\lambda_1}, \epsilon_{\lambda_2}, \dots, \epsilon_{\lambda_n}\}$ be a set of real values that each agent utilizes for each hyperparameter to control the size of slots in any iteration. Similarly, let $s = \{s_{\lambda_1}, s_{\lambda_2}, \dots, s_{\lambda_n}\}$ specify the coordinate of the position that an agent starts its searching operation in any iteration. To sample b random values in

the domain \mathbb{D}_{λ_j} of any arbitrary hyperparameter λ_j , the agent will generate one uniform random value in range

$$\mathcal{R} = \left[\max(\inf \mathbb{D}_{\lambda_j}, s_{\lambda_j} - \epsilon_{\lambda_j}), \min(\sup \mathbb{D}_{\lambda_j}, s_{\lambda_j} + \epsilon_{\lambda_j}) \right] \quad (4)$$

and $b - 1$ random values in $\mathbb{D}_{\lambda_j} - \mathcal{R}$ by splitting it into $b - 1$ slots and choosing one uniform random value in each slot (lines 6 and 8 of the algorithm). The generation of the uniform random values is achieved by calling the function `UNIFORMRAND(A_1, A_2, A_3)`. This function divides range A_1 into A_2 equal-sized slots and returns the uniform random value generated in the A_3 -th slot. As it can be seen in line 12 of the algorithm, the agent employs the same function to generate one and only one value per each element in its subsidiary objective hyperparameter set $\hat{\lambda}'_o$.

The slot width parameter set \mathcal{E} is used to control the exploration behavior of the agent around the starting coordinates in the search space. For instance, for any arbitrary hyperparameter λ_i , very small values of ϵ_{λ_i} emphasize generating candidates in the close vicinity of the starting position. Moreover, larger values of ϵ_{λ_i} decrease the chance that the generated candidate will be close to the starting position. In the proposed method, the agents adjust \mathcal{E} adaptively. To put it formally, each agent starts the tuning process with the pre-specified value set $\mathcal{E}^{(0)}$, and assuming that $\mathcal{C}^{(*)}$ denotes the best candidate that the agent has found in the previous iteration, the width parameter set \mathcal{E} in iteration i is updated as follows:

$$\mathcal{E}^{(i)} = \begin{cases} \Delta \odot \mathcal{E}^{(i-1)} & \text{If } \mathcal{V} = \mathcal{C}^{(*)} \\ \mathcal{E}^{(i-1)} & \text{otherwise} \end{cases} \quad (5)$$

where $\Delta = \{\delta_{\lambda_1}, \delta_{\lambda_2}, \dots, \delta_{\lambda_n}\}$ denotes the scaling changes to apply to the width parameters, and \odot denotes the element-wise multiplication operator. As the paper discusses in Section 3, despite the generic definitions provided here for futuristic extensions, using the same scaling value for all primary hyperparameters has led to satisfactory results in our experiments.

Algorithm 3: A terminal agent’s randomized tuning process.

```

1 Function RUNTUNINGALGORITHM( $\mathcal{F} = \mathcal{V} = \{(\lambda_m, v_m)\}_{1 \leq m \leq n}$ ):
2    $\mathcal{C}[0] \leftarrow \mathcal{V}$ 
3    $\mathcal{R}_{\lambda_i} \leftarrow [\max(\inf \mathbb{D}_{\lambda_i}, v_i - \epsilon_{\lambda_i}), \min(\sup \mathbb{D}_{\lambda_i}, v_i + \epsilon_{\lambda_i})]$ 
4   for  $c \leftarrow 1$  to  $c = b$  do
5     if  $c = 1$  then ▷ the first sample for  $\hat{\lambda}_o = \{\lambda_i\}$ 
6        $\mathcal{C}[c](\lambda_i) \leftarrow \text{UNIFORMRAND}(\mathcal{R}_{\lambda_i}, 1, 1)$ 
7     else ▷ the remaining samples for  $\hat{\lambda}_o = \{\lambda_i\}$ 
8        $\mathcal{C}[c](\lambda_i) \leftarrow \text{UNIFORMRAND}(\mathbb{D}_{\lambda_i} - \mathcal{R}_{\lambda_i}, b - 1, c - 1)$ 
9     end
10    forall  $\lambda_k \in \hat{\lambda}'_o$  do
11       $\mathcal{R}_{\lambda_k} \leftarrow [\max(\inf \mathbb{D}_{\lambda_k}, v_k - \epsilon_{\lambda_k}), \min(\sup \mathbb{D}_{\lambda_k}, v_k + \epsilon_{\lambda_k})]$ 
12       $\mathcal{C}[c](\lambda_k) \leftarrow \text{UNIFORMRAND}(\mathcal{R}_{\lambda_k}, 1, 1)$ 
13    end
14  end
15   $\mathcal{C}^{(*)} \leftarrow \arg \min_{0 \leq j \leq b} \Psi(\mathcal{C}[j])$ 
16  return  $\{(\lambda_i, \mathcal{C}^{(*)}, \Psi(\mathcal{C}^{(*)}))\}$ 
17 end

```

To better understand the suggested collaborative randomized tuning process of agents, an illustrative example is depicted in Figure 2. In this figure, each agent is represented by a different color, and the best candidate that each agent finds at the end of each iteration is

shown by a filled shape. Moreover, we have assumed that the value of the loss function becomes smaller as we move inwards in the depicted contour lines, and to prevent any exploration in the domain of the subsidiary hyperparameters, we have set $\mathcal{E} = \{\epsilon_{\lambda_1} = \frac{1}{6}, \epsilon_{\lambda_2} = 0\}$ and $\mathcal{E} = \{\epsilon_{\lambda_1} = 0, \epsilon_{\lambda_2} = \frac{1}{6}\}$ for agents $g_{\lambda_1}^1$ and $g_{\lambda_2}^1$, respectively, assuming that the domain size of each hyperparameter is 1 and $b = 3$. In Iteration 1, both agents start at the top right corner of the search space and are able to find candidates that yield lower loss function values than the starting coordinate. For iteration 2, the starting coordinate of each agent is set to the coordinate of the best candidate found by all agents in the previous iteration. As the best candidate was found by agent $g_{\lambda_2}^1$, we only see the change in the searching direction of the red agent, i.e., $g_{\lambda_2}^1$. The winner agent at the end of this iteration is agent $g_{\lambda_1}^1$; hence, we do not see any change to its searching direction in iteration 3. Please note that the four circles for agent $g_{\lambda_1}^1$ in the last depicted iteration is because it shows the starting coordinate, which happens to remain the best candidate in this iteration. It is also worth emphasizing that the starting coordinates are not evaluated again by the agents, as they have already been accompanied by their corresponding response values from the previous iterations.

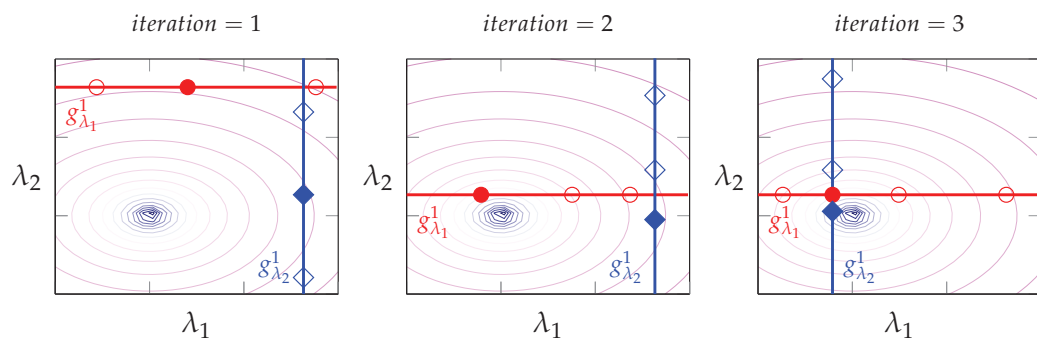


Figure 2. A toy example demonstrating three iterations of running the proposed method for tuning two hyperparameters λ_1 and λ_2 using terminal agents $g_{\lambda_1}^1$ and $g_{\lambda_2}^1$, respectively. It is assumed that for each agent, $b = 3$.

3. Results and Discussion

This section dissects the performance of the proposed method in more detail. It begins with the computational complexity of the technique and then provides empirical results on both machine learning and general function optimization tasks.

3.1. Computational Complexity

Forming the hierarchical structure and conducting the collaborative searching process are the two major stages of the proposed method and these stages need to be conducted in sequence. The rest of this section investigates the complexity of each step separately and in relation to one another.

Regarding the structural formation phase of the suggested method, the shape of the hierarchy depends on the maximum number of connections that each agent can handle; the fewer the number of manageable concurrent connections, the deeper the resulting hierarchy. Using the same notations presented in Section 2.1 and assuming the same $c > 1$ for all agents, the depth of the formed hierarchy is $\lceil \log_c |\lambda_o| \rceil$. Thanks to the distributed nature of the formation algorithm and the concurrent execution of the agents, the worst-case time complexity of the first stage will be $\mathcal{O}(\log_c |\lambda_o|)$. With the same assumption, it can be easily shown that the resulting hierarchical structure is a complete tree. Hence, denoting the total number of agents in the system by \mathbb{G} , this quantity would be:

$$\frac{c^{\lceil \log_c |\lambda_o| \rceil} - 1}{c - 1} < \mathbb{G} \leq \frac{c^{\lceil \log_c |\lambda_o| \rceil + 1} - 1}{c - 1} \tag{6}$$

With that said, the space complexity for the first phase of the proposed technique would be $\mathcal{O}\left(\frac{c^{\lceil \log_c |\lambda_o| \rceil + 1} - 1}{c - 1}\right) = \mathcal{O}(|\lambda_o|)$. It is worth noting that among all created agents, only $|\lambda_o|$ terminal agents would require dedicated computational resources as they are completing the actual searching and optimization process, and the remaining $\mathbb{G} - |\lambda_o|$ can all be hosted and managed together.

The procedures in each round of the second phase of the suggested method can be broken down into two main components: (i) transmitting the start coordinates from the root of the hierarchy to the terminal agents, transmitting the results back to the root, and preparing the feedback; and (ii) conducting the actual searching process by the terminal agents to locate a local optimum. The worst-case time complexity of preparing the feedback based on the algorithms that were discussed in Section 2 would be $\mathcal{O}(|\lambda_o|)$, which is because it finds the best candidate among all returned results. In addition, due to the concurrency of the agents, the first component is only processed at the height of the built structure. Therefore, the time complexity of component (i) would be $\mathcal{O}(|\lambda_o| + \log_c |\lambda_o|) = \mathcal{O}(|\lambda_o|)$. The complexity of the second component, moreover, depends on both the budget of the agent, i.e., b , and the complexity of building and evaluating response function Ψ . Let $\mathcal{O}(\mathcal{R})$ denote the time complexity of a single evaluation. As a terminal agent makes a b number of such evaluations to choose its candidate optima, the time complexity for the agent would be $\mathcal{O}(b\mathcal{R})$. As all agents work in parallel, the complexity of a single iteration at the terminal agents would be $\mathcal{O}(b\mathcal{R})$, leading to the overall time complexity of $\mathcal{O}(|\lambda_o| + b\mathcal{R})$. In machine learning problems, we often have $\mathcal{O}(|\lambda_o|) \ll \mathcal{O}(\mathcal{R})$. Therefore, if \mathcal{I} denotes the number of iterations until the second phase of the tuning method stops, the complexity of the second stage would be $\mathcal{O}(\mathcal{I}b\mathcal{R})$. The space complexity of the second phase of the tuning method depends on the way that each agent is implementing the main functionalities, such as the learning algorithms they represent, transmitting the coordinates, and providing feedback. Except for the ML algorithms, all internal functionalities of each agent can be implemented using $\mathcal{O}(|\lambda_o|)$ space. Moreover, we have \mathbb{G} agents in the system, which leads to a total space complexity of $\mathcal{O}(|\lambda_o|^2)$ for non-ML tasks. Let $\mathcal{O}(\mathcal{S})$ denote the worst-case space complexity of a machine learning algorithm that we are tuning. The total space complexity of the second phase of the proposed tuning method would be $\mathcal{O}(|\lambda_o|^2 + \mathcal{S})$. Similar to the time complexity, in machine learning, we often have $\mathcal{O}(|\lambda_o|) \ll \mathcal{O}(\mathcal{S})$, which makes the total space complexity of the second phase $\mathcal{O}(\mathcal{S})$. Please note that we have factored out the budgets of the agents and the number of iterations because we did not store the history between different evaluations and iterations.

Considering both stages of the proposed technique and due to the fact that they are conducted in sequence, the time complexity of the entire steps in an ML hyperparameter tuning problem, from structure formation to completing the searching operations, would be $\mathcal{O}(\log_c |\lambda_o| + \mathcal{I}b\mathcal{R}) = \mathcal{O}(\mathcal{I}b\mathcal{R})$. Similarly, the space complexity would be $\mathcal{O}(|\lambda_o| + \mathcal{S}) = \mathcal{O}(\mathcal{S})$.

3.2. Empirical Results

This section presents the empirical results of employing the proposed agent-based randomized searching algorithm and discusses the improvements resulting from the suggested inter-agent collaborations. Hyperparameter tuning in machine learning is basically a back-box optimization problem, and hence, to enrich our empirical discussions, this section also includes results from multiple multi-dimensional optimization problems.

The performance metrics used for the experiments are based on those that are commonly used by the ML and optimization communities. Additionally, we analyze the behavior of the suggested methodology based on its own design parameter values, such as budget, width, etc. The methods that have been chosen for the sake of comparison are the standard random search and the Latin hypercube search methods [1] that are commonly used in practice. Our choices are based on the fact that not only are these methods embarrassingly parallel and among the top choices to be considered in distributed scenarios, but they are also used as the core optimization mechanisms of the terminal agents in the suggested

method, and hence can better present the impact of the inter-agent collaborations. In its generic format, as emphasized in [20], one can easily employ alternative searching methods or diversify them at the terminal level, as needed.

Throughout the experiments, each terminal agent runs on a separate process, and to make the comparisons fair, we keep the number of model/function evaluations fixed among all of the experimented methods. To put it in more detail, for a budget value of b for each of $|\lambda_o|$ terminal agents and \mathcal{I} number of iterations, the proposed method will evaluate the search space in $b \times \mathcal{I}$ coordinates. We use the same $|\lambda_o|$ number of independent agents for the compared random-based methodologies and, keeping the evaluation budgets of the agents fixed—the budgets are assumed to be enforced by the computational limitations of the devices or processes running the agents—we repeat those methods \mathcal{I} times and report the best performance among all agents' repetition histories as their final result.

The experiments assess the performance of the proposed method in comparison to the other random-based techniques in four categories: (1) iteration-based assessment, which checks the performance of the methods for a particular iteration threshold. In this category, all other parameters, such as budget, connection number, etc., are kept fixed; (2) budget-based assessment, which examines the performance under various evaluation budgets for the terminal agents. It is assumed that all agents have the same budget; (3) width-based assessment, which checks how the proposed method performs for various exploration criteria specified by the slot width parameter; and finally, (4) connection-based evaluation, which inspects the effect of the parallel connection numbers that the internal agents can handle. In other words, this evaluation checks if the proposed method is sensitive to the way that the hyperparameter or decision variables are split during the hierarchy formation phase. All implementations use Python 3.9 and the scikit-learn library [26], and the results reported in all experiments are based on 50 different trials.

For the ML hyperparameter tuning experiments, we have dissected the behavior of the proposed algorithm in two classifications and two regression problems. The details of such problems, including the hyperparameters that are tuned and the used datasets are presented in Table 1. In all of the ML experiments, we have used five-fold cross-validation as the model evaluation method. The results obtained for the classification and regression problems are plotted in Figure 3 and Figure 4, respectively. Please note that there are numerous ML algorithms that can be used to evaluate our approach. Our selected algorithms are representative of different types of classifiers/regressors, including linear and non-linear models with different regularization methods, and we found them widely used in hyperparameter tuning literature based on their performance sensitivity to the choice of hyperparameter values. We also experienced this empirically during our evaluations of some other ML algorithms. We found that all of the compared models converged to a local optimum point quickly, potentially due to the geometry of their response functions, which would not demonstrate the improvements of our model. By comparing the performance of the presented methods on these models, we hope to draw more general conclusions about the effectiveness of the methods in various settings.

For the *iterations* plot in the first column plots of Figures 3 and 4, we fixed the parameters of the proposed method for all agents as follows: $b = 3, \mathcal{E} = 2^{-6}, c = 2, \Delta = \{2, 2, \dots, 2\}$. As can be seen, when the proposed method is allowed to run for more iterations, it yields better performance, and its superiority against the other two random-based methods is evident. Comparing the relative performance improvements resulting from the proposed method in the presented ML tasks, it can be seen that as the search space of the agents and the number of hyperparameters needed to be tuned increased, the proposed collaborative method achieved a higher improvement. For the Stochastic Gradient Descent (SGD) classifier, for instance, the objective hyperparameter set comprises six members with continuous domain spaces, and the number of improvements that have been made after 10 iterations is much higher, about 17%, than in the other experiments with three to four hyperparameters and mixed continuous and discrete domain spaces.

Table 1. The details of the machine learning algorithms and the datasets used for hyperparameter tuning experiments.

ML Algorithm	λ_o	Dataset	Performance Metric
C-Support Vector Classification (SVC) [26,27]	$\{c, \gamma, \text{kernel}\}^1$	artificial (100,20) [†]	accuracy
Stochastic Gradient Descent (SGD) Classifier [26]	$\{\alpha, \text{l1_ratio}, \text{tol}, \epsilon, \eta_0, \text{val_frac}\}^2$	artificial (500,20) [†]	accuracy
Passive Aggressive Regressor [26,28]	$\{c, \text{tol}, \epsilon, \text{val_frac}\}^3$	artificial (300,100) [‡]	mean squared error
Elastic Net Regressor [26,29]	$\{\alpha, \text{l1_ratio}, \text{tol}, \text{selection}\}^4$	artificial (300,100) [‡]	mean squared error

¹ $c \sim \log \text{Uniform}(10^{-2}, 10^{13})$, $\gamma \sim \text{Uniform}(0, 1)$, $\text{kernel} \in \{\text{poly}, \text{linear}, \text{rbf}, \text{sigmoid}\}$. ² $\alpha \sim \text{Uniform}(0, 10^3)$, $\text{l1_ratio} \sim \text{Uniform}(0, 1)$, $\text{tolerance} \sim \text{Uniform}(0, 10^3)$, $\epsilon \sim \text{Uniform}(0, 10^3)$, $\eta_0 \sim \text{Uniform}(0, 10^3)$, $\text{validation_fraction} \sim \text{Uniform}(0, 1)$. ³ $c \sim \text{Uniform}(0, 10^3)$, $\text{tolerance} \sim \text{Uniform}(0, 10^3)$, $\text{validation_fraction} \sim \text{Uniform}(0, 1)$, $\epsilon \sim \text{Uniform}(0, 1)$. ⁴ $\alpha \sim \text{Uniform}(0, 1)$, $\text{l1_ratio} \sim \text{Uniform}(0, 1)$, $\text{tolerance} \sim \text{Uniform}(0, 1)$, $\text{selection} \in \{\text{cyclic}, \text{random}\}$. [†] An artificially generated binary classification dataset using scikit-learn’s `make_classification` function [30]. The first number represents the number of samples and the second figure is the number of features. [‡] An artificially generated regression dataset using scikit-learn’s `make_regression` function [30]. The first number represents the number of samples and the second figure is the number of features.

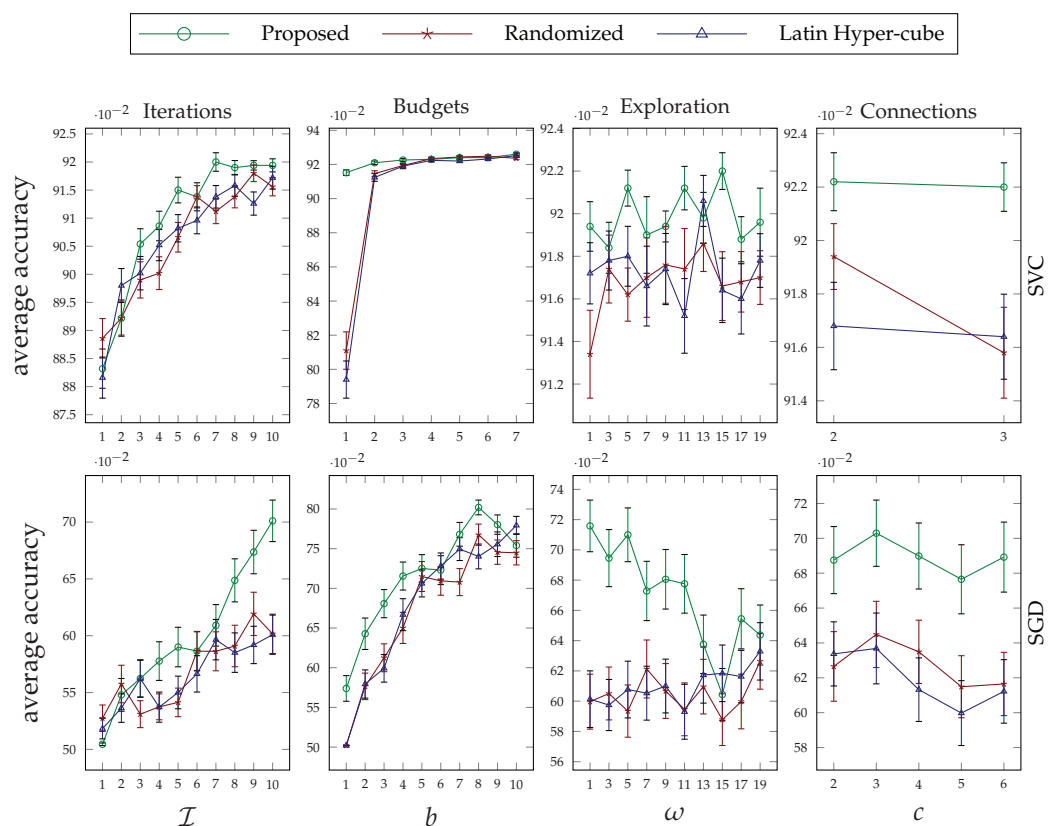


Figure 3. Average performance of the C-support vector classification (SVC) (first row) and stochastic gradient descent (SGD) (second row) classifiers on two synthetic classification datasets based on the accuracy measure. The error bars in each plot are calculated based on the standard error.

The second column of Figures 3 and 4 illustrate how the performance of the proposed technique changes when we increase the evaluation budgets of the terminal agents. For this set of experiments, we set the parameter values of our method as follows: $\mathcal{I} = 10$, $\mathcal{E} = 2^{-6}$, $c = 2$, $\Delta = \{2, 2, \dots, 2\}$. By increasing the budget value, the performance of the suggested approach per se improves. However, the rate of improvement slows down for higher budget values, and comparing it against the performance of the other two random-based searching methods, the improvement is significant for lower budget values. In other words, the proposed tuning method surpasses the other two methods when the agents have limited searching resources. This makes our method a good candidate for tuning the hyperparameters of deep learning approaches with expensive model evaluations.

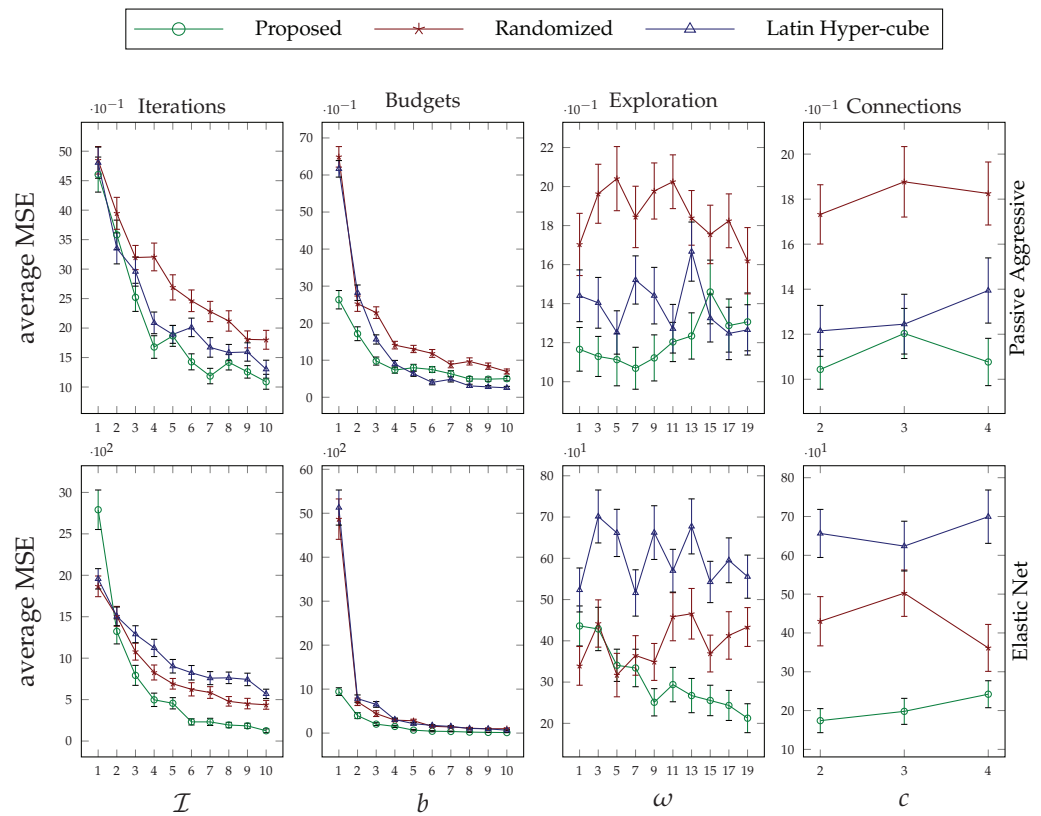


Figure 4. Average performance of the passive aggressive (first row) and elastic net (second row) regression algorithms on two synthetic regression datasets based on the mean squared error (MSE) measure. The error bars in each plot are calculated based on the standard error.

The behavior of the suggested method under various exploration parameter values can be seen in column 3 of Figures 3 and 4. The ω values on the x-axis of the plots are used to set the initial value for the slot width parameter of all agents using $\mathcal{E} = 2^{-\omega-1}$. Based on this configuration, higher values of ω yield lower values of \mathcal{E} , and as a result, there is more exploitation around the starting coordinates. The other parameters of the method are configured as follows: $\mathcal{I} = 10, b = 3, c = 2, \Delta = \{2, 2, \dots, 2\}$. Recall from Section 2 that the exploration parameter is used by an agent for the dimensions that it does not represent. Based on the results obtained from various tasks, choosing a proper value for this parameter depends on the characteristics of the response function. Having said that, the behavior for a particular task remains almost consistent. Hence, trying one small and one large value for this parameter in a specific problem will reveal its sensitivity and help choose an appropriate value for it.

Finally, the last set of experiments investigates the impact of the number of parallel connections that the internal agents can manage, i.e., c , on the performance of the suggested method. The results of this study are plotted in the last column of Figures 3 and 4. The difference in the number of data points in each plot is because of the difference in the size of the hyperparameters that we tune for each task. The values of the parameters that we kept fixed for this set of experiments are as follows: $\mathcal{I} = 10, b = 3, \mathcal{E} = 2^{-6}, \Delta = \{2, 2, \dots, 2\}$. As can be seen from the illustrated results, the proposed method is not very sensitive to the value that we choose or that is enforced by the system for parameter c . This parameter plays a critical role in the shape of the hierarchy that is distributedly formed in phase 1 of the suggested approach; therefore, one can opt to choose a value that fits with the connection or computational resources that are available without sacrificing performance very much.

As stated before, we have also studied the suggested technique for the black-box optimization problem to see how it performs in finding the optima of various convex

and non-convex functions. These experiments also help us to closely check the relative performance improvements in higher dimensions. We have chosen three non-convex benchmark optimization functions and a convex toy function, the details of which are presented in Table 2. For each function, we run the experiments in three different dimension sizes, and the goal of the optimization is to find the global minimum. Very similar to the settings that we discussed for ML hyperparameter tuning, whenever we mean to fix the value of each parameter value in different experiment sets, we use the following parameter values: $\mathcal{I} = 10$, $b = 3$, $\mathcal{E} = 2^{-10}$, $c = 2$, $\Delta = \{2, 2, \dots, 2\}$.

Table 2. The details of the multi-dimensional functions used for black-box optimization experiments.

Function	λ_o	Domain	$f(x^*)$
Hartmann, 3D, 4D, 6D [31]	$\{x_1, \dots, x_d\}, d \in \{3, 4, 6\}$	$x_i \in [0, 1]$	3D :−3.86278, 4D :−3.135474, 6D :−3.32237
Rastrigin, 3D, 6D, 10D [32]	$\{x_1, \dots, x_d\}, d \in \{3, 6, 10\}$	$x_i \in [-5.12, 5.12]$	3D :0, 6D :0, 10D :0
Styblinski–Tang, 3D, 6D, 10D [33]	$\{x_1, \dots, x_d\}, d \in \{3, 6, 10\}$	$x_i \in [-5, 5]$	3D :−117.4979, 6D :−234.9959, 10D :391.6599
Mean Average Error, 3D, 6D, 10D †	$\{x_1, \dots, x_d\}, d \in \{3, 6, 10\}$	$x_i \in [0, 100]$	3D :0, 6D :0, 10D :0 ‡

† This is a toy multi-dimensional MAE function that is defined as $f(x) = \frac{1}{n} \sum_{i=1}^n |x - \chi|$, where χ denotes a ground truth vector that is generated randomly in the domain space for each experiment. ‡ This is a convex function and the coordinate of its minimum value depends on the ground truth vector that is generated, i.e., when $x = \chi$.

The plots are grouped by functions and can be found in Figures 5–8. The conclusion that was drawn concerning the behavior of the proposed approach under different values of its design parameters applies to these optimization experiments as well. That is, the more the proposed method runs, the better performance it achieves; its superiority on low budget values is clear; its sensitivity to exploration parameter values is consistent; and the way that the decision variables are broken down during the formation of the hierarchy does not affect the performance very much. Furthermore, as can be seen in each group figure, the proposed algorithm yields a better minimum point in comparison to the other two random-based methods when the dimensionality of a function increases.

Disregarding its multi-agent formulation, autonomy, and inter-agent collaborations, the proposed method shares similarities with heuristic and population-based black-box optimization approaches. We believe that even with such a viewpoint, our method can be more applicable due to its simple architecture, low number of hyperparameters, its innate distribution, and because it requires less domain knowledge. Figures 9 and 10 provide a comparison between the performance of our agent-based method and the ones of particle swarm optimization (PSO) [34] and simulated annealing (SA) [35]. Please note that these comparisons are not to prove our method’s superiority over population-based and/or heuristic methods, but to give a glimpse into some additional behaviors and the potentiality of the agent-based solution. In its current immature condition, we do not doubt that our immature approach will most probably be outperformed by the many mature heuristic methods available.

For the PSO algorithm, we have employed the standard version and set its hyperparameter values as $c_1 = c_2 = 1.5$ and $\omega = 0.7$. As for the SA algorithm, we have used Kirkpatrick’s method [35] to define the accepting probabilities with $T_0 = 100$ and the geometric process for the annealing schedule, i.e., $T_k = T_0 \alpha^k$ with $\alpha = 0.95$. Please note that our choices for the aforementioned values are based on multiple trials and errors and the general practical suggestions found in the literature. Finally, the values that we have utilized for our agent-based solution are as follows: $c = 2$, $\mathcal{E} = 2^{-10}$, $\Delta = \{2, 2, \dots, 2\}$ and $\mathcal{I} = 10$, $b = 3$, whenever they are assumed fixed. Please note that these values are the same as the ones we applied in the previous set of analyses, and we have not conducted any optimization to choose the best possible values.

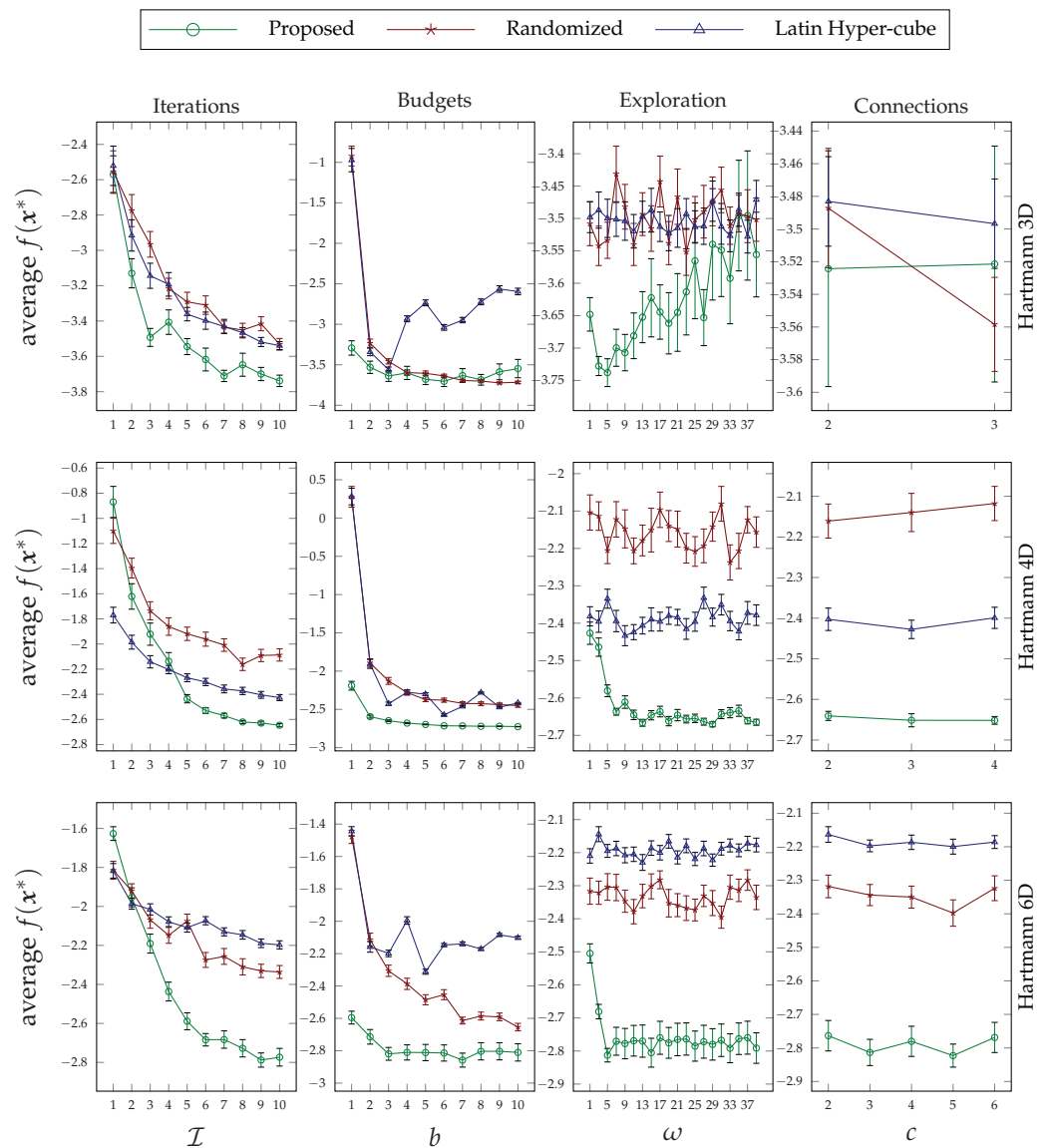


Figure 5. Average values of the Hartmann function optimized under variable iterations, budgets, explorations, and connection thresholds. Each row of the figure pertains to a particular dimension size, and the error bars are calculated based on the standard error.

Due to the different underlying principles used in each of these algorithms, providing an absolutely fair comparison would not be possible. For instance, in our method, the number of agents is fixed, and each agent has an evaluation budget. In the PSO algorithm, however, the population size is a hyperparameter, and each particle makes a single evaluation. The SA, moreover, is a single-agent, centralized approach with one evaluation in each of its iterations. To the best of our ability, in this empirical comparison, we have tried to keep the total number of evaluations fixed among all experiments. Strictly speaking, we set the same number of iterations, i.e., \mathcal{I} , in the PSO but set its population size to $b \times |\lambda_o|$. Similarly, in the SA, we set the number of iterations to $b \times |\lambda_o| \times \mathcal{I}$.

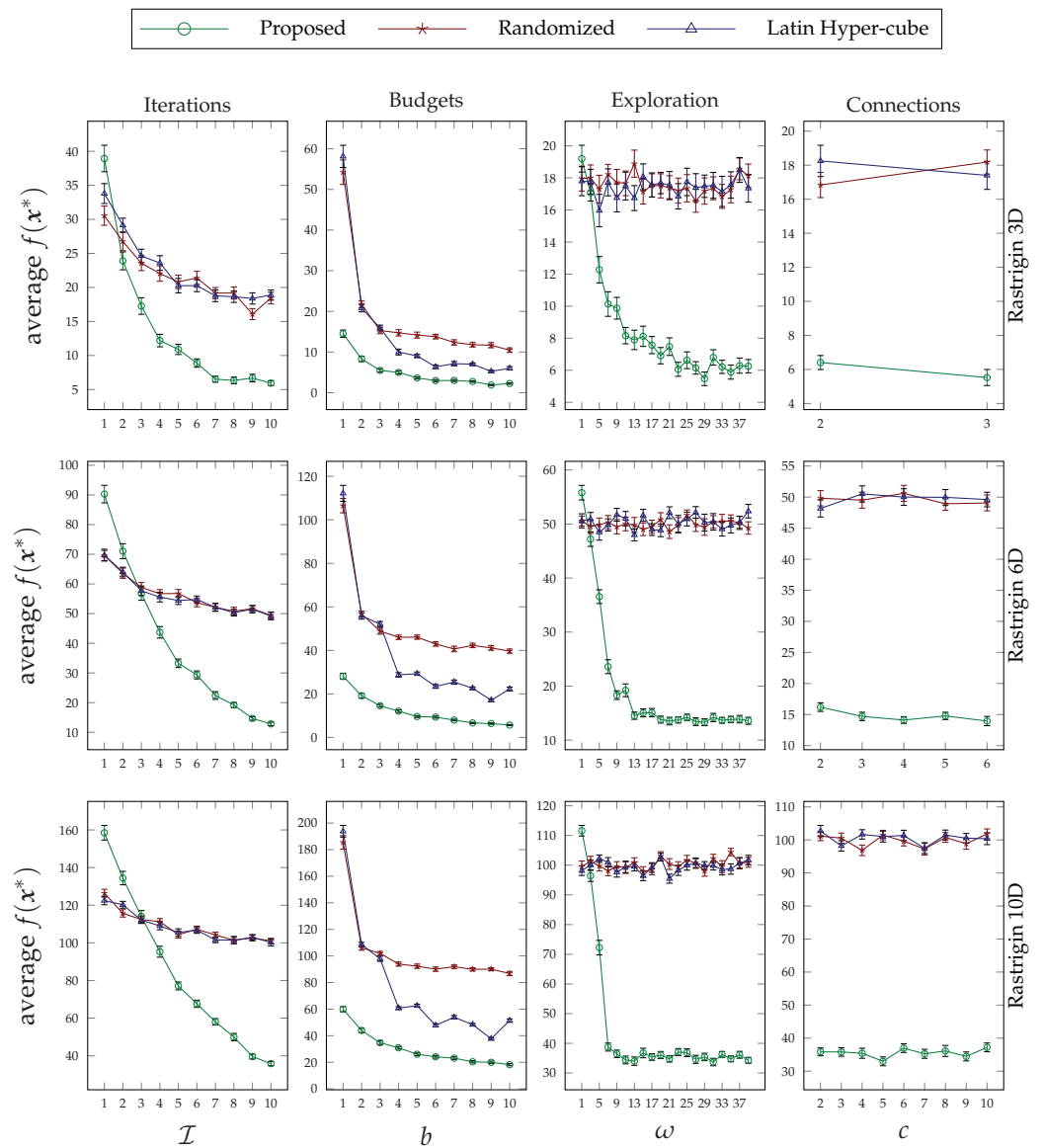


Figure 6. Average values of the Rastrigin function optimized under variable iterations, budgets, explorations, and connection thresholds. Each row of the figure pertains to a particular dimension size, and the error bars are calculated based on the standard error.

The results presented in Figure 9 show how each different method behaves under different numbers of iterations in each of the benchmark problems. As can be seen, in most benchmarks, the proposed method has outperformed both PSO and SA in higher iteration numbers. Recalling the true optimal function values from Table 2, the tied or close conditions among all methods happen near the global optima, which we believe can be improved through an adaptive exploitation method. Furthermore, due to the relatively higher improvements in the Rastrigin and Styblinski–Tang functions and the fact that these two functions are composed of several local optima, we can conclude that our proposed method has better capability to escape those local positions.

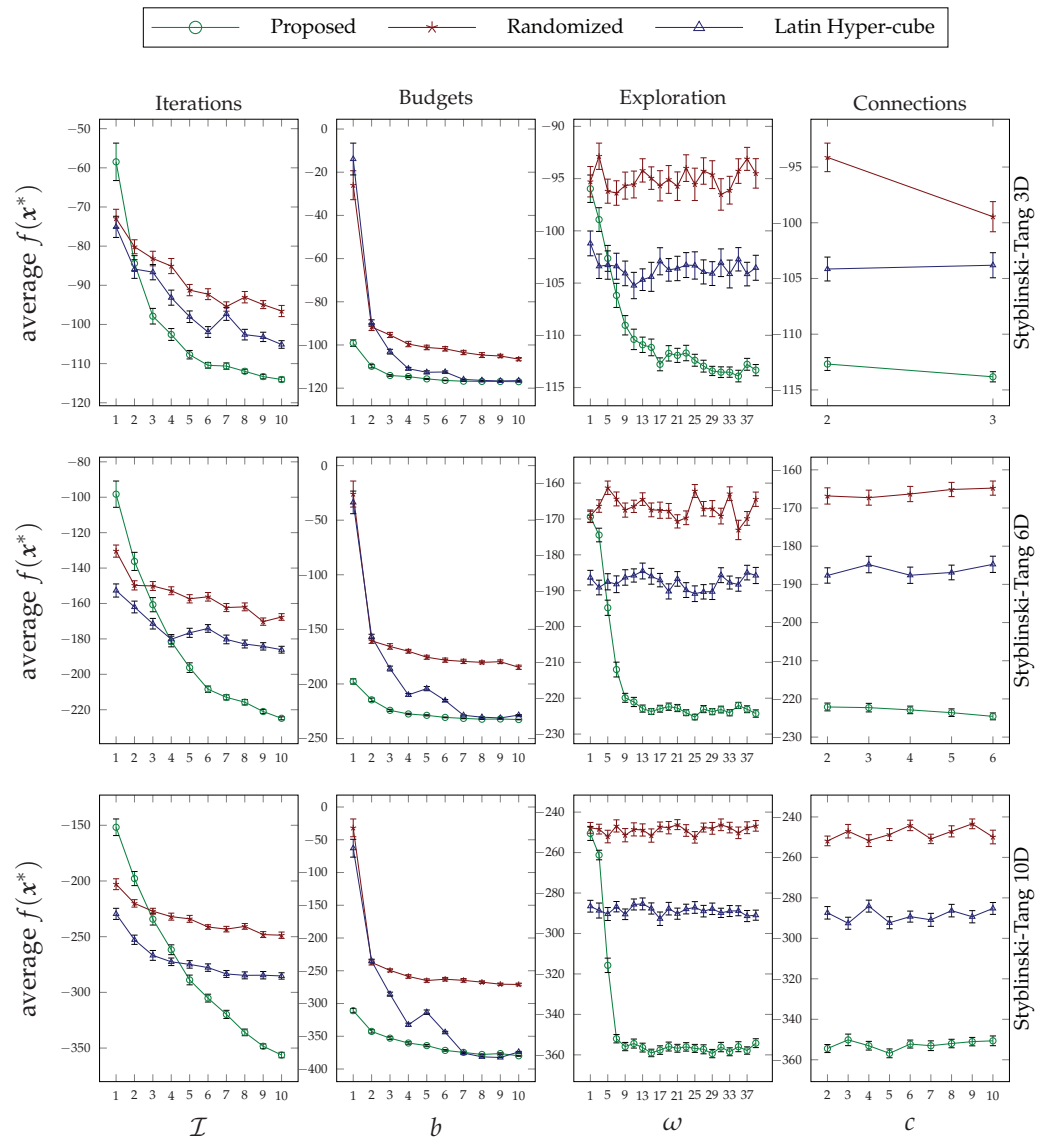


Figure 7. Average values of the Styblinski–Tang function optimized under variable iterations, budgets, explorations, and connection thresholds. Each row of the figure pertains to a particular dimension size, and the error bars are calculated based on the standard error.

The results exhibited in Figure 10 show the behavior of the tested optimization algorithms under various budget restrictions. In this set of experiments, we have fixed the number of iterations to $\mathcal{I} = 10$, and the results show a promising success of our method in outperforming the other two in most problems. Similar to the rationale provided above, the amount of improvement in Rastrigin and Styblinski–Tang functions is evident. Moreover, our method also shines when we have a low budget for the number of evaluations in each iteration. In other words, it can be a good candidate for optimizing expensive-to-evaluate problems or its use in computationally limited devices.

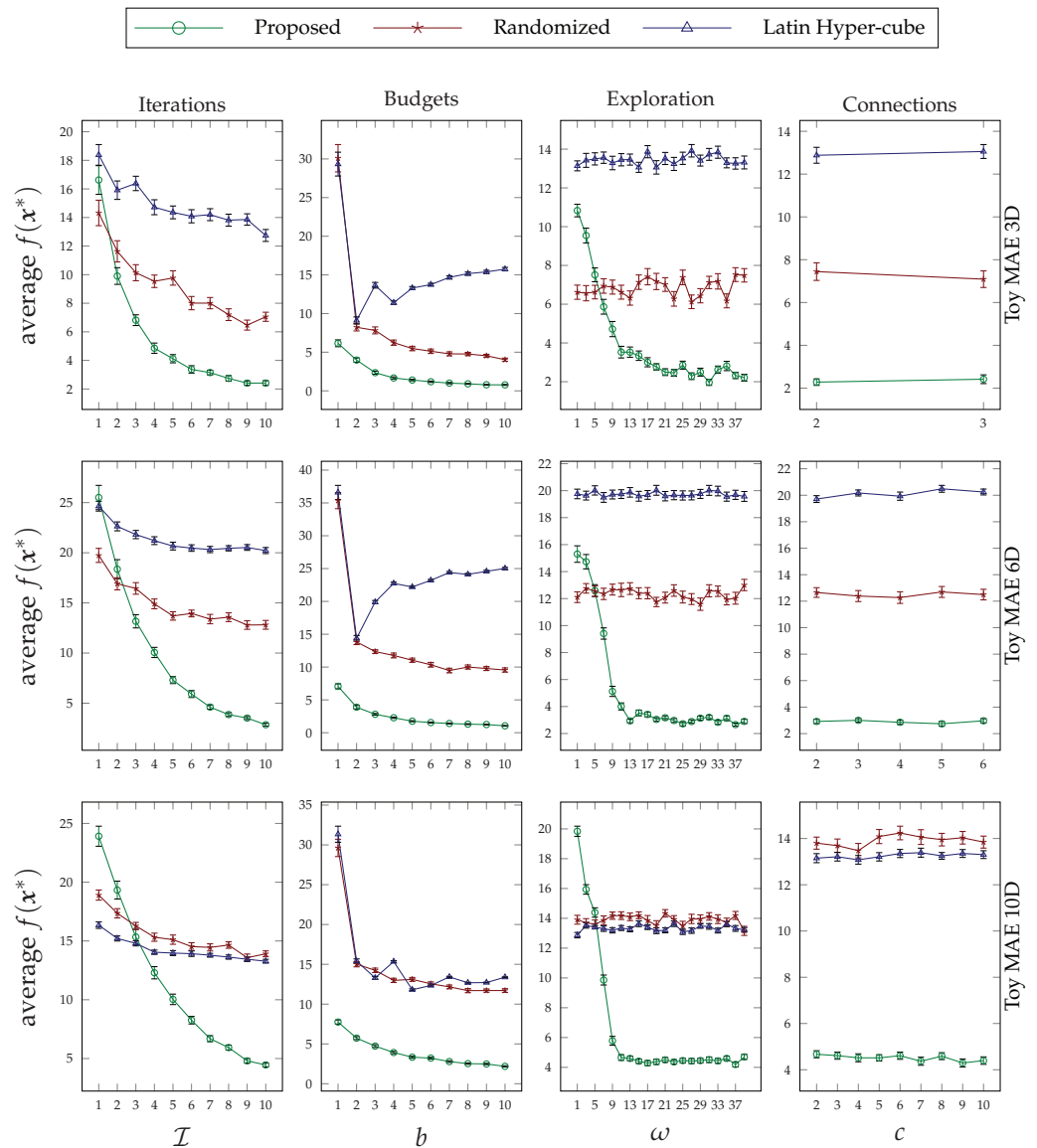


Figure 8. Average values of the toy mean absolute error function optimized under variable iterations, budgets, explorations, and connection thresholds. Each row of the figure pertains to a particular dimension size, and the error bars are calculated based on the standard error.

Regarding the computational time, we extend our analysis of the time complexity of the proposed methods in the previous section to the compared random-based methods. Let b, \mathcal{I} , and $|\lambda_o|$ denote the evaluation budget of each agent, the number of iterations, and the total number of hyperparameter/decision variables to be optimized, respectively. As we have compared the methods under fair conditions, i.e., giving each agent the opportunity to run its randomized algorithm for \mathcal{I} times, and since we have assumed that all $|\lambda_o|$ agents run independently in parallel, the time complexity of both “randomized” and “Latin hypercube” methods would be $\mathcal{O}(Ib\mathcal{R})$, where \mathcal{R} denotes the complexity of the underlying ML model or function evaluation. Recall from the previous section that the time complexity of the proposed method is $\mathcal{O}(|\lambda_o| + Ib\mathcal{R})$ due to its initial structure formation phase and the vertical communication of non-terminal agents. In other words, our proposed approach requires additional $\mathcal{O}(|\lambda_o|)$ computational time in the worst case. The worst case occurs when the computational time complexity of the evaluation of the objective function or the ML model, i.e., $\mathcal{O}(\mathcal{R})$, is low. In almost all ML tasks however, we have $\mathcal{O}(|\lambda_o|) \ll \mathcal{O}(\mathcal{R})$, hence the time difference is negligible. It is worth emphasizing that this comparison is based on the assumption of a fair comparison and parallel execution of the budgeted agents.

It is clear that any changes applied to the benefit of a particular method will definitely change the requirements. For instance, if we limit the number of evaluations in randomized methods, they will require less time to find a local minimum; however, the result will be of lower quality. Regarding the tested heuristic methods, as we have kept the number of evaluations fixed and due to using a similar amount of work internally, we expect a computational complexity similar to our approach for them.

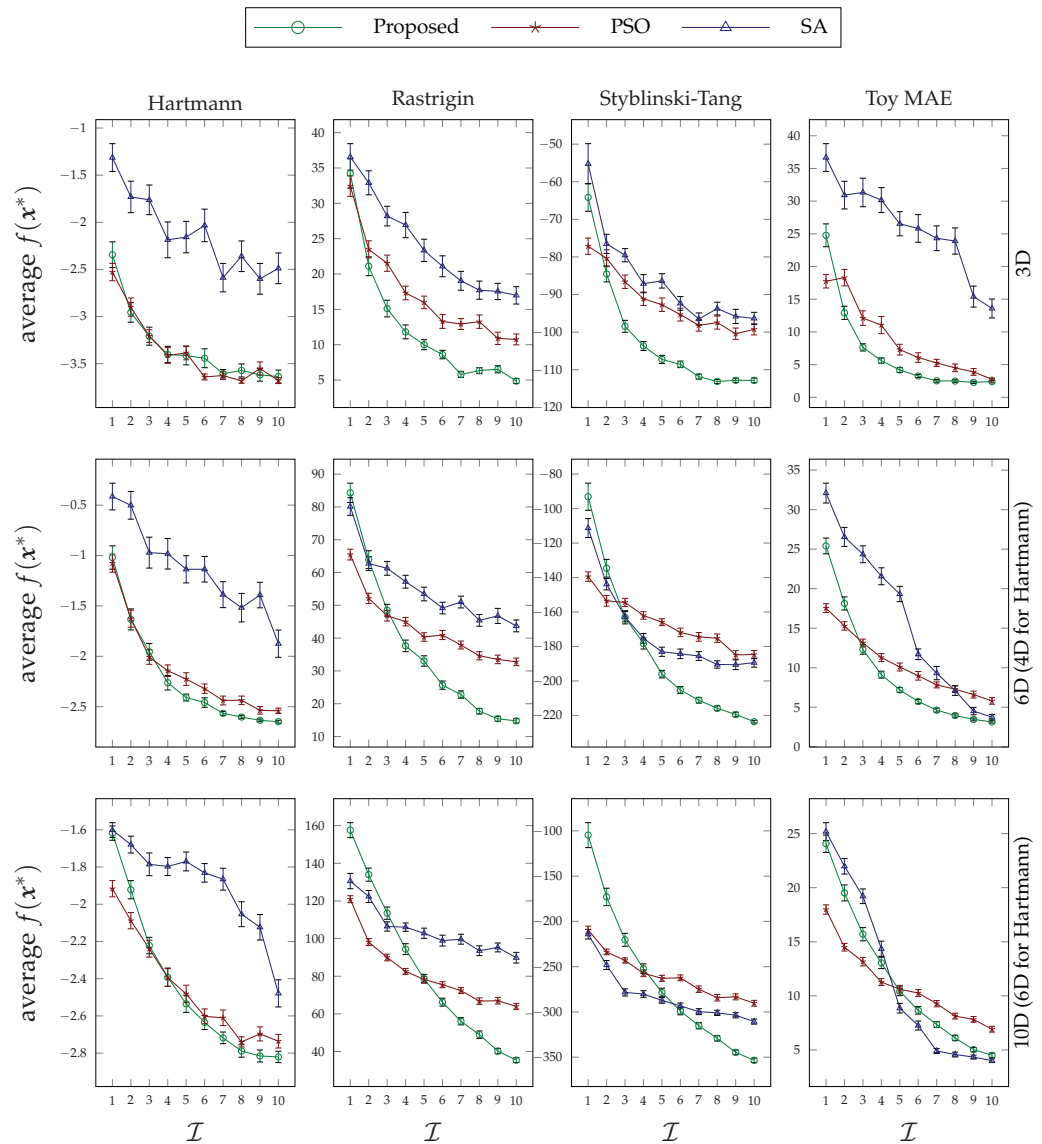


Figure 9. Average function values of four objective functions optimized under a variable number of iterations. Each row of the figure pertains to a particular dimension size, and the error bars are calculated based on the standard error.

It is worth reiterating that the contribution of this paper is not to compete with the state-of-the-art algorithms in function optimization, but to propose a distributed tuning/optimization approach that can be deployed on a set of distributed and networked devices. The discussed analytical and empirical results not only demonstrated the behavior and impact of the design parameters that we have used in our approach, but also suggested the way that they can be adjusted for different needs. We believe the contribution of this paper can be significantly improved with more sophisticated and carefully chosen tuning strategies and corresponding configurations.

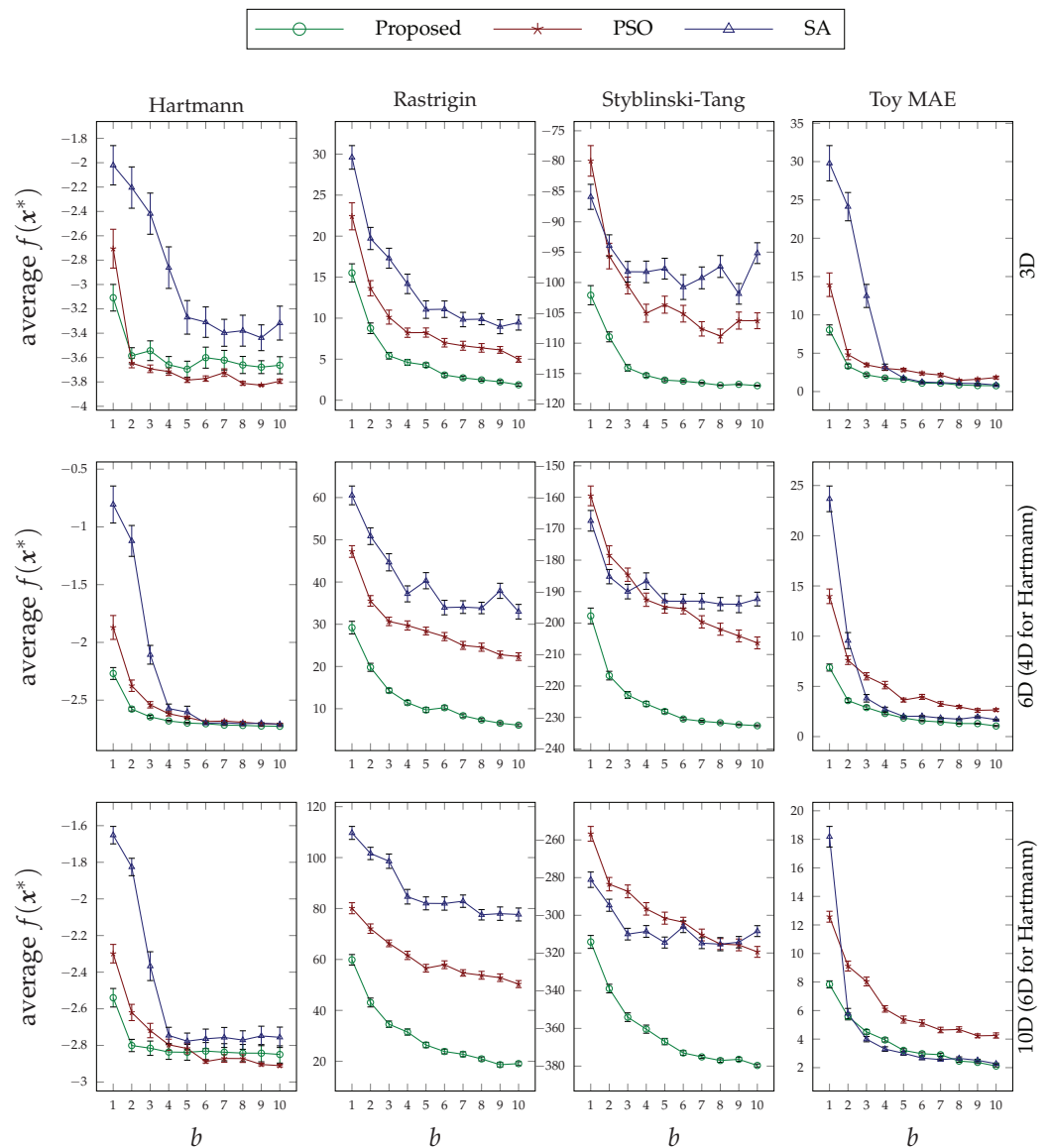


Figure 10. Average function values of four objective functions optimized under a variable number of budget values. Each row of the figure pertains to a particular dimension size, and the error bars are calculated based on the standard error.

4. Conclusions

This paper presented an agent-based collaborative random search method that can be used for machine learning hyper-parameter tuning and black-box optimization problems. The approach employs two types of agents during the tuning/optimization process: the internal and terminal agents that are responsible for facilitating collaborations and tuning individual decision variables, respectively. Such agents and the interaction network between them are created during the hierarchy formation phase and remain the same for the entire runtime of the suggested method. Thanks to the modular and distributed nature of the approach and its procedures, it can be easily deployed on a network of devices with various computational capabilities. Furthermore, the design parameters used in this technique enable each individual agent to customize its own searching process and behavior independent from its peers in the hierarchy, allowing for diversity in both algorithmic and deployment levels.

The paper dissected the proposed model from different aspects and provided some tips on handling its behavior for various applications. According to the analytical dis-

cussions, our approach requires slightly more computational and storage resources than the traditional and Latin hypercube randomized search methods that are commonly used for both hyper-parameter tuning and black-box optimization problems. However, this results in significant performance improvements, especially in computationally restricted circumstances with higher numbers of decision variables. This conclusion was verified in both machine learning model tuning tasks and general multi-dimensional function optimization problems. Furthermore, the empirical results on two widely used heuristic methods, namely PSO and SA, showed that our method exhibits better exploration and potential for escaping local optima while using limited computational resources.

The presented work can be further extended both technically and empirically. As was discussed throughout this paper, we kept the searching strategies and the way the design parameters are configured as simple as possible so we could reach a better understanding of the effectiveness of the collaborations and searching space divisions. A few potential extensions in this direction include: the utilization of diverse searching methods, hence the possession of a heterogeneous multi-agent system at the terminal level; the split of the searching space that is not based on the dimensions, but rather on the range of the values that decision variables in each dimension can have; employment of more sophisticated collaboration techniques; and the use of a learning-based approach to dynamically adapt the values of the design parameters during the runtime of the method. Empirically, the presented research can be extended by completing an in-depth comparison with population-based methods and applying our method to expensive machine learning tasks, such as tuning deep learning models with a large number of hyper-parameters. We are currently working on some of these studies and suggest them as future work.

Author Contributions: Methodology, A.E.; validation, Z.G.; investigation, A.E. and Z.G.; writing—original draft, A.E.; writing—review and editing, Z.G.; visualization, A.E.; supervision, E.T.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
2. Kohavi, R.; John, G.H. Automatic parameter selection by minimizing estimated error. In *Machine Learning Proceedings 1995*; Elsevier: Amsterdam, The Netherlands, 1995; pp. 304–312.
3. Bischl, B.; Mersmann, O.; Trautmann, H.; Weihs, C. Resampling Methods for Meta-Model Validation with Recommendations for Evolutionary Computation. *Evol. Comput.* **2012**, *20*, 249–275. [[CrossRef](#)] [[PubMed](#)]
4. Montgomery, D.C. *Design and Analysis of Experiments*; John Wiley & Sons: Hoboken, NJ, USA, 2017.
5. John, G.H. *Cross-Validated C4.5: Using Error Estimation for Automatic Parameter Selection*; Technical Report; Stanford University: Stanford, CA, USA, 1994.
6. Močkus, J. On Bayesian methods for seeking the extremum. In Proceedings of the Optimization Techniques IFIP Technical Conference, Novosibirsk, Russia, 1–7 July 1974; Springer: Berlin/Heidelberg, Germany, 1975; pp. 400–404.
7. Mockus, J. *Bayesian Approach to Global Optimization: Theory and Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 37.
8. Feurer, M.; Hutter, F. Hyperparameter optimization. In *Automated Machine Learning*; Springer: Cham, Switzerland, 2019; pp. 3–33.
9. Simon, D. *Evolutionary Optimization Algorithms*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
10. Alibrahim, H.; Ludwig, S.A. Hyperparameter Optimization: Comparing Genetic Algorithm against Grid Search and Bayesian Optimization. In Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June–1 July 2021; pp. 1551–1559.
11. Bellman, R.E. *Adaptive Control Processes*; Princeton University Press: Princeton, NJ, USA, 1961.
12. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; De Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* **2015**, *104*, 148–175. [[CrossRef](#)]
13. Garcia-Barcos, J.; Martinez-Cantin, R. Fully Distributed Bayesian Optimization with Stochastic Policies. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019.
14. Young, M.T.; Hinkle, J.D.; Kannan, R.; Ramanathan, A. Distributed Bayesian optimization of deep reinforcement learning algorithms. *J. Parallel Distrib. Comput.* **2020**, *139*, 43–52. [[CrossRef](#)]
15. Frazier, P.I. A Tutorial on Bayesian Optimization. *arXiv* **2018**, arXiv:1807.02811.

16. Friedrichs, F.; Igel, C. Evolutionary tuning of multiple SVM parameters. *Neurocomputing* **2005**, *64*, 107–117. [[CrossRef](#)]
17. Loshchilov, I.; Hutter, F. CMA-ES for hyperparameter optimization of deep neural networks. *arXiv* **2016**, arXiv:1604.07269.
18. Ryzko, D. *Modern Big Data Architectures: A Multi-Agent Systems Perspective*; John Wiley & Sons: Hoboken, NJ, USA, 2020.
19. Esmaeili, A.; Gallagher, J.C.; Springer, J.A.; Matson, E.T. HAMLET: A Hierarchical Agent-Based Machine Learning Platform. *ACM Trans. Auton. Adapt. Syst.* **2022**, *16*, 1–46. [[CrossRef](#)]
20. Esmaeili, A.; Ghorrati, Z.; Matson, E.T. Hierarchical Collaborative Hyper-Parameter Tuning. In Proceedings of the Advances in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation. The PAAMS Collection, L'Aquila, Italy, 13–15 July 2022; Springer International Publishing: Cham, Switzerland, 2022; pp. 127–139.
21. Bardenet, R.; Brendel, M.; Kégl, B.; Sebag, M. Collaborative hyperparameter tuning. In Proceedings of the International Conference on Machine Learning, PMLR, Atlanta, GA, USA, 17–19 June 2013; pp. 199–207.
22. Swearingen, T.; Drevo, W.; Cyphers, B.; Cuesta-Infante, A.; Ross, A.; Veeramachaneni, K. ATM: A distributed, collaborative, scalable system for automated machine learning. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 151–162.
23. Koch, P.; Golovidov, O.; Gardner, S.; Wujek, B.; Griffin, J.; Xu, Y. Autotune: A derivative-free optimization framework for hyperparameter tuning. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 443–452.
24. Iranfar, A.; Zapater, M.; Atienza, D. Multi-agent reinforcement learning for hyperparameter optimization of convolutional neural networks. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2021**, *41*, 1034–1047. [[CrossRef](#)]
25. Parker-Holder, J.; Nguyen, V.; Roberts, S.J. Provably efficient online hyperparameter optimization with population-based bandits. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17200–17211.
26. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
27. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol. (TIST)* **2011**, *2*, 1–27. [[CrossRef](#)]
28. Crammer, K.; Dekel, O.; Keshet, J.; Shalev-Shwartz, S.; Singer, Y. Online passive aggressive algorithms. *J. Mach. Learn. Res.* **2006**, *7*, 551–585.
29. Friedman, J.; Hastie, T.; Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* **2010**, *33*, 1. [[CrossRef](#)] [[PubMed](#)]
30. Scikit-Learn API Reference. Available online: <https://scikit-learn.org/stable/modules/classes.html> (accessed on 24 November 2022).
31. Jamil, M.; Yang, X.S. A literature survey of benchmark functions for global optimisation problems. *Int. J. Math. Model. Numer. Optim.* **2013**, *4*, 150. [[CrossRef](#)]
32. Rudolph, G. Globale Optimierung Mit Parallelen Evolutionsstrategien. Ph.D. Thesis, Diplomarbeit, Universit at Dortmund, Fachbereich Informatik, Dortmund, Germany, 1990.
33. Styblinski, M.; Tang, T.S. Experiments in nonconvex optimization: Stochastic approximation with function smoothing and simulated annealing. *Neural Netw.* **1990**, *3*, 467–483. [[CrossRef](#)]
34. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
35. Kirkpatrick, S.; Gelatt, C.D., Jr.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Conflicting Bundle Allocation with Preferences in Weighted Directed Acyclic Graphs: Application to Orbit Slot Allocation Problems[†]

Stéphanie Roussel^{1,*}, Gauthier Picard¹, Cédric Pralet¹ and Sara Maqrot²

¹ ONERA/DTIS, Université de Toulouse, F-31055 Toulouse, France; gauthier.picard@onera.fr (G.P.); cedric.pralet@onera.fr (C.P.)

² Berger-Levrault/DRIT, 31670 Labège, France; sara.maqrot@berger-levrault.com

* Correspondence: stephanie.roussel@onera.fr

† This paper is an extended version of our paper published in *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation. The PAAMS Collection. PAAMS 2022. Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2022; Volume 13616.

Abstract: We introduce resource allocation techniques for problems where (i) the agents express requests for obtaining item bundles as compact edge-weighted directed acyclic graphs (each path in such a graph is a bundle whose valuation is the sum of the weights of the traversed edges), and (ii) the agents do not bid on the exact same items but may bid on conflicting items that cannot be both assigned or that require accessing a specific resource with limited capacity. This setting is motivated by real applications such as Earth observation slot allocation, virtual network functions, or multi-agent path finding. We model several directed path allocation problems (vertex-constrained and resource-constrained), investigate several solution methods (qualified as exact or approximate, and utilitarian or fair), and analyze their performances on an orbit slot ownership problem, for realistic requests and constellation configurations.

Keywords: path allocation; fairness; constraint optimization; satellite constellation

Citation: Roussel, S.; Picard, G.; Pralet, C.; Maqrot, S. Conflicting Bundle Allocation with Preferences in Weighted Directed Acyclic Graphs: Application to Orbit Slot Allocation Problems. *Systems* **2023**, *11*, 297. <https://doi.org/10.3390/systems11060297>

Academic Editors: Fernando De la Prieta Pintado, Philippe Mathieu, Juan M. Corchado and Alfonso González-Briones

Received: 24 March 2023

Revised: 22 May 2023

Accepted: 1 June 2023

Published: 9 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Earth observation satellites capture a vast number of images of the Earth's surface every day. These images are delivered to end-users who have made observation requests for several purposes such as monitoring critical areas affected by natural disasters or crises, observing infrastructures, monitoring the environment, etc. The observation request process operates in the following manner. First, users submit their observation requests to the main mission center. The mission center then computes observation plans which are transmitted to the satellites when they overfly a ground control station. Subsequently, each satellite captures the requested images and transmits the collected data when it passes over a ground reception station. The satellites we consider in this work are on low Earth orbit and complete around 16 orbits per day, which allows them to pass over several Earth areas at different times every day.

In order to improve the capability to deliver images as early as possible after requests are formulated, one can rely on constellations of Earth observation satellites that are currently deployed. Constellations also offer the possibility for users to express more complex requests. An example of such a complex request is a *periodic request*, that consists in observing an area of interest at regularly spaced dates. Generally, the number of posted requests on a given time horizon is too large to satisfy them all. Therefore, the main mission center has to select which requests to perform for the upcoming time horizon, for instance using manually defined prioritization rules. As such a selection process does not offer guarantees for users with regards to the satisfaction of their requests, Earth observation satellite constellations' managers now propose a new observation paradigm, namely, *exclusivity orbit slots* booking. Whenever users

buy exclusive orbit slots of a satellite, they can exploit this satellite during the associated time windows using their own ground stations. This allows users to send observation plans to satellites and collect the observations realized during the orbit slots.

In this respect, from the point of view of the operator of an Earth observation satellite constellation, we consider the following problem. The goal is to attribute ownership of some orbit portions to several clients. Each client has some points of interest (POIs) to acquire at some frequency, e.g., capture L'Aquila city every 2 h for 6 months. Since several satellites may capture the very same point on Earth around the requested observation times, several possible bundles of orbit slots are specified by each client, together with a preference for some bundles depending on the quality of the sequence of orbit slots, e.g., based on the POI viewing angle provided by each slot. Moreover, as several clients may be interested in very close POIs, several requested orbit slots may overlap. Each orbit slot in this category can be either allocated to a single client or divided between clients. These situations can be captured by the models we propose in this article.

More precisely, we consider a problem of allocation of conflicting bundles of items constrained by item chaining (to allocate to each agent a chain of successive items). The chaining constraint is captured by using, for each agent, an edge-weighted directed acyclic graph (DAG) representing all the valid bundles (i.e., paths) of items for the agent, where the quality of a bundle is represented by additive edge weights. Then, conflicting bundles cannot be allocated at the same time and have to be handled so that each agent obtains one conflict-free path in its graph. Such a setting occurs in application domains such as network function virtualization (NFV), where users request allocating directed graphs of services into a shared networked infrastructure [1]. As explained before, this also occurs in Earth observation using a constellation of satellites in a scenario where users demand the ownership of some repetitive orbit slots, without overlapping with other users' slots, to fulfill periodic observation requests [2,3]. In such settings, beside the additive edge weights, other criteria can be considered to guide the allocation process, especially when constellation users are stakeholders expecting allocations to be fair or proportional to their investment.

In this paper, we contribute on the following points:

- We define a generic modeling framework for the path allocation problem with conflict (directed path allocation problem, or DPAP) and consider two optimization criteria (global utility and leximin).
- We instantiate this framework with two compact representations of conflicts, one based on a vertex conflict (vertex-constrained directed path allocation problem, or V-DPAP) and one based on a resource consumption conflict (resource-constrained directed path allocation problem, or R-DPAP)—note that V-DPAP comes from the path allocation in the directed acyclic graph (PADAG) problem defined in [4].
- We show that the decision problems associated with V-DPAP and R-DPAP are NP-complete, whatever the optimization criteria.
- We define several complete and incomplete allocation schemes for solving V-DPAP and R-DPAP.
- We evaluate all of the algorithmic approaches on dozens of orbit slot allocation benchmarks and discuss the obtained results.

The paper is structured as follows. Section 2 discusses related works focusing on the allocation of goods as paths. Section 3 presents the DPAP framework to tackle path allocation in multiple conflicting edge-weighted directed acyclic graphs. In Sections 4 and 5, we consider vertex-based conflicts (V-DPAP) and resource-based conflicts (R-DPAP). We analyze the theoretical complexity of the associated decision problems and discuss the relationship between the two frameworks. Section 6 lists some algorithms, complete and incomplete, that can be used to solve V-DPAP and R-DPAP. Section 7 presents the experiments used to evaluate the performances and behaviors of our solution methods on problem instances coming from the Earth observation domain. Finally, Section 8 concludes the article with some perspectives.

2. Related Works

The literature contains some work related to the allocation of goods structured as graphs. In fair division of graphs, the objective is to divide a graph of items between several agents, with additive utilities attached to nodes [5,6]. These works provide interesting properties to find envy-free or Pareto-optimal, allocations in an efficient manner in some specific graph structures, e.g., paths, trees, stars. However, in our problem, (i) agents do not compete for the very same set of items, (ii) the graph is directed to compose paths from a start time to an end time, and (iii) even by mapping our problem to a graph division problem and by regrouping conflicting items into composite items, it is highly improbable that the resulting graph is acyclic. Here, graphs are used to express preferences and not the goods to allocate. In short, our work does not fall into the existing graph fair division frameworks, and cannot benefit from theoretical results on path-shaped or star-shaped graphs.

Another related method is path auctions [7–9], where agents bid for paths in a graph where each edge is owned by an agent. The goal is to assign paths to agents by the means of auctions, and optionally to keep some privacy for the edge owners. In the case of a utilitarian objective function for the winner determination problem, without price privacy, this falls into the Vickrey–Clarke–Groves framework, and thus guarantees some efficient and *strategy-proof* mechanisms. However, here again, agents bid on the very same set of nodes and edges.

In the transportation domain, investigations on very similar structures, that is flow networks, provide techniques for fair maximum flow in multi-source and multi-sink networks [10]. While the techniques used are very similar to ours (linear programming), the maximum flow objective is very different from path utility maximization with a single path per agent. Furthermore, [11] worked on multiple shortest path problems based on deconflicting techniques. While the problem displays similar characteristics, once again the agents evolve on the very same graphs, and the objective is focused on minimizing path length and minimizing conflicting paths, without fairness desiderata.

In congestion games, agents are allocated paths so that the delays incurred by crossing paths are minimized. The more agents are allocated the same nodes, the more delay is attached to their paths [12,13]. In our work, we do not consider delay but incompatibilities. Even if they could be modeled as nonlinear $\{0, \infty\}$ functions, in our problem some path allocations are unfeasible, contrarily to congestion games. Furthermore, using congestion game solution methods, as in [13], may result in unfair Nash equilibria, because of numerous unfeasible paths¹.

More generally, another classical approach to the fair allocation of indivisible goods is *round-robin*, which is almost envy-free [14]. This is notably one favored technique to allocate virtual network functions in network function virtualization infrastructures [15], or to schedule tasks. We will use it as a competitor for our techniques.

In [16], we proposed constraint-programming approaches for fair sharing of orbit slots in the case of Earth observation satellites. We considered several types of requests, such as periodic and global requests. The latter type of requests cannot be modeled within the graph-based framework proposed in this paper. Therefore, we had to enumerate all the ways to (partially) satisfy requests. This enumeration is not required within the framework we propose here, because of the compact graph representation. Moreover, the approaches of [16] were evaluated on small horizons, due to the computational intensiveness of the proposed solution methods. The horizons considered in this paper are much longer.

In this paper, we investigate several mathematical programming-based (utilitarian, leximin, approximate leximin) and ad hoc algorithms (greedy, round-robin) to allocate paths in conflicting graphs. We generalize our previous work [4] to the case of directed path allocation problems (DPAP) and consider another conflict expression that is based on resources. Note that a more detailed description of the work performed in [4] is presented later in the paper.

In another direction, there is a wide literature on Earth observation scheduling problems (EOSPs) [17]. In such problems, some observation requests have to be assigned to

satellites and scheduled for each satellite so that several constraints (e.g., temporal constraints related to the possible maneuvers of the satellites) are satisfied. Various criteria have been studied in the literature. Nevertheless, as the problem is generally over-constrained because of the number of requests to satisfy, a classical criterion is to optimize a (weighted) total reward provided by the satisfied requests. The Earth observation scheduling problem has a lot in common with the orbit slot allocation problem (OSAP) considered in this paper. Indeed, both problems involve observation requests posted by users, visibility windows, constraints related to the satellite disjunctive nature, etc. However, there are several differences between the two problems. First, in the orbit slot allocation problem, users want to “own” the satellite during some orbit portions in order to perform a set of observations. The targets to be observed during each slot are not precisely known in advance, which means that constraints about the satellites’ maneuvers are irrelevant in the case of OSAP. Then, the requests’ nature is different. In fact, in OSAP, the requests are composed of several slots possibly over several months ahead, and each slot is quite long as it is supposed to allow the user to perform several observations. In the case of EOSPs, there are many more requests but on a very short time horizon (a few days at most), and each request requires a very small amount of satellite time. Finally, fairness between users is essential in the case of OSAPs, whereas it is rarely considered in EOSPs. Two exceptions are the work described in [18], where the authors study a multi-objective EOSP and aim at maximizing the total profit and minimizing the maximum profit difference between each pair of users, and the work described in [19], where a heuristic method is proposed to solve the EOSP while taking into account fairness.

Using graphs in the context of EOSPs is not novel. In [20], an activity-on-node graph allows modeling of all the alternatives to satisfy observation requests by a set of satellites (one node is one opportunity to observe a request target by a satellite, and the edges allow conflicts between observation candidates to be represented). Then, maximizing the number of satisfied requests amounts to computing the maximum independent set of the graph.

3. Directed Path Allocation Problems

In this section, we define the so-called directed path allocation problem (DPAP), where agents’ valuations of item bundles are represented as edge-weighted DAGs, as illustrated in Figure 1, and where the goal is to select one path in each DAG while satisfying set compatibility constraints over the selected paths. We first introduce some notation related to graphs and then formalize the generic problem we consider.

Definition 1. A single-source single-sink edge-weighted DAG g is a triple $\langle V_g, E_g, u_g \rangle$ such that:

- V_g is a set of nodes; in our case, each node corresponds to an item that can be allocated to an agent, except for two specific nodes referred to as the source s_g and the sink t_g ;
- $E_g \subset V_g \times V_g$ is the set of arcs of the acyclic graph, with the assumption that s_g and t_g are, respectively, the unique source and sink of the graph; an arc $v_1 \rightarrow v_2$ indicates that items v_1 and v_2 can be selected sequentially;
- $u_g : E_g \rightarrow \mathbb{R}^+$ is a utility function that associates a weight to each arc of the graph to represent a preference over the combinations of item selections; we assume that E_g contains an arc from s_g to t_g labeled by utility 0, to deal with cases where no bundle of items can be selected in g .

In the following, the set of paths from s_g to t_g is denoted by Π_g .

For each graph g and each set of edges $X \subseteq E_g$, the utility of X for g is defined by $u_g(X) = \sum_{e \in X} u_g(e)$, which means that edge valuations are additive. As a result, each path from s_g to t_g in a graph g is evaluated by summing the utilities of the traversed edges, and each DAG represents, in a compact manner, a set of valuations for bundles of items, as in combinatorial auctions.

Definition 2. A Directed Path Allocation Problem (DPAP) is a tuple $\langle \mathcal{A}, \mathcal{G}, \mu, \phi \rangle$, where

- $\mathcal{A} = \{1, \dots, n\}$ is a set of agents;

- $\mathcal{G} = \{g_1, \dots, g_m\}$ is a set of single-source single-sink edge-weighted DAGs, as introduced in Definition 1;
- $\mu : \mathcal{G} \rightarrow \mathcal{A}$ maps each graph g in \mathcal{G} to its owner a in \mathcal{A} ; we also denote by $\mathcal{G}_a = \mu^{-1}(a)$ the set of graphs owned by agent a ;
- $\phi : \Pi_{g_1} \times \dots \times \Pi_{g_m} \rightarrow \{0, 1\}$ is a path compatibility function that indicates whether a combination of paths (p_1, \dots, p_m) (one path per graph) is feasible (value 1) or not (value 0).

In a DPAP, the definition of the path compatibility function ϕ is related to the presence of items that cannot be shared by the agents. More precisely, a conflict between two paths represents the fact that assigning these paths to clients is infeasible (e.g., because some orbit slots overlap) or strongly undesirable for the constellation manager. A naive definition of the compatibility function is the list of combinations of paths that are compatible with each other. However, the number of paths in a DAG is exponential, which makes this definition impractical in the general case. Therefore, in the next sections, we propose and discuss different ways to define the compatibility function in a compact way.

Example 1. Figure 1 illustrates a DPAP representing an orbit slot allocation problem. In such a problem, satellite orbit slots must be allocated to agents so that the latter can make several observations of a POI on Earth. In this example, we consider two agents A and B that each have one observation request, request a for agent A and request b for agent B.

Within the DPAP modeling framework, we consider a graph for each request: graph g_a for request a and graph g_b for request b. The nodes of these graphs are the orbit slot candidates for each request (slots $a_1, a_2,$ and a_3 for request a, and slots $b_1, b_2, b_3,$ and b_4 for request b). A path in a graph represents a way to satisfy the corresponding request. For instance, for satisfying request a, starting from s_a , one can either select first slot a_1 and then slot a_2 , or select first slot a_3 and then slot a_2 . Each edge has a utility that represents the reward for selecting slots in a given order. For instance, edges $s_a \rightarrow a_1$ and $s_a \rightarrow a_3$ have utilities equal to 0.2 and 0.5, respectively. This represents the fact that agent A prefers selecting slot a_3 rather than selecting slot a_1 . Such a difference can be due to a satellite viewing angle that is better for a_3 than for a_1 . Note that for a node, its incoming edges do not necessarily have the same utility value. For instance, the utility of edge $b_1 \rightarrow b_4$ is equal to 0.1, whereas the utility of edge $b_3 \rightarrow b_4$ is equal to 0.3.

The graph associated with request a contains three possible paths while the graph associated with request b contains five possible paths. We assume here that only 10 combinations of paths are allowed by the path compatibility function ϕ among the 15 possible ones. For instance, paths $\pi_{a,1}$ and $\pi_{b,2}$ are compatible ($\phi(\pi_{a,1}, \pi_{b,2}) = 1$) but paths $\pi_{a,1}$ and $\pi_{b,3}$ are not ($\phi(\pi_{a,1}, \pi_{b,3}) = 0$).

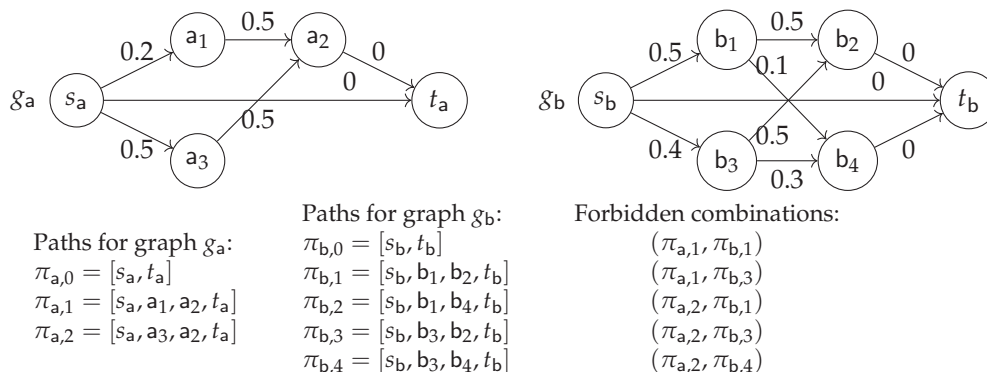


Figure 1. Sample users' bundle valuations (or preferences) represented as a DPAP.

Definition 3. For a DPAP $\langle \mathcal{A}, \mathcal{G}, \mu, \phi \rangle$, an allocation is a function π that associates, with each graph $g \in \mathcal{G}$, one path $\pi(g)$ from s_g to t_g in g . If $\mathcal{G} = \{g_1, \dots, g_m\}$, such an allocation is valid if and only if $\phi(\pi(g_1), \dots, \pi(g_m)) = 1$ holds. Formally, $\pi(g)$ can be represented as a set of nodes in V_g . Indeed, as DAGs are manipulated, it is easy to reconstruct the edges successively traversed by the path from this set.

Definition 4. For a DPAP $\langle \mathcal{A}, \mathcal{G}, \mu, \phi \rangle$, the global utility $u(\pi)$ associated with an allocation π is the sum of the utilities obtained in each graph, that is $u(\pi) = \sum_{g \in \mathcal{G}} u_g(\pi(g))$. The utility obtained for agent a is $u_a(\pi) = \sum_{g \in \mathcal{G}_a} u_g(\pi(g))$.

Definition 5. For a DPAP $\langle \mathcal{A}, \mathcal{G}, \mu, \phi \rangle$ involving n agents, the leximin utility vector associated with an allocation π is the vector $\text{lex}(\pi) = (\Lambda_1, \dots, \Lambda_n)$ that corresponds to vector $(u_1(\pi), \dots, u_n(\pi))$ sorted following an increasing order ($\Lambda_i \leq \Lambda_j$ holds for $i < j$).

If π and π' denote two allocations for a given DPAP, and $\text{lex}(\pi) = (\Lambda_1, \dots, \Lambda_n)$ and $\text{lex}(\pi') = (\Lambda'_1, \dots, \Lambda'_n)$ are their associated leximin utility vectors, π is strictly better than π' with respect to the leximin criterion if there exists k in $[1..n]$ such that $\Lambda_k > \Lambda'_k$ and for all $i < k$, $\Lambda_i = \Lambda'_i$. Note that leximin-based fair allocations allow the favoring of agents that are less satisfied.

The problems we consider in this paper are: (i) how to compute an optimal (utilitarian) valid allocation π that maximizes $u(\pi)$, and (ii) how to compute an optimal fair valid allocation π that maximizes $\text{lex}(\pi)$.

Example 2. In the graphs described in Example 1 and illustrated in Figure 1, the individual best paths for agents A and B are $\{s_a, a_3, a_2, t_a\}$ and $\{s_b, b_1, b_2, t_b\}$, respectively. They both have a utility equal to 1. However, these paths are not compatible according to the list of forbidden paths and cannot both belong to a valid allocation.

Figure 2a gives an example of a valid allocation $\pi_{\text{ex}} = \{g_a \mapsto \{s_a, a_1, a_2, t_a\}, g_b \mapsto \{s_b, b_1, b_4, t_b\}\}$ for the DPAP introduced before. The global utility of π_{ex} is $u(\pi_{\text{ex}}) = u(\pi_{\text{ex}}(A)) + u(\pi_{\text{ex}}(B)) = 0.7 + 0.6 = 1.3$. The leximin vector associated with π_{ex} is $\text{lex}(\pi_{\text{ex}}) = (0.6, 0.7)$: agent B has the lowest utility (0.6), and agent A's utility is equal to 0.7.

Figure 2b illustrates allocation $\pi_{\text{util}} = \{g_a \mapsto \{s_a, a_3, a_2, t_a\}, g_b \mapsto \{s_b, b_1, b_4, t_b\}\}$ that maximizes the global utility: $u(\pi_{\text{util}}) = 1.0 + 0.6 = 1.6$. The leximin vector is $\text{lex}(\pi_{\text{util}}) = (0.6, 1.0)$.

Figure 2c illustrates allocation $\pi_{\text{lex}} = \{g_a \mapsto \{s_a, a_1, a_2, t_a\}, g_b \mapsto \{s_b, b_3, b_4, t_b\}\}$ that maximizes the leximin vector: $\text{lex}(\pi_{\text{lex}}) = (0.7, 0.7)$. The global utility associated with π_{lex} is lower: $u(\pi_{\text{lex}}) = 1.4$.

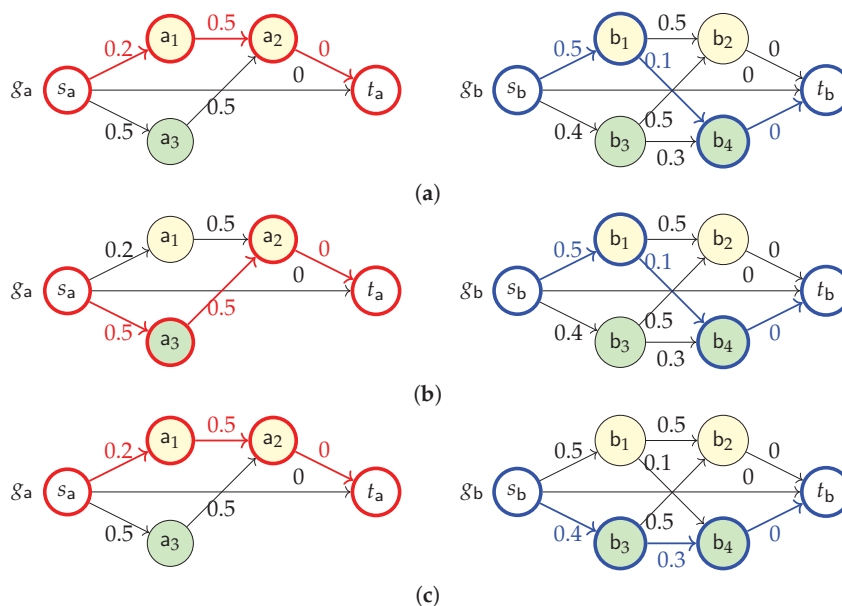


Figure 2. Examples of valid allocations for the DPAP described in Figure 1. (a) Illustration of allocation π_{ex} with the paths selected in graphs g_a and g_b . (b) Allocation π_{util} that maximizes the global utility: $u(\pi_{\text{util}}) = 1.6$. (c) Allocation π_{lex} that maximizes the leximin vector: $\text{lex}(\pi_{\text{lex}}) = (0.7, 0.7)$.

4. V-DPAP: Vertex-Constrained Directed Path Allocation Problems

In practice, the compatibility function ϕ that describes the allowed combinations of paths must be described in a compact way. We study the case where ϕ is simply defined by a set of conflicts between vertices, where each conflict corresponds to a subset of items that cannot all be simultaneously selected. For our target application related to booking orbit slots over a constellation of satellites, this is useful to model situations where two satellite slots required for two distinct booking requests are not compatible because they overlap and require the same satellite. The introduction of conflicts between vertices leads us to a specific case of DPAP called the vertex-constrained directed path allocation problem (V-DPAP). Note that V-DPAP is very close to the problem presented in [4].

4.1. Framework Definition

Definition 6. A Vertex-Constrained Directed Path Allocation Problem (V-DPAP) is a DPAP $\langle \mathcal{A}, \mathcal{G}, \mu, \phi \rangle$ where function ϕ is defined by a set of conflicts \mathcal{C} between vertices of the graph. Each conflict $\sigma \in \mathcal{C}$ is a non-empty set of vertices V_σ that cannot all be selected by an allocation. Moreover, we assume that the vertices in V_σ all belong to distinct graphs.

From this, function ϕ returns a value of 0 for a selection of paths (p_1, \dots, p_m) if and only if there exists a conflict $\sigma \in \mathcal{C}$ such that all vertices in V_σ are traversed by one path in (p_1, \dots, p_m) . Formally, $\phi(p_1, \dots, p_m) = 0$ if there exists $\sigma \in \mathcal{C}$ such that $V_\sigma \subseteq \bigcup_{i=1}^m V_{p_i}$, where V_{p_i} denotes the set of vertices in path p_i .

The previous definition covers both binary conflicts holding on two vertices and n-ary conflicts holding on any set of vertices. This differs from our initial framework, called PADAG, where only binary vertex conflicts were considered [4]. We will sometimes define a V-DPAP as a tuple $\langle \mathcal{A}, \mathcal{G}, \mu, \mathcal{C} \rangle$ equivalent to $\langle \mathcal{A}, \mathcal{G}, \mu, \phi \rangle$ since ϕ is non-ambiguously defined by the set of conflicts \mathcal{C} .

Example 3. Figure 3 illustrates a V-DPAP that contains two conflicts, namely, conflict $\sigma_1 = \{a_2, b_2\}$ that invalidates any combination of paths traversing both a_2 and b_2 , and conflict $\sigma_2 = \{a_3, b_3\}$ that invalidates any combination of paths traversing both a_3 and b_3 . It can be shown that these conflicts lead to the same valid allocations as the ones provided in the DPAP of Figure 1.

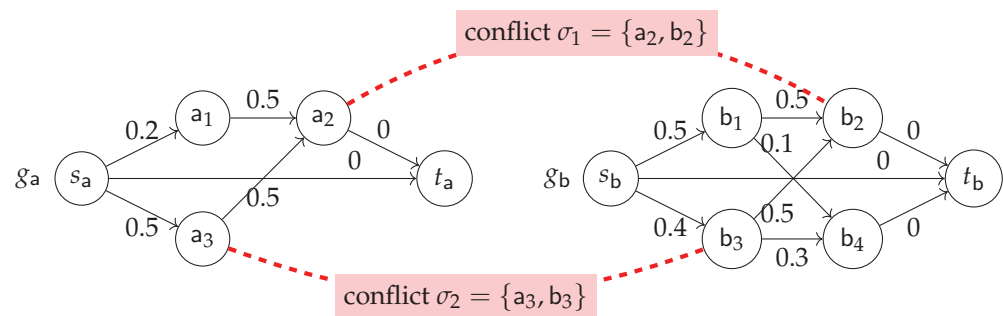


Figure 3. V-DPAP equivalent to the DPAP example of Figure 1; the set of vertex conflicts, represented as red hypernodes, gives a compact representation of the set of allowed combinations of paths.

4.2. Theoretical Complexity

Proposition 1. For a V-DPAP, determining whether there exists a valid allocation π such that utilitarian evaluation $u(\pi)$ is greater than or equal to a given value is NP-complete.

Proof. First, the problem is NP since $u(\pi)$ is computable in polynomial time. Then, there exists a polynomial reduction of 3-SAT (which is NP-complete) to our problem. In a 3-SAT formula that contains m clauses, each clause over the propositional variables x, y, z can be represented as a weighted DAG g , where:

1. the set of nodes is $V_g = \{x, \neg x, y, \neg y, z, \neg z, s_g, t_g\}$,

2. the set of paths from s_g to t_g in g corresponds to the set of truth values for x, y, z that satisfy the clause (decision diagram representation),
3. the weight of every edge is set to 0, except for edges $s_g \rightarrow n$ where $n \neq t_g$, that have weight 1.

Last, for every propositional variable x , we can add one conflict (n, n') for each pair of nodes labeled by the literals x and $\neg x$ in two distinct graphs.

For instance, the 3-SAT problem $(x \vee y \vee z) \wedge (\neg x \vee y \vee \neg w)$ can be represented by the V-DPAP illustrated in Figure 4. Clause $(x \vee y \vee z)$ is translated into graph g_1 and clause $(\neg x \vee y \vee \neg w)$ into graph g_2 . Vertices linked by dashed edges correspond to conflicts.

Then, as one path is selected in each graph and as there are m graphs, determining whether there exists a valid allocation π such that $u(\pi) \geq m$, with m the number of clauses in the 3-SAT formula, is equivalent to finding a solution that satisfies all the clauses, hence the NP-completeness result given that all operations used in the transformation are polynomial. \square

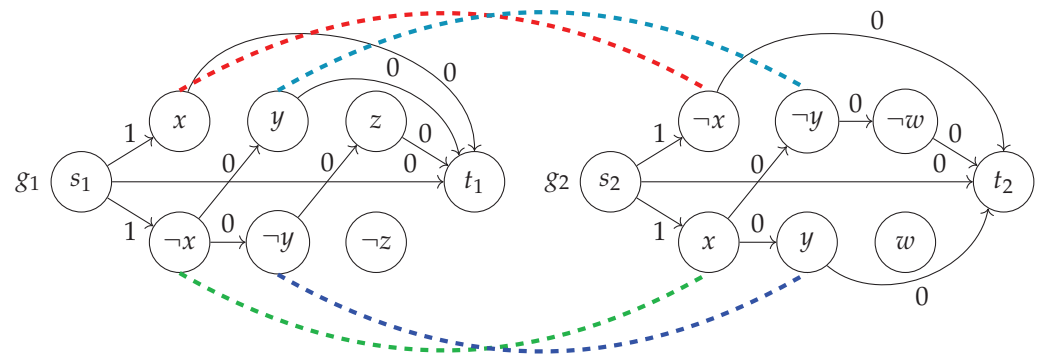


Figure 4. V-DPAP associated with the 3-SAT instance $(x \vee y \vee z) \wedge (\neg x \vee y \vee \neg w)$. Nodes in conflict are linked through a dashed edge.

Proposition 2. For a V-DPAP, it is NP-complete to decide whether there exists a valid allocation whose leximin evaluation is greater than or equal to a given utility vector. The proposition holds even if there is a unique graph per agent.

Proof. In the general case, it suffices to consider a problem involving a unique agent owning all the graphs, and to use the result of the previous proposition. If there is a unique graph per agent, it suffices to use the exact same 3-SAT encoding as before. Then, it is possible to show that there exists a valid allocation whose leximin evaluation is greater than or equal to $(1, 1, \dots, 1)$ if and only if there exists a solution for the 3-SAT problem. Furthermore, the leximin evaluation of an allocation π can be computed in polynomial time, hence the NP-completeness result. \square

5. R-DPAP: Resource-Constrained Directed Path Allocation Problems

The V-DPAP framework allows the posting of constraints on the simultaneous selection of items from different graphs. This is particularly relevant when the items correspond to tasks that require disjunctive resources over a given time frame. In this case, if two tasks i and j need to book the same resource over two time intervals $[ws(i), we(i)]$ and $[ws(j), we(j)]$, respectively, and if these two time intervals overlap, then a conflict $\{i, j\}$ can be defined. However, in practice, tasks i and j can be temporally flexible and can require the resource only during limited durations $d(i)$ and $d(j)$, respectively. In this case, even if time windows $[ws(i), we(i)]$ and $[ws(j), we(j)]$ overlap, tasks i and j may still be compatible. Such specifications are useful for our target application, where an agent may request a satellite only during 2 or 3 min over the whole 10 min pass of that satellite over the area of interest. This section introduces another extension of DPAP that is adapted to path allocation for items corresponding to such temporally flexible tasks.

5.1. Framework Definition

Definition 7. A Resource-Constrained Directed Path Allocation Problem (R-DPAP) is a DPAP $\langle \mathcal{A}, \mathcal{G}, \mu, \phi \rangle$ where function ϕ is defined by:

- a set of disjunctive resources $\mathcal{R} = \{r_1, \dots, r_p\}$;
- for each graph $g \in \mathcal{G}$, a triple (ws_g, we_g, c_g) such that:
 - $ws_g : V_g \rightarrow \mathbb{N}$ and $we_g : V_g \rightarrow \mathbb{N}$ associate a start date and an end date, respectively, that together define a time window for each item;
 - $c_g : V_g \rightarrow \mathcal{R} \cup \{r_\emptyset\}$ returns the resource required for each item. For any vertex $v \in V_g$, $c_g(v) = r_\emptyset$ indicates that v does not require any resource in \mathcal{R} . In particular, the source and sink nodes do not consume any resource. Moreover, we assume that for two items v and v' belonging to the same graph and requiring the same resource in \mathcal{R} , the time windows of v and v' do not overlap;
 - $d_g : V_g \rightarrow \mathbb{N}$ associates a duration with each item; resource $c_g(v)$ must be used during $d_g(v)$ time units within time window $[ws_g(v), we_g(v)]$ without any interruption (non-preemptive consumption).

From this, function ϕ returns a value of 1 for a path allocation if and only if, given the items selected by the paths, there exists a way to schedule the consumptions over the disjunctive resources in \mathcal{R} (see Definition 8).

Definition 8. In an R-DPAP $\langle \mathcal{A}, \mathcal{G}, \mu, \phi \rangle$, an allocation π is valid if and only if for each graph $g \in \mathcal{G}$, there exists a function $\tau_{\pi,g} : \pi(g) \rightarrow \mathbb{N}$ that assigns a start date to each node v in $\pi(g)$ such that:

- for all graphs $g \in \mathcal{G}$, for all nodes $v \in \pi(g)$, $\tau_{\pi,g}(v) \geq ws_g(v)$ and $\tau_{\pi,g}(v) + d_g(v) \leq we_g(v)$;
- there is no conflict for nodes in $\pi(g)$ with respect to resource consumption. Formally, for each pair of distinct graphs g and g' , for each node $v \in \pi(g)$ and each node $v' \in \pi(g')$ such that $c_g(v) = c_{g'}(v')$ and $c_g(v) \neq r_\emptyset$ (i.e., v and v' consume the same resource in \mathcal{R}), either $\tau_{\pi,g}(v) + d_g(v) \leq \tau_{\pi,g'}(v')$ or $\tau_{\pi,g'}(v') + d_{g'}(v') \leq \tau_{\pi,g}(v)$ holds.

Example 4. We reuse the orbit slot allocation problem whose graph is given in Figure 1, and where two requests **a** and **b** are involved. We assume here that each request requires two observation slots of duration 2. For both requests, the first slot must occur around time 3 and the second slot around time 9. We consider two satellites sat_1 and sat_2 . For request **a**, there are two time windows around time 3 during which satellites pass over the target area of **a**: time window $a_1 = [1, 4]$ for satellite sat_1 and time window $a_3 = [2, 4]$ for satellite sat_2 . Around time 9, only satellite sat_1 passes over the target area, which results in time window $a_2 = [7, 10]$. Similarly, for request **b**, time windows $b_1 = [2, 5]$ and $b_3 = [1, 4]$ allow the target area to be observed around time 3 with satellites sat_1 and sat_2 , respectively. Time windows $b_2 = [8, 10]$ and $b_4 = [9, 12]$ are available for observing around time 9. Such a problem can be represented through the R-DPAP illustrated in Figure 5. Each satellite can be seen as a resource. Each request is represented through a graph: graph g_a for request **a** and graph g_b for request **b**. The nodes in the graph correspond to the time windows associated with each request and each satellite. For instance, node a_1 in graph g_a corresponds to time window $a_1 = [1, 4]$. Formally, $d_{g_a}(a_1) = 2$ (because an observation duration equal to 2 is required), $ws_{g_a}(a_1) = 1$, $we_{g_a}(a_1) = 4$ (corresponding to time window $[1, 4]$) and $c_{g_a}(a_1) = sat_1$.

Example 5. Figure 6a illustrates an allocation for the problem presented in Example 4. A valid allocation could be $\pi_{ex} = \{g_a \mapsto \{s_a, a_1, a_2, t_a\}, g_b \mapsto \{s_b, b_1, b_4, t_b\}\}$. In fact, we can consider two functions τ_{π_{ex},g_a} and τ_{π_{ex},g_b} that assign start dates to nodes in π_{ex} without any conflict in resources. As illustrated in Figure 6b, it is possible to have $\tau_{\pi_{ex},g_a}(a_1) = 1$ (i.e., a slot starting at time 1 and ending at time 3 is booked within time window a_1), $\tau_{\pi_{ex},g_a}(a_2) = 7$, $\tau_{\pi_{ex},g_b}(b_1) = 3$, and $\tau_{\pi_{ex},g_b}(b_4) = 9$, which results in a non-conflicting access to the resources sat_1 and sat_2 .

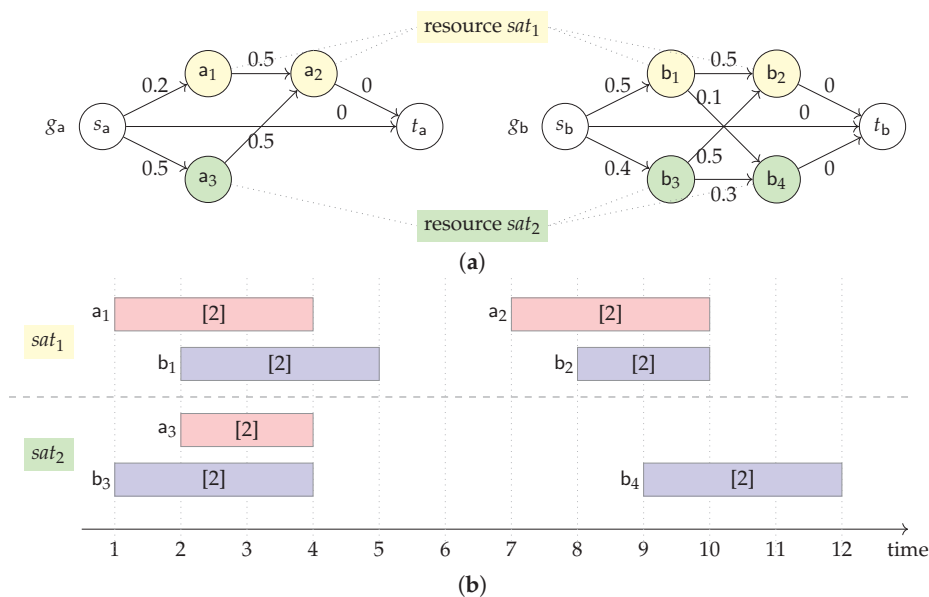


Figure 5. Orbit slot allocation problem involving two satellites sat_1 and sat_2 and two agents A and B that each have one request, denoted a and b, respectively. Two observation slots with a duration equal to 2 must be allocated for each request (represented by [2] in each observation slot). The first orbit slot of each request should be around time 3 and the second one around time 9. (a) Graphs g_a and g_b representing the requests and resources of Example 4. (b) Description of the resources, time windows, and durations associated with the vertices of graphs g_a and g_b .

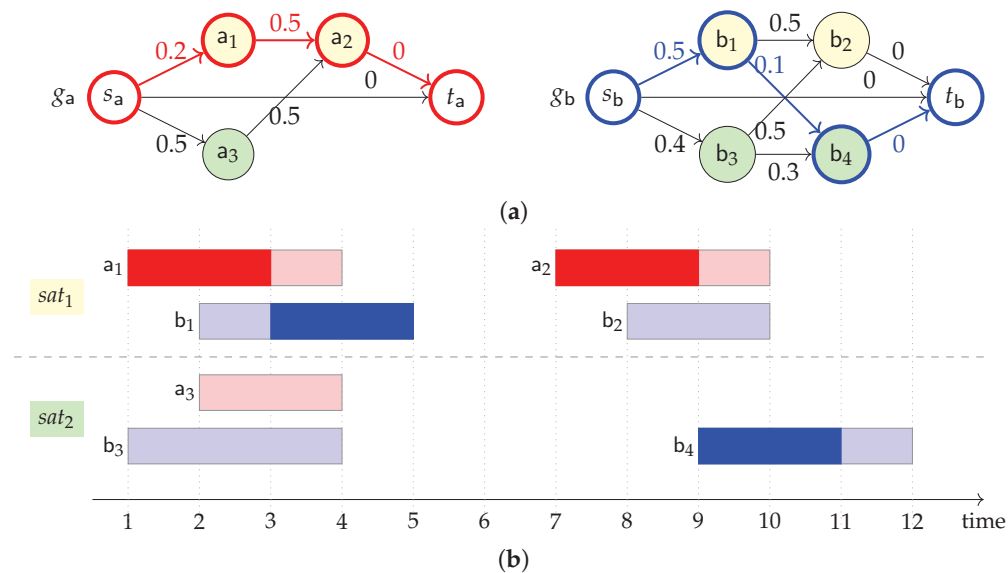


Figure 6. Valid allocation example, π_{ex} , for the R-DPAP described in Example 4. (a) Illustration of allocation π_{ex} with the paths selected for graphs g_a and g_b . (b) Start dates that allow the selection of the nodes of π_{ex} without any conflict in resources.

5.2. Theoretical Complexity

Proposition 3. The R-DPAP-UTIL-DEC problem, which consists in determining whether, for a given R-DPAP problem, there exists an allocation π and start time functions $\tau_{\pi,g}$ such that π is valid and utilitarian evaluation $u(\pi)$ is greater than or equal to a given value, is NP-complete.

Proof. Given an R-DPAP, an allocation π for it, a start time function $\tau_{\pi,g}$ for each graph g , and a utility lower bound L , verifying that the scheduling constraints are satisfied and that

$u(\pi)$ is greater than or equal to L is polynomial. This proves that R-DPAP-UTIL-DEC is in class NP.

To prove the NP-completeness of R-DPAP-UTIL-DEC, we rely on the fact that the one-machine scheduling problem with release dates and due dates in which the objective is to minimize the maximum lateness of jobs is NP-complete [21].

Let $\langle A, P, R, D \rangle$ be such a problem where:

- $A = \{a_1, \dots, a_n\}$ is a set of activities;
- $P : A \rightarrow \mathbb{N}$ is a function that assigns a processing time to each activity of A ;
- $R : A \rightarrow \mathbb{N}$ is a function that assigns a release date to each activity of A ;
- $D : A \rightarrow \mathbb{N}$ is a function that assigns a due date to each activity of A .

The objective of the problem is to define a function $\sigma : A \rightarrow \mathbb{N}$ that assigns a start date $\sigma(a)$ to each activity a in A such that:

- the release dates are satisfied, i.e., $\forall a \in A, \sigma(a) \geq R(a)$;
- the machine performs at most one activity at each time step, i.e., $\forall a_i, a_j \in A^2$ with $i \neq j$, either $\sigma(a_i) + R(a_i) \leq \sigma(a_j)$ or $\sigma(a_j) + R(a_j) \leq \sigma(a_i)$ holds;
- the maximum lateness L_{max} is minimized, where $L_{max} = \max_{i=1}^n (\sigma(a_i) + P(a_i) - D(a_i))$.

In the associated decision problem, we consider a bound l , and the objective is to decide if it is possible to define σ such that $L_{max} \leq l$.

Such a problem can be transformed to an R-DPAP as follows:

- we consider a unique resource r ;
- we consider an agent $agent_a$ for each activity a in A ;
- for each activity a in A , we consider the graph g_a (illustrated in Figure 7a) that belongs to agent $agent_a$ and that has the following features:
 - its set of vertices is composed of three nodes: s_a, t_a , and v_a ;
 - its set of edges is composed of (s_a, v_a) with a utility equal to 1, and $(s_a, t_a), (v_a, t_a)$ that both have a null utility;
 - as illustrated in Figure 7b, node v_a requires resource r during $D(a)$ time units within time window $[R(a), D(a) + l]$;
- the obtained R-DPAP is $\langle \{agent_a \mid a \in A\}, \{r\}, \{g_a \mid a \in A\}, \mu \rangle$, with μ a function that assigns, for each activity a in A , agent $agent_a$ to graph g_a .

The maximum lateness is lower than or equal to l in the machine scheduling problem if and only if there exists a valid allocation π with a utility greater than or equal to n . Indeed, to reach such a utility value, the paths selected in the n graphs must each have a utility equal to 1. The selection of such paths indicates that all activities v_a can be scheduled on the unique resource while satisfying the release date and the due date, to which is added the lateness bound l . □

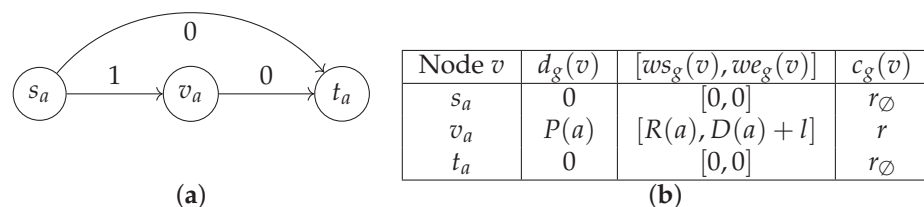


Figure 7. R-DPAP part generated for each activity a in A . (a) Graph generated for each activity a in A . (b) Description of nodes in the graph generated for each activity a in A .

Proposition 4. The R-DPAP-LEX-DEC problem, which consists in determining whether, for a given R-DPAP problem, there exists an allocation π and start time functions $\tau_{\pi,g}$ such that π is valid and its leximin evaluation is greater than or equal to a given utility vector, is NP-complete.

Proof. By using the same encoding as in the previous proof, there exists a solution such that $L_{\max} \leq l$ if and only if the leximin-optimal allocation has a value greater than or equal to $(1, 1, \dots, 1)$. Further, the leximin evaluation of an allocation π can be computed in polynomial time, hence the NP-completeness result. \square

5.3. Relationship between R-DPAP and V-DPAP

An R-DPAP combines a path selection problem and a scheduling problem over the resources used by the selected items. In the following, we show that it is possible to transform an R-DPAP into an equivalent V-DPAP by generating a set of item selection conflicts that is equivalent to the set of selections forbidden by the scheduling problem.

To illustrate this point, let us consider the example given in Figure 8, that involves four requests: a, b, c, d . It is first possible to decompose the scheduling problem of the R-DPAP into a set of subproblems containing items that may be in competition for using a given resource (gray rectangles depicted in the figure). For example, items a_4 and b_5 belong to the same subproblem because their time windows overlap, and items b_5 and c_1 , whose time windows do not overlap, also belong to the same subproblem because the presence of items a_4 and d_1 creates an indirect interaction between b_5 and c_1 . More formally, to compute the content of these scheduling subproblems, we can build, for each resource r , the graph G_r containing one node per item and one edge between item i and item j if and only if the time windows of i and j overlap. Then, the scheduling subproblems to consider correspond to the connected components of graph G_r . In Figure 8, we obtain three connected components for resource sat_1 , namely, $\{a_1, b_1\}$, $\{a_2, b_2\}$, and $\{a_4, b_5, c_1, d_1\}$, and three connected components for resource sat_2 , namely, $\{a_3, b_3\}$, $\{b_4\}$, and $\{a_5, b_6, c_2, d_2\}$.

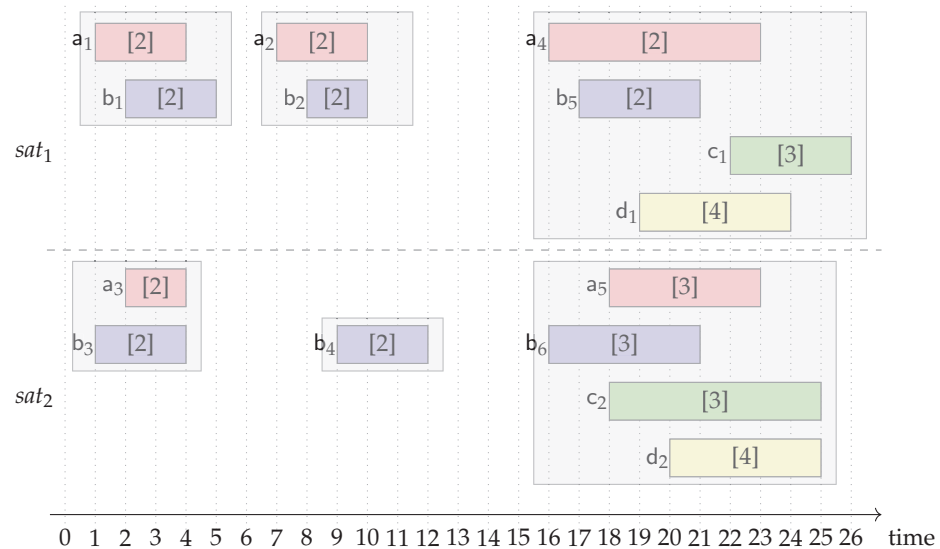


Figure 8. Orbit slot allocation problem involving two satellites sat_1 and sat_2 and four requests a, b, c, d posted by four agents A, B, C, D ; the duration associated with each item is also indicated (e.g., a duration of 2 time units for item a_1 and a duration of 3 time units for item c_1).

After these steps, for each component Γ obtained, we can compute the set of *minimal scheduling conflicts* associated with Γ . This set contains all the sets $S \subseteq \Gamma$ such that, (1) there is no feasible schedule performing all the tasks in S while respecting their time window and duration constraints, and (2) set S is minimal for inclusion, that is, for every set $S' \subset S$, there exists a way to schedule all the tasks in S' . To compute these minimal conflicts, we proceed as follows.

- We consider the non-empty subsets S of Γ one by one, following an increasing cardinality order. For a given set S , if there exists a subset $S' \subset S$ of size $|S| - 1$ such that S' is a conflict, S is marked as being a conflict but is not added to the set of minimal

- conflicts. Otherwise, we test whether there exists a schedule containing all the tasks in S . If not, S is marked as a conflict and added to the set of minimal conflicts.
- To determine whether there exists a schedule containing all the items in a set S , we use a dynamic programming algorithm. More precisely, we consider the subsets S' of S following an increasing cardinality order and we determine, for each subset $S' \subseteq S$, the minimum time $mt(S')$ at which all items in S' can be served in a feasible schedule. To do this, we start from $mt(\emptyset) = -\infty$ and apply recursive formulas. If item $i \in S'$ belongs to graph g and is the last item visited, the minimum time at which the visit of i can end is given by $mt(S', i) = \max(mt(S' \setminus \{i\}), ws_g(i)) + d_g(i)$, and visiting i at the latest position among the items in S' is feasible if and only if $mt(S', i) \leq we_g(i)$. From this, the minimum time $mt(S')$ at which all items in S' can be served in a feasible schedule is given by $mt(S') = \min_{i \in S' \mid mt(S', i) \leq we_g(i)} mt(S', i)$. It can be shown that at the end of the process, all the items in S can be scheduled if and only if $mt(S) < +\infty$. The dynamic programming algorithm described before has a time complexity that is exponential in the size of S ; however, the number of requests is low for the practical application we are targeting.

Example 6. For the example given in Figure 8, the set of minimal conflicts obtained is

$$\{\{a_2, b_2\}, \{a_4, b_5, c_1, d_1\}, \{a_3, b_3\}, \{a_5, b_6, d_2\}, \{b_6, c_2, d_2\}\}$$

Such conflicts are equivalent to the constraints of the initial scheduling problem.

The method described before allows us to transform an R-DPAP \mathcal{P} into a V-DPAP \mathcal{P}' that contains the exact same set of items as \mathcal{P} and has the same graph topology as \mathcal{P} , and where the conflicts in \mathcal{P}' are those obtained by preprocessing the scheduling problem of \mathcal{P} . In the following, given the (restricted) number of requests in our target application, we consider that such a transformation from R-DPAP to V-DPAP can be used and we focus on the definition of algorithms for solving V-DPAP.

6. V-DPAP Solution Methods

We propose here several allocation schemes for V-DPAP. Some of them are based on integer linear programming (ILP) and mixed integer linear programming (MILP), so we first introduce decision variables and constraints for these models. For any DAG $g = \langle V_g, E_g, u_g \rangle$, we define binary variables $x_e \in \{0, 1\}$, for any $e \in E_g$, stating whether edge e is selected in the path defining the solution bundle. We also use auxiliary binary variables β_v , stating whether node v is selected in solution path $\pi(g)$, i.e., $\beta_v = 1$ if $v \in \pi(g)$, and 0 otherwise. For any node v in V_g , we denote by $\text{In}(v)$ (respectively $\text{Out}(v)$) its set of incoming (respectively outgoing) edges. In all ILP models introduced hereafter, we impose constraints (1)–(3) to define all the possible paths, (4) and (5) to account for item selection conflicts, (6) to ensure that sources and sinks are selected, and (7) to define the edge selection variables.

$$\sum_{e \in \text{In}(v)} x_e = \sum_{e \in \text{Out}(v)} x_e, \quad \forall g \in \mathcal{G}, \forall v \in V_g \setminus \{s_g, t_g\} \tag{1}$$

$$\sum_{e \in \text{Out}(s_g)} x_e = 1, \quad \forall g \in \mathcal{G} \tag{2}$$

$$\sum_{e \in \text{In}(t_g)} x_e = 1, \quad \forall g \in \mathcal{G} \tag{3}$$

$$\sum_{e \in \text{In}(v)} x_e = \beta_v, \quad \forall g \in \mathcal{G}, \forall v \in V_g \setminus \{s_g, t_g\} \tag{4}$$

$$\sum_{v \in \sigma} \beta_v \leq |\sigma| - 1, \quad \forall \sigma \in \mathcal{C} \tag{5}$$

$$\beta_{s_g} = \beta_{t_g} = 1, \quad \forall g \in \mathcal{G} \tag{6}$$

$$x_e \in \{0, 1\}, \quad \forall a \in \mathcal{A}, \forall g \in \mathcal{G}_a, \forall e \in E_g \tag{7}$$

6.1. Utilitarian Allocation (util)

The classical approach to allocation is the utilitarian one. This consists in finding the allocation that maximizes the sum of utilities of all selected paths. This corresponds to solving the integer linear program $P_{\text{util}}(\langle \mathcal{A}, \mathcal{G}, \mu, \mathcal{C} \rangle)$ composed of constraints (1)–(7) and the objective function given below:

$$\text{maximize} \quad \sum_{a \in \mathcal{A}} \sum_{g \in \mathcal{G}_a} \sum_{e \in E_g} u_g(e) \cdot x_e \tag{8}$$

The resulting allocation π is decoded from the β_v variables. Formally, for all $g \in \mathcal{G}$, $\pi(g) = \{v \in V_g \mid \beta_v = 1\}$.

Example 7. In Figure 3, the utilitarian allocation is $\pi_{\text{util}} = \{a \mapsto \{s_a, a_3, a_2, t_a\}, b \mapsto \{s_b, b_1, b_4, t_b\}\}$, with utility $u(\pi_{\text{util}}) = u_a(\pi_{\text{util}}) + u_b(\pi_{\text{util}}) = 1.0 + 0.6 = 1.6$.

6.2. Leximin Allocation (lex)

Beyond utilitarianism, one way to implement fair allocation and Pareto-optimality is to consider the *leximin* rule, that selects, among all possible allocations, an allocation leading to the best utility profiles with respect to the leximin order [22]. More precisely, let $z = [z_1, \dots, z_n]$ be the utility vector, where each component $z_a \in [0, Z_a]$ represents the utility for agent $a \in \mathcal{A}$. Z_a denotes here the best utility value for user a considered alone, i.e., for the mono-agent problem, where the best path can be chosen for each graph $g \in \mathcal{G}_a$. In leximin optimization, the objective is to lexicographically maximize vector $\Lambda = [\Lambda_1, \dots, \Lambda_n]$ obtained after ordering $[z_1, \dots, z_n]$ following an increasing order. Such a leximin rule can be implemented through a sequence of ILP [23]. We adapt here such a procedure to the specific case of V-DPAP. Suppose we have already optimized over the first $K - 1$ components $[\Lambda_1, \dots, \Lambda_{K-1}]$ of Λ , for $K \in [1..n]$. Then, one can use the MILP presented thereafter to optimize the K^{th} component Λ_K of the leximin profile.

In this MILP model, variable λ represents the utility optimized at level K in Λ , with $\lambda \in [\Lambda_{K-1}, \max_{a \in \mathcal{A}} Z_a]$, using convention $\Lambda_0 = 0$. Variable y_{ak} is a binary variable that takes value 1 if agent $a \in \mathcal{A}$ plays the role of the agent associated with level $k \in [1..K - 1]$ in $[\Lambda_1, \dots, \Lambda_{K-1}]$, and 0 otherwise. Constraint (10) computes the utility associated with each agent. Constraints (11) and (12) ensure that a unique agent is associated with each level $k \in [1..K - 1]$ already dealt with. Constraint (13) ensures that the utility obtained for the agent associated with level $k \in [1..K - 1]$ must not be less than Λ_k . Last, together with the objective function, Constraint (14) ensures that λ will be equal to the minimum utility value obtained for the agents that are not associated with levels $[1..K - 1]$ in Λ . In this constraint, $M = \max_{a \in \mathcal{A}} Z_a$ is used to ignore the agents associated with levels strictly lower than K when optimizing λ (big-M formulation). In the end, the optimization of Λ_K can be performed using program $P_{\text{lex}}(\langle \mathcal{A}, \mathcal{G}, \mu, \mathcal{C} \rangle, K, [\Lambda_1, \dots, \Lambda_{K-1}])$ that is composed of constraints (1)–(7) and the additional constraints and objective function given below:

$$\text{maximize} \quad \lambda \tag{9}$$

$$z_a = \sum_{g \in \mathcal{G}_a} \sum_{e \in E_g} u_g(e) \cdot x_e, \quad \forall a \in \mathcal{A} \tag{10}$$

$$\sum_{a \in \mathcal{A}} y_{ak} = 1, \quad \forall k \in [1..K - 1] \tag{11}$$

$$\sum_{k \in [1..K-1]} y_{ak} \leq 1, \quad \forall a \in \mathcal{A} \tag{12}$$

$$z_a \geq \sum_{k \in [1..K-1]} \Lambda_k \cdot y_{ak}, \quad \forall a \in \mathcal{A} \tag{13}$$

$$\lambda \leq z_a + M \sum_{k \in [1..K-1]} y_{ak}, \quad \forall a \in \mathcal{A} \tag{14}$$

$$z_a \in [0, Z_a], \forall a \in \mathcal{A} \tag{15}$$

$$y_{ak} \in \{0, 1\}, \forall a \in \mathcal{A}, \forall k \in [1..K - 1] \tag{16}$$

$$\lambda \in [\Lambda_{K-1}, \max_{a \in \mathcal{A}} Z_a] \tag{17}$$

To implement the leximin rule, it then suffices to solve a sequence of P_{lex} problems for $K \in \mathcal{A}$ to optimize the value of each component of the utility profile, as presented in Algorithm 1.

Algorithm 1: Leximin algorithm.

Data: A V-DPAP $\langle \mathcal{A}, \mathcal{G}, \mu, \mathcal{C} \rangle$

Result: A leximin-optimal path allocation π

```

1 for  $K = 1$  to  $|\mathcal{A}|$  do
2    $(\lambda^*, \text{sol}) \leftarrow \text{solve } P_{\text{lex}}(\langle \mathcal{A}, \mathcal{G}, \mu, \mathcal{C} \rangle, K, [\Lambda_1, \dots, \Lambda_{K-1}])$ 
3    $\Lambda_K \leftarrow \lambda^*$ 
4 for  $g \in \mathcal{G}$  do  $\pi(g) \leftarrow \{v \in V_g \mid \text{sol}(\beta_v) = 1\}$ 
5 return  $\pi$ 

```

Example 8. For the example in Figure 3, the leximin-optimal allocation is $\pi_{\text{lex}} = \{g_a \mapsto \{s_a, a_1, a_2, t_a\}, g_b \mapsto \{s_b, b_3, b_4, t_b\}\}$, with utility vector $(u_A(\pi_{\text{lex}}), u_B(\pi_{\text{lex}})) = (0.7, 0.7)$.

6.3. Approximate Leximin Allocation (a-lex)

The previous model implements an exact leximin rule, and thus enforces fairness in the resulting allocation. However, it may not scale well when increasing the number of agents and edges. This is why we provide an approximate version of the computation of the leximin based on an iterated maximin scheme. This approach considers at each step a minimum utility $\Delta_a \geq 0$ for some agents and maximizes the worst utility among the remaining agents, for which we arbitrarily assume $\Delta_a = -1$. The problem to solve, referred to as $P_{\text{a-lex}}(\langle \mathcal{A}, \mathcal{G}, \mu, \mathcal{C} \rangle, \Delta)$, is the following one:

$$\text{maximize } \delta \tag{18}$$

$$\text{such that } (1), (2), (3), (4), (5), (6), (7)$$

$$\delta \leq \sum_{g \in \mathcal{G}_a} \sum_{e \in E_g} u_g(e)x_e, \quad \forall a \in \mathcal{A} \mid \Delta_a = -1 \tag{19}$$

$$\sum_{g \in \mathcal{G}_a} \sum_{e \in E_g} u_g(e)x_e \geq \Delta_a, \quad \forall a \in \mathcal{A} \mid \Delta_a \neq -1 \tag{20}$$

$$\delta \in \mathbb{R}^+ \tag{21}$$

The solution method then consists in optimizing in an iterative manner, as for leximin. As sketched in Algorithm 2, at each iteration (one per agent), $P_{\text{a-lex}}$ is solved, one worst agent \hat{a} is determined, and its minimum utility $\Delta_{\hat{a}}$ is fixed. The main difference with P_{lex} , is that at each iteration in $P_{\text{a-lex}}$ the position of an agent in the order is implicitly determined once for the whole algorithm, while in P_{lex} the order can be revised at each iteration. Moreover, if any equality occurs at line 5 to determine the worst agent (case $|S| > 1$), one may rely on some heuristic or arbitrary choice. Thus, $P_{\text{a-lex}}$ is an approximation of P_{lex} that contains fewer variables and constraints.

Example 9. The approximate leximin allocation for the example in Figure 1 is $\pi_{\text{a-lex}} = \{g_a \mapsto \{s_a, a_1, a_2, t_a\}, g_b \mapsto \{s_b, b_3, b_4, t_b\}\}$, with utility vector $(u_A(\pi_{\text{a-lex}}), u_B(\pi_{\text{a-lex}})) = (0.7, 0.7)$. This is the same as π_{lex} , but in the general case, $\pi_{\text{a-lex}}$ and π_{lex} can differ.

Algorithm 2: Approximate leximin algorithm.

Data: A V-DPAP $\langle \mathcal{A}, \mathcal{G}, \mu, \mathcal{C} \rangle$
Result: An iterated maximin-optimal allocation π

- 1 $\Delta \leftarrow [-1, \dots, -1]$
- 2 **for** $K = 1$ to $|\mathcal{A}|$ **do**
- 3 $(\delta^*, sol) \leftarrow \text{solve } P_{a\text{-lex}}(\langle \mathcal{A}, \mathcal{G}, \mu, \mathcal{C} \rangle, \Delta)$
- 4 $S \leftarrow \underset{a \in \mathcal{A} \mid \Delta_a = -1}{\text{argmin}} \sum_{g \in \mathcal{G}_a} \sum_{e \in E_g} u_g(e) sol(x_e)$
- 5 $\hat{a} \leftarrow$ choose an agent a in S
- 6 $\Delta_{\hat{a}} \leftarrow \delta^*$
- 7 **for** $g \in \mathcal{G}$ **do** $\pi(g) \leftarrow \{v \in V_g \mid sol(\beta_v) = 1\}$
- 8 **return** π

6.4. Greedy Allocation (greedy)

For very fast decisions, approximate leximin might still be too slow. In such cases, a greedy approach can quickly provide valid allocations. The main idea of greedy path allocation is to iterate over the set of graphs. At each step, one graph g^* that has the best utility path is selected and this path is chosen as $\pi(g^*)$. Moreover, given the nodes already selected and the new ones in $\pi(g^*)$, all the nodes in the other graphs that are in conflict are deactivated. Graph g^* is then removed, and the process continues until there is no more graphs to consider. This process ensures that constraints (1)–(6) are met. Determining the best path in a DAG g has a linear time complexity $\mathcal{O}(|E_g| + |V_g|)$ [24]. Obviously, greedy is equivalent to utilitarian when there is no conflict between graphs. Indeed, greedy will return the best path for each graph, which is the best utilitarian solution in such settings. Moreover, if there are no ties when selecting the best path for each graph, then this greedy approach leads to a Nash equilibrium, where no agent can improve its utility without a negative impact on other agents. This is equivalent to the *Nashify* procedure from [13] in the context of congestion games, with only one turn. We will see in the experiments that this equilibrium is far from being fair.

Example 10. For the example in Figure 3, there is a path of value 1 in the two graphs g_a and g_b . If the best path in g_a is chosen first, then the allocation obtained in the end is $\pi_{\text{greedy}} = \{g_a \mapsto \{s_a, a_3, a_2, t_a\}, g_b \mapsto \{s_b, b_1, b_4, t_b\}\}$, with global utility $u(\pi_{\text{greedy}}) = u_A(\pi_{\text{greedy}}) + u_B(\pi_{\text{greedy}}) = 1.0 + 0.6 = 1.6$ and utility vector $(1.0, 0.6)$.

6.5. Round-Robin Allocations (p-rr and n-rr)

One fast approach to the fair allocation of indivisible goods is *round-robin*. This consists in making each agent choose in turn (in a predefined fixed order) one item (depending on the preferences) until there is no more item to allocate. It is polynomial in the number of agents and items. In our case, one may consider two kinds of items to allocate: paths (denoted p-rr) or nodes (denoted n-rr). In the case of paths, each agent selects at its turn its best feasible path, given the already allocated nodes (to prevent conflicts). This process operates similarly to greedy, but alternates between users to balance utilities. In the case of nodes, each agent incrementally builds the path associated with each of its graphs, by choosing in turn a next best feasible node until either the sink is reached or there is no more feasible nodes to choose (dead-end path). In the latter case, the agent is allocated the 0-utility source-to-sink path and loses the previously chosen nodes. In both approaches, constraints (1)–(6) are met since all the paths considered are feasible. Note that if there are no ties for the best path chosen by an agent at its turn, then p-rr results in a Nash equilibrium. This is not true for n-rr, since some nodes left by some agents reaching a dead-end may have prevented some other agents from finding a better solution. To overcome this difficulty, it is possible to increase the possible partial satisfaction schemes for a request, e.g., by adding arcs with a null utility from any node v to the sink node.

Example 11. In Figure 3, if request a begins the path round-robin allocation, π_{p-rr} for the example in Figure 3 is equivalent to π_{greedy} , since a chooses $\{s_a, a_3, a_2, t_a\}$ and then b chooses $\{s_b, b_1, b_4, t_b\}$. If request b begins, then b chooses $\{s_b, b_1, b_2, t_b\}$ and then the only possible path for a is $\{s_a, t_a\}$, meaning that agent A receives a null utility.

If request a begins, the node round-robin allocation π_{n-rr} is equivalent to π_{greedy} because a first chooses a_3 , then b chooses b_1 (only feasible option), then a chooses a_2 , and finally b chooses b_4 (only feasible option). However, if request b begins, b first chooses b_1 , then a chooses a_3 , then b chooses b_2 , and finally a reaches a dead-end, since the selection of b_2 implies that a_2 cannot be selected.

7. Experimental Evaluation

In this section, we evaluate the different allocation methods proposed when applied to orbit slot allocation problems encoded as V-DPAP or R-DPAP. We present the experimental setup and analyze some results obtained on synthetic realistic instances. In addition to the experimental evaluation, this section also illustrates how a concrete application can be modeled in our theoretical framework.

7.1. Benchmarks

We first describe the benchmark generation in the case of orbit slot allocation problems.

7.1.1. Constellation and Requests Features

We consider a low-Earth-orbit constellation (500 km altitude) composed of n_p regularly spaced orbital planes having a 40-degree inclination, with $n_p \in \{2, 4, 8, 16\}$ and two regularly spaced satellites over each orbital plane (Walker constellation). We randomly generate requests for four agents wishing to obtain orbit slot ownerships to implement some repetitive ground acquisitions of POIs belonging to the same area. POIs are randomly selected within an extracted subset from [25], around Grenoble, France. All the agents have the same template for each request r , that is, communicating and getting observations every day at three requested times (RTs): $8:00 + \delta_r$, $12:00 + \delta_r$, and $16:00 + \delta_r$, where δ_r is uniform random time shift in $[-2h, 2h]$. Note that δ_r applies to all RTs of the same request. For each request r and each RT t for r , the slots over which orbit ownership can be claimed for achieving r around time t are determined thanks to a space mechanics toolbox, based on the assumption that a satellite is relevant for a POI as soon as its elevation above the horizon is greater than 15 degrees. Depending on the number of satellites in the constellation, there might not be a satellite passing over a POI exactly at the RT. We consider a tolerance window Δ equal to 1 h before and after each RT, meaning that an orbit slot is considered as valid for an RT t if the middle of its temporal window is less than an hour from that RT. Finally, we impose a minimum duration $minD$ of 120 s for all requests and do not consider orbit slots whose duration is shorter than this duration.

Note that these features were validated as realistic by a satellite constellation manager we work with. In fact, in the case of orbit slot allocation problems, the number of users that can afford to own orbit slots is quite low and so is their number of requests.

7.1.2. From Requests to DAGs

In order to encode the problem within the DPAP framework, we first create an agent u for each user that has an observation request. Then, for each request r associated with agent u , we first create a graph g_r and define a function μ such that $\mu(g_r) = u$. In a graph g_r created for request r , the nodes are the orbit slots usable for capturing the POI targeted by r at some RT, and the edges link two such consecutive orbit slots. We also add a source node that precedes all of the orbit slots of the earliest RT and a sink node that follows all of the orbit slots of the latest RT. Consequently, a path in the graph (i.e., a sequence of consecutive orbit slots) represents a way to satisfy r . Figure 8 represents four requests (a , b , c , and d) from four users (respectively, A , B , C , and D), with three RTs that are time 3, time 9, and time 21. In this example, each RT has at most two possible orbit slots per request. For instance, for request a , there are two orbit

slots for RT 3 (a_1 and a_3), one orbit slot for RT 9 (a_2), and two orbit slots for RT 21 (a_4 and a_5). Requests c and d do not have any orbit slot for RTs 3 and 9 but two each for RT 21.

For simplicity, even if the incoming arcs in the graphs of a DPAP for a given slot can have different weights, we only consider in our experiments utilities attached to the slots and not to the transitions between slots. As illustrated in Figure 9, for each candidate orbit slot for a given RT, we consider a utility function that is piecewise linear in the distance between the middle τ of the slot and that RT. The utility linearly decreases from 1 when τ is exactly on the RT to 0.25 when τ reaches the bounds of the tolerance window, i.e., $RT + \Delta$ and $RT - \Delta$. We normalize each utility with respect to the maximum utility that can be achieved for each user individually along by using its best paths. Therefore, each user’s set of best paths has a utility equal to 1.

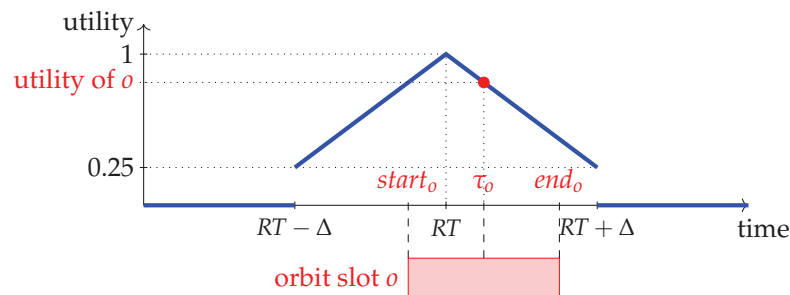


Figure 9. Utility function used to compute the utility of an orbit slot with respect to some RT and a tolerance Δ .

In order to limit the number of edges in graphs, we add a virtual node between all slots of one RT and all slots of the next RT. If there are n orbit slots for an RT t and m slots for the next RT t' , this allows there to be $n + m$ edges (n edges with utility 0 going into the virtual node and m edges weighted by the utility of orbit slots going out of the virtual node) instead of $n \cdot m$ edges (all n nodes connected to all m nodes).

Last, we consider two variants of the problem depending on whether requests can be partially satisfied.

- In the *full satisfaction* variant, each path goes through one orbit slot for each RT, except for a specific direct source-to-sink path that allows us to guarantee that there exists at least one feasible path for each request. In other words, it is not possible to skip one RT for an observation request, unless this request is not served at all.
- In the *partial satisfaction* variant, it is possible to skip some RTs for a request. In terms of generated graphs, it simply consists in adding edges with a null utility between successive virtual nodes, between the source and the first virtual node and between the last virtual node and the sink.

Figure 10 illustrates the request a of Figure 8 in a full configuration (only black edges) and in a partial configuration (black and thick blue edges).

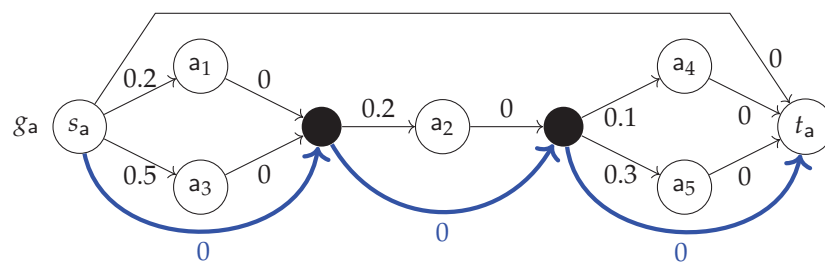


Figure 10. Graph for request a of Figure 8 with virtual nodes between successive RTs. The graph with only black edges represents the problem in full satisfaction mode. The graph with both black and thick blue edges represents the problem in partial request satisfaction mode.

7.1.3. V-DPAP and R-DPAP Generation

We describe here how the function ϕ is implemented in the case of the orbit slot problem for generating V-DPAP and R-DPAP instances.

V-DPAP. For generating the set of conflicts \mathcal{C} associated with a V-DPAP, we define a conflict for each pair of nodes corresponding to orbit slots that: 1. belong to the same satellite; 2. temporally overlap; and 3. are from different users. For the last assumption, we consider that it is possible to allocate to some user two orbit slots from the same satellite that overlap. In fact, as the allocation of an orbit slot consists in allowing an agent to dispose of the satellite during the associated time interval, two overlapping orbit slots o_1 and o_2 can be seen as a unique orbit slot that is the union of o_1 and o_2 . With this conflict generation scheme, all the conflicts obtained are binary. Note that it would be possible to compute these conflicts more finely, for instance, by following the approach proposed in [26].

R-DPAP. In the case of the ϕ function in R-DPAP, we follow the same process as in Example 4. More precisely, we create a resource ρ_s for each satellite s of the constellation. Then, for each graph g_r associated with an observation request r , for each vertex v in V_{g_r} that corresponds to an orbit slot o (i.e., all vertices except source, sink, and the ones added between successive RTs), we define $ws_{g_r}(v) = start_o$, $we_{g_r}(v) = end_o$ (i.e., the temporal window associated with vertex v is exactly the temporal window associated with orbit slot o), $d_{g_r}(v) = minD$ (i.e., the duration associated with v is the minimum duration required) and $c_{g_r}(v) = \rho_s$ where s is the satellite associated with orbit slot o . For each vertex v that is a virtual node, we consider that $ws_{g_r}(v) = we_{g_r}(v) = 0$, $d_{g_r}(v) = 0$, and $c_{g_r}(v) = r_{\emptyset}$. Note that R-DPAPs are next transformed into V-DPAPs as explained in Section 5.3, and in this case the conflicts obtained are not necessarily binary ones.

7.1.4. Instance Generation Parameters and Properties

Table 1 summarizes all the parameters used for configuring the instances. Some of these parameters do not vary, e.g., the number of requests for each user, which is equal to two. The parameters that have different values as per configuration are the number of orbital planes n_p , the type of problem (V-DPAP or R-DPAP), and the request mode satisfaction (full or partial). For each configuration, 100 random instances have been generated. For 2 requests per agent, 3 RTs per day, and a horizon $h = 180$ days, the DAGs generated contain $3 \cdot (2h - 1) = 1077$ layers. These settings generate DAGs having the features displayed in Table 2.

7.1.5. Experimental Conditions

Our experimental environment has been implemented in Java 1.8 and executed on a 20-core Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60 GHz, 62 GB RAM, Ubuntu 18.04.5 LTS. Utilitarian, leximin, and approximate leximin make use of the Java API of IBM CPLEX 20.1 (using a 2 min timeout). Note that the computation time does not need to be as tight as in Earth observation scheduling problems. In fact, in EOSPs, it might be operationally required to generate a schedule within a few minutes. Such operational constraints are not relevant for orbit slot allocation problems as plans are computed months in advance. Nevertheless, we limit the time taken by each call to the MILP solver.

Table 1. Generation parameters along with their possible values for configuring instances.

Generation Parameters		Values
Constellation	Altitude	500 km
	Number of orbital planes n_p	2, 4, 8, 16
	Number of satellites/plane	2
	Inclination	40°
Scheduling horizon	Start	1 January 2020
	Duration	180 days
Problems	Number of users	4
	Type	V-DPAP, R-DPAP
Requests	Number of requests/user	2
	Requested observation Times	3 RTs/request
	Maximum random time shift δ_r	1 h
	Tolerance Δ	1 h
	Minimum slot duration $minD$	120 s
Algorithms	Satisfaction mode	full, partial
	Type	util, lex, a-lex, greedy, p-rr, n-rr
	CPLEX time limit	120 s

Table 2. Properties of generated problems used in the experimental evaluation (average values over 100 instances per configuration are reported).

Problem	Properties	n_p			
		2	4	8	16
V-DPAP	Conflicts	37,715.34	74,009.12	146,657.94	291,831.52
	Conflict size	2.0	2.0	2.0	2.0
	Slots per RT	1.94	3.81	7.54	15.01
	Slot duration (s)	618.10	616.44	616.91	616.66
R-DPAP	Conflicts	1715.38	3527.42	6981.19	13,929.55
	Conflict size	3.28	3.17	3.21	3.19
	Slots per RT	1.94	3.81	7.54	15.01
	Slot duration (s)	618.10	616.44	616.91	616.66

For each pair (problem type, request satisfaction mode) in $\{V\text{-DPAP}, R\text{-DPAP}\} \times \{\text{full}, \text{partial}\}$, we have generated four types of plots. The first and second types of plots (e.g., Figure 11a,b) allow visualization of the average normalized global utility and the average global reward (i.e., utility not normalized), respectively, both with $[0.05, 0.95]$ as a confidence interval ² for each constellation size and for each algorithm. In the second type of plot (e.g., Figure 11c), the average computation time (logarithmic time scale) is indicated, also for each constellation size and each algorithm. Finally, the fourth type of plot (e.g., Figures 12) allows us to analyze the fairness of the resulting allocations. More precisely, we show the average utility profile in all instances for each algorithm and for each constellation size. Such a utility profile is in leximin order: for each radar, among the four agents, the south represents the agent having the best utility over all agents, the west is the second best utility, the north is the third best utility, and the east corresponds to the agent with the worst utility. For some cases, we sometimes detail utility profiles obtained by each algorithm in some specific instances.

We first present results associated with the full request satisfaction mode, and then results associated with the partial request satisfaction mode.

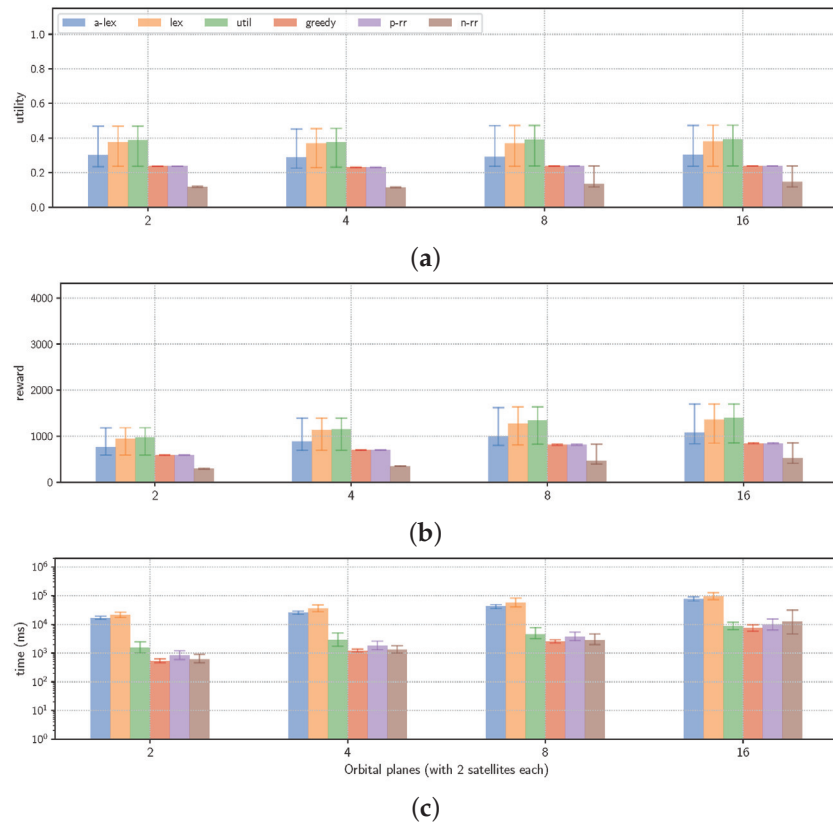


Figure 11. Performance metrics obtained by each algorithm for each constellation size, for full request satisfaction mode and encoded as V-DPAP. (a) Normalized utility; (b) global reward; (c) computation time.

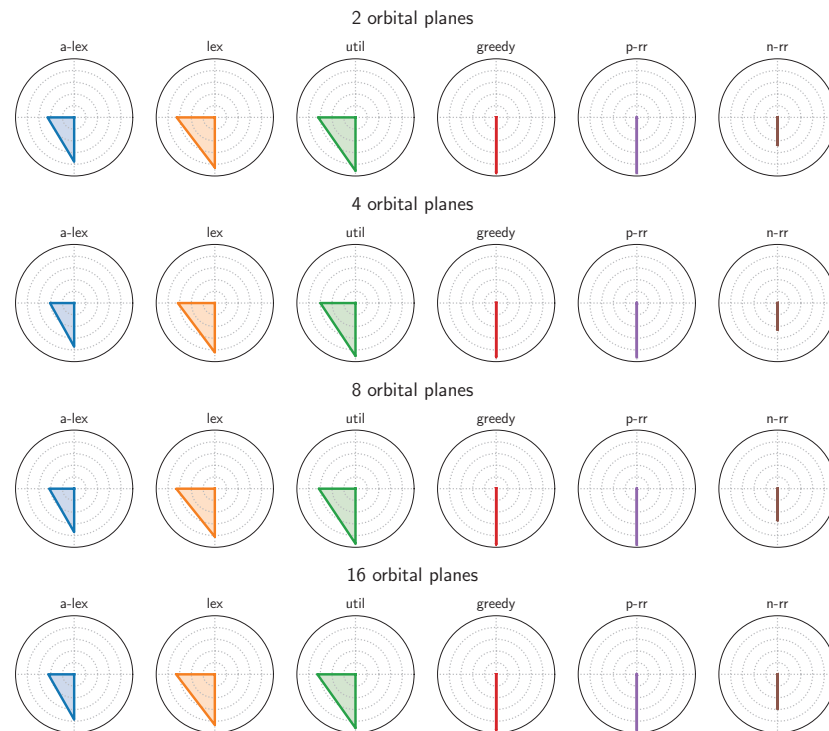


Figure 12. Average utility profiles (in leximin order) for each constellation size and each algorithm (south: best utility over all agents; west: second best utility; north: third best utility; east: worst utility), for full request satisfaction mode and encoded as V-DPAP.

7.2. Results for the Full Request Satisfaction Mode

7.2.1. V-DPAP Results Analysis for $n_p = 2$

Figure 11a compares the normalized utility obtained by each algorithm. As expected, the utilitarian allocation algorithm (util) returns the best global utility. Such a utility is nevertheless quite low, as it does not reach 0.4 on average. The leximin allocation algorithm (lex) is the second in terms of the normalized utility. Its normalized utility score is slightly lower than that of the util. The approximate leximin (a-lex) algorithm's utility is around 0.3. The fact that a-lex's utility is lower than lex's comes from the fact that a-lex cannot backtrack on its decision on the agent's order in the leximin vector, which is prejudicial in the case of the utility's equality within agents. Greedy allocation (greedy) and path round-robin allocation (p-rr) have almost the same global utility (around 0.2) and finally, node round-robin allocation (n-rr) has a global utility lower than 0.1. In terms of global reward, this corresponds to 1000 for the best algorithm (util) and around 150 for the worst one (n-rr).

The time required by each algorithm is reported in Figure 11c. The most time-consuming approaches are a-lex and lex (around 20 s). In fact, they have to call the MILP solver (CPLEX) as many times as the number of agents (here four). Algorithms greedy, n-rr, and p-rr are the fastest ones as they return a solution in less than a second. Algorithm util returns a solution in approximately 10 s.

The top line of Figure 12 displays the average utility profiles involving two orbital planes. The best served agent has a utility very close to 1. Such radars show that the worst served and second worst served agents all have a null utility. This comes from the fact that the corresponding instances are very conflicting. Once two agents receive a path with a utility strictly greater than 0, this prevents the others from satisfying their requests. We can notice that algorithms util and lex have very similar utility profiles on average. Algorithm a-lex does not perform as well for fairness, specifically for the second best served agent. Algorithms n-rr, p-rr, and greedy serve only one agent.

7.2.2. Sensitivity to Constellation Size

The comparison between the algorithms for utility, computation time, and leximin profiles does not change with respect to the number of orbital planes. In other words, the relative performance of the algorithms is the same whatever the size of the constellation. Figure 11a shows that the normalized global utility obtained by the agents does not increase a lot with the constellation's size. However, the allocation's global reward increases with the growing number of orbital planes. In fact, as shown in Figure 11b, the global reward obtained for 2 orbital planes (i.e., a constellation with 4 satellites) is around 1000 for the util algorithm. When considering 16 orbital planes (i.e., a constellation with 32 satellites), such a reward almost reaches 1500, at best. The fact that the normalized utility does not increase with the constellation size but the global reward does, comes from the normalization factor. In fact, with 32 satellites, the global utility that the agents can obtain individually is higher than with 4 satellites. However, the global utility obtained by the agents is relatively the same compared with their best paths and results in a similar normalized global utility.

The time required for computing the allocations also increases with the size of the constellation. More precisely, Figure 11c shows that computation time is multiplied by 10 when the number of satellites increases from 4 to 32. This comes from the higher number of orbit slots and consequently much larger graphs (see Table 2) with more paths to explore and more constraints to check.

The average utility profiles given in Figure 12 show that the utility profiles obtained by algorithms do not change much with the constellation size. Even with more satellites, at the most two agents have a utility strictly greater than 0. This confirms the high number of conflicts of the requests in the considered instances. This illustrates the low utility and reward obtained in this setting: few requests are fulfilled in the end.

These results show that, in the case of V-DPAP with full request satisfaction mode, algorithm util is quite interesting in terms of global utility versus required time. Moreover,

as the instances do not allow the utility profiles to be balanced between the agents, this algorithm provides as fair allocations as algorithm lex.

7.2.3. R-DPAP Results

Figures 13–15 present the results associated with R-DPAP instances in the full request mode satisfaction. The algorithms behave relatively to each other as for the V-DPAP case. More precisely, with respect to the global utility, algorithm util returns the best global utility, then, lex, a-lex, greedy, and n-rr equivalently, and finally, p-rr.

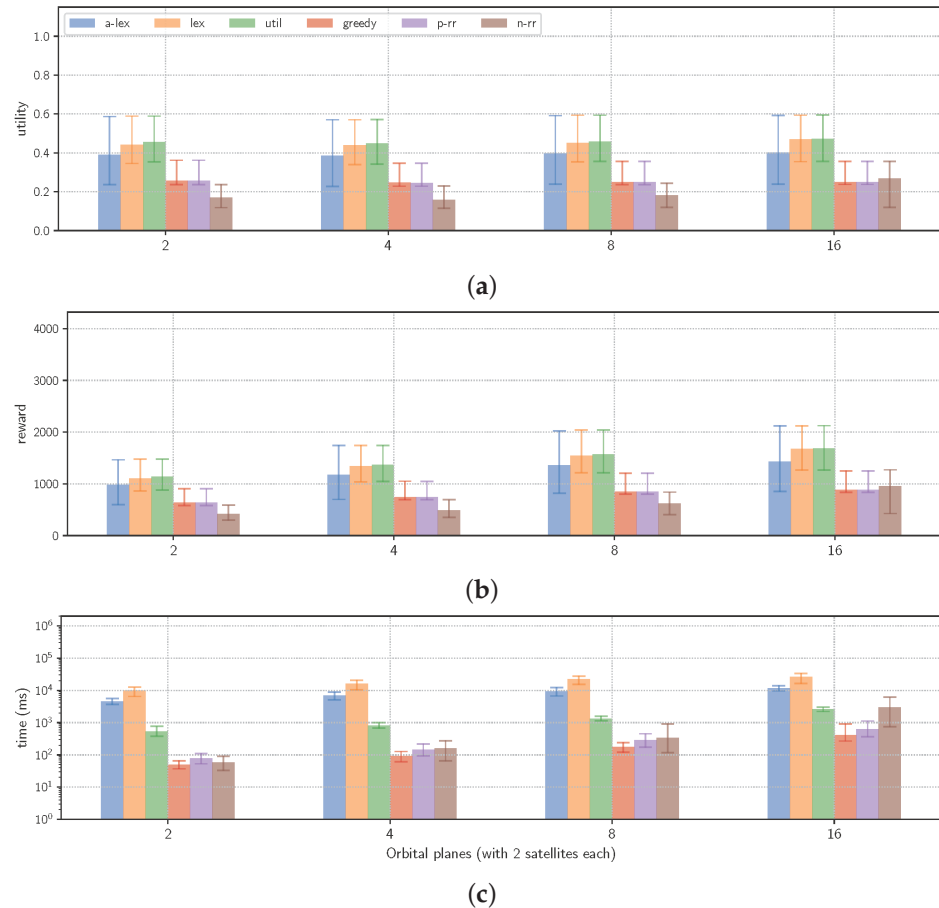


Figure 13. Performance metrics obtained by each algorithm for each constellation size, for full request satisfaction mode and encoded as R-DPAP. (a) Normalized utility; (b) global reward; (c) computation time.

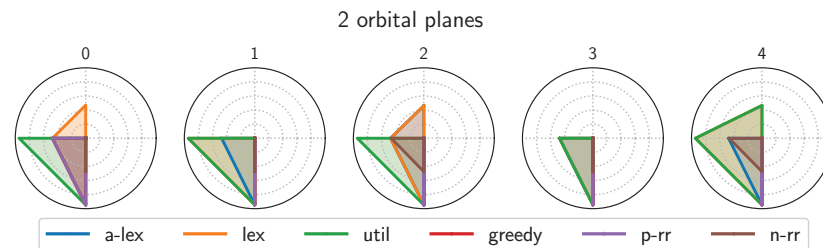


Figure 14. Utility profiles (in leximin order) for the first 5 instances for a constellation with 2 orbital planes (4 satellites) and each algorithm (south: best utility over all agents; west: second best utility; north: third best utility; east: worst utility), for full request satisfaction mode and encoded as R-DPAP.

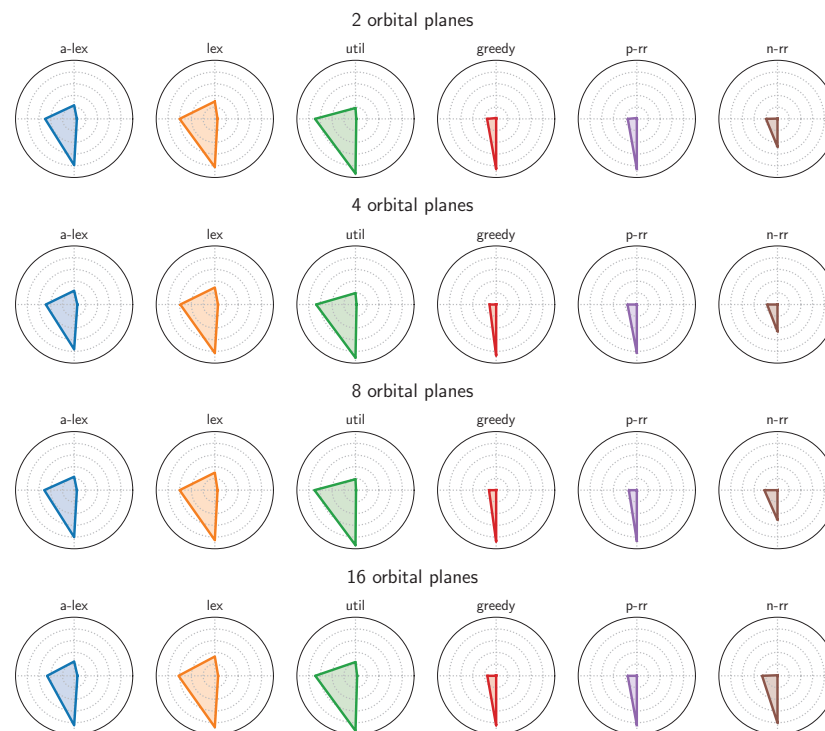


Figure 15. Average utility profiles (in leximin order) for each constellation size and each algorithm (south: best utility over all agents; west: second best utility; north: third best utility; east: worst utility), for full request satisfaction mode and encoded as R-DPAP.

In comparison with the V-DPAP results, instances encoded in the R-DPAP framework allow a higher utility to be reached. First, the normalized global utility (Figure 13a) is around 0.5 for algorithm util with 32 satellites. Note that it is only a little less with 4 satellites (around 0.45). In terms of global reward (Figure 13b), the average score is around 1700 for util with 32 satellites.

The fact that the utility is higher in the R-DPAP benchmark than in the V-DPAP framework comes from the fact that the first allows the orbit slots to be split while the second does not. Consequently, with R-DPAP, when an agent is given an orbit slot on a path with a non-null utility, overlapping orbit slots for other agents might still be selectable. Such a phenomenon can be confirmed by the radars of Figure 14. Indeed, instance 0 with 2 orbital planes (left radar) shows that it is possible for three agents out of four to have a non-null utility. Such an allocation is obtained with algorithm lex. In this case, note that algorithm a-lex performs worse than lex in the sense that two agents have a zero utility with a-lex. This is probably due to the fact that both algorithms compute that the worst served agent has a null utility, but a-lex has to choose to which agent this null utility is allocated. In the case of a bad choice, this prevents a-lex from obtaining a higher utility for the second worst served agent.

The left radar of Figure 14 also shows that util tends to favor agents with high utilities (two agents with utility equal to 1, and two agents with 0), whereas lex splits utility between agents (best agent with utility 1, two others with utility 0.45, and the last with 0). The average utility profiles of Figure 15 confirm this difference of behavior between algorithms util and lex. As for V-DPAP, algorithm a-lex's performance is lower than util and lex with respect to fairness. Other approaches manage on average to serve a second agent but with a very low utility.

Finally, the order of magnitude for the time required to compute solutions is the same between V-DPAP and R-DPAP.

These results show that, in the case of R-DPAP with full request mode satisfaction, the best trade-off between global utility and computation time is given by algorithm util.

However, in terms of fairness, this algorithm is not as good as algorithm lex in several instances, even if lex gives larger computation times.

Note that R-DPAP is still parametric in the sense that it requires defining the duration $minD$ (here 120 s) requested in each orbit slot. With a low $minD$ value, orbit sharing can be possible, while using a high $minD$ value may prevent such splitting, and in the extreme case R-DPAP becomes equivalent to V-DPAP, utility-wise.

7.3. Results for the Partial Request Satisfaction Mode

We now analyze the results for the instances in which requests can be partially satisfied by skipping some RTs.

7.3.1. V-DPAP Results

Figures 16 and 17 show the results for instances encoded as V-DPAP. From Figure 16a, we can observe that the normalized utility is much higher than with instances encoded in V-DPAP with the request full satisfaction mode. For instance, for a constellation involving 4 satellites, algorithms util, lex and a-lex almost reach a 0.6 normalized utility value. For 32 satellites, this normalized utility is equal to 0.85. In terms of reward (Figure 16b), the global reward is also much higher. Note that the relative performance of the algorithms is the same as for V-DPAP with the full satisfaction mode, i.e., algorithm util returns the allocation with the best global utility, then lex, a-lex, p-rr, greedy, and n-rr. Nevertheless, with 32 satellites, all algorithms except n-rr return allocations with approximately the same global utility. This increase in performance with the change in request mode satisfaction shows that even if paths conflict, the skip possibility allows many more requests to be tackled.

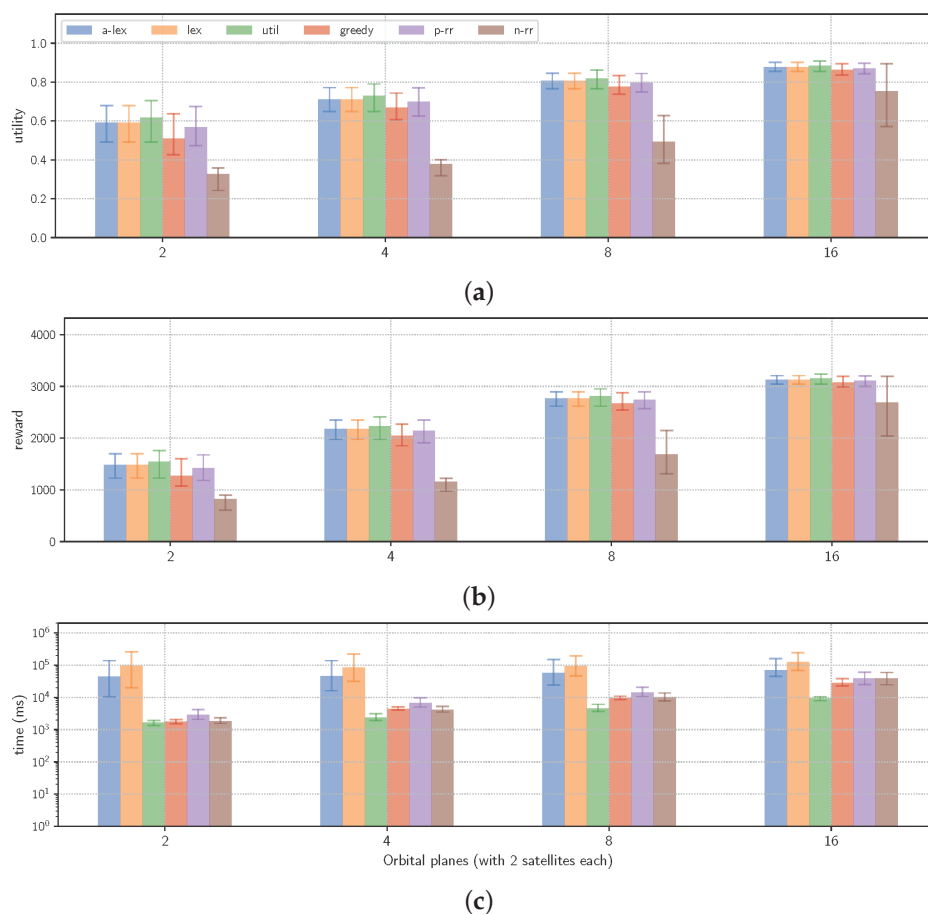


Figure 16. Performance metrics obtained by each algorithm for each constellation size, for flexible requests encoded as V-DPAP. (a) Normalized utility; (b) global reward; (c) computation time.

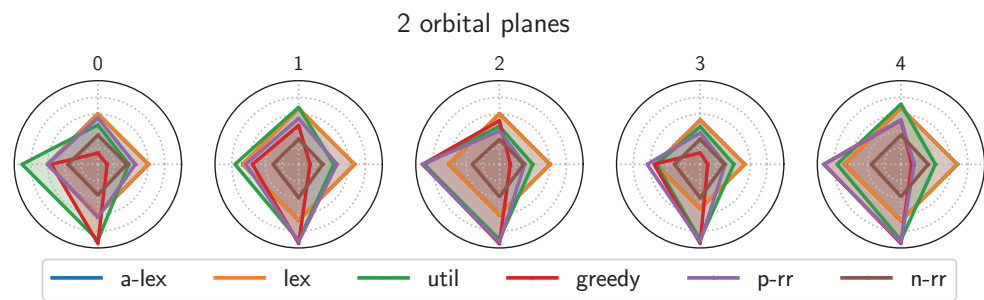


Figure 17. Utility profiles (in leximin order) for the first 5 instances for a constellation with 2 orbital planes (4 satellites) and each algorithm (south: best utility over all agents; west: second best utility; north: third best utility; east: worst utility), for flexible requests encoded as V-DPAP.

From Figure 16c, we can first notice that the time required by algorithms lex and a-lex is much higher than for V-DPAP with request full satisfaction mode. In fact, for algorithm lex, 10 s are required for V-DPAP with request full satisfaction mode but 100 for V-DPAP with request partial satisfaction mode. However, for these algorithms, the order of magnitude does not change with the constellation size. Such a phenomenon is probably due to the fact that there are a lot of complex paths (i.e., paths that are not *source* → *sink*) with the same utility, which makes it harder to compute the worst utility for a given agent. Algorithms greedy, p-rr, and n-rr also require much more time than for instances in V-DPAP with full request satisfaction mode. This can be explained by the fact the number of paths is much larger but that nodes still belong to several conflicts. Therefore, every time a path is selected in a graph, other graphs have many nodes that are deactivated, which forces new best paths to be computed and overall requires some computation time. In comparison, algorithm util requires approximately the same time in the partial and full satisfaction modes.

Next, Figures 17 and 18 show that in the partial satisfaction mode, the utility profiles are much more balanced between agents. The radars in Figure 17 allow the algorithms' behaviors to be compared over some instances involving two satellites. It shows that algorithm util favors high utilities, which is sometimes quite fair (instance 3) and sometimes not (instance 0). Algorithm greedy serves very well one agent but cannot serve well the others because of conflicts between paths. Algorithm p-rr performs a little better than greedy in terms of fairness. Algorithms lex and a-lex allow the utility to be balanced between the agents. For instance, the top line radars show that it is possible to reach a solution where all agents have approximately the same utility (around 0.6). Algorithm n-rr is also quite fair but the utility per agent is much lower (0.2). Figure 18 shows that these comments can be generalized to all instances on average.

In the case of a larger constellation, the algorithms (except n-rr) behave almost the same in terms of leximin vectors, and there exist solutions where all agents can be served quite well.

These results show that in the case of V-DPAP with requests partial satisfaction mode, algorithm util offers the best utility/time trade-off. However, in terms of fairness, such an algorithm gives good performance only for constellations with at least 8 orbital planes (16 satellites). For a smaller number of satellites, algorithms lex and a-lex can be much fairer (depending on the instance), despite a greater computation time.

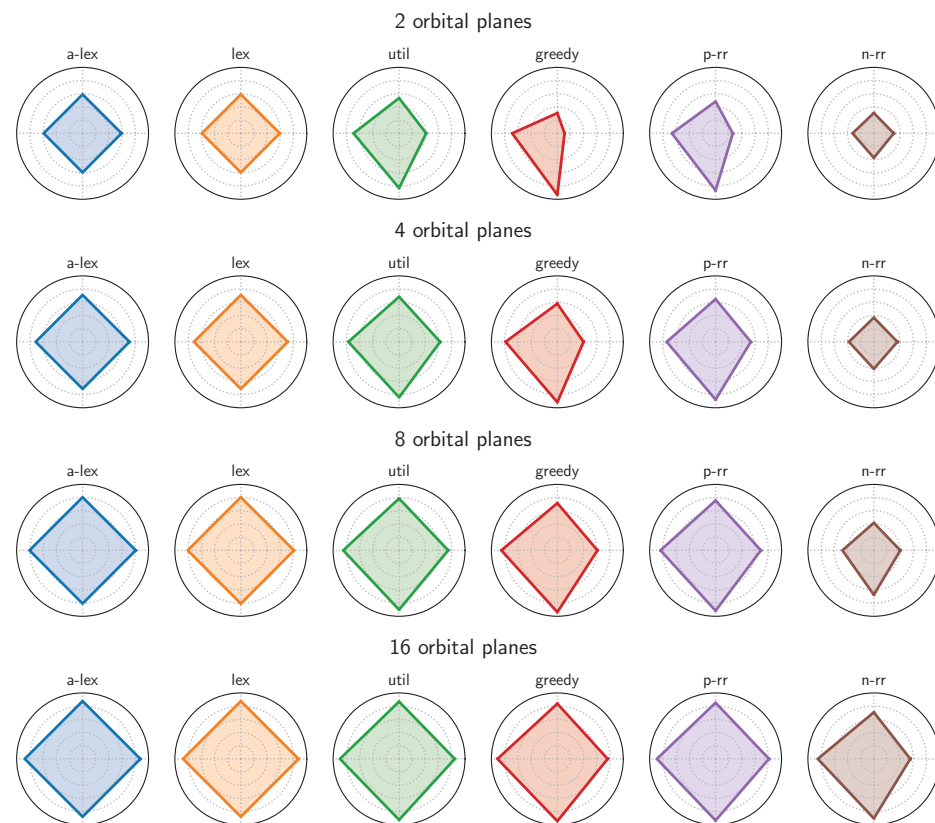


Figure 18. Average utility profiles (in leximin order) for each constellation size and each algorithm (south: best utility over all agents; west: second best utility; north: third best utility; east: worst utility), for partial request satisfaction mode and encoded as V-DPAP.

7.3.2. R-DPAP Results

In the case of R-DPAP with partial request satisfaction mode, Figure 19a shows that the maximum utility is reached by all algorithms whatever the constellation size, except for n-rr. Note that the obtained global normalized utility is not equal to 1 because there are still some conflicts between some orbit slots that prevent the agents from obtaining their best paths.

Figure 19b shows that the global reward increases with the number of satellites in the constellation. In fact, the larger the constellations, the higher the number of orbit slots and the higher the number of paths with a higher utility in the graphs.

For all of the algorithms, the computation time required is much lower than for V-DPAP with partial satisfaction mode. This is quite natural, since even if there is a large number of paths, the selection of one path for an agent does not require deactivating many nodes in other graphs. This comes from the fact that orbit slots can be split between agents, which results in less conflicts between nodes.

We do not provide here radars per instance, since the profiles obtained by the algorithms all overlap. Indeed, Figure 20 confirms that all the agents have a utility almost equal to 1 for all the algorithms except n-rr. The latter struggles with highly conflicting settings. With less conflicting settings (with more satellites) n-rr drastically improves its performance, since there is less chance to reach a situation where an agent must skip one RT.

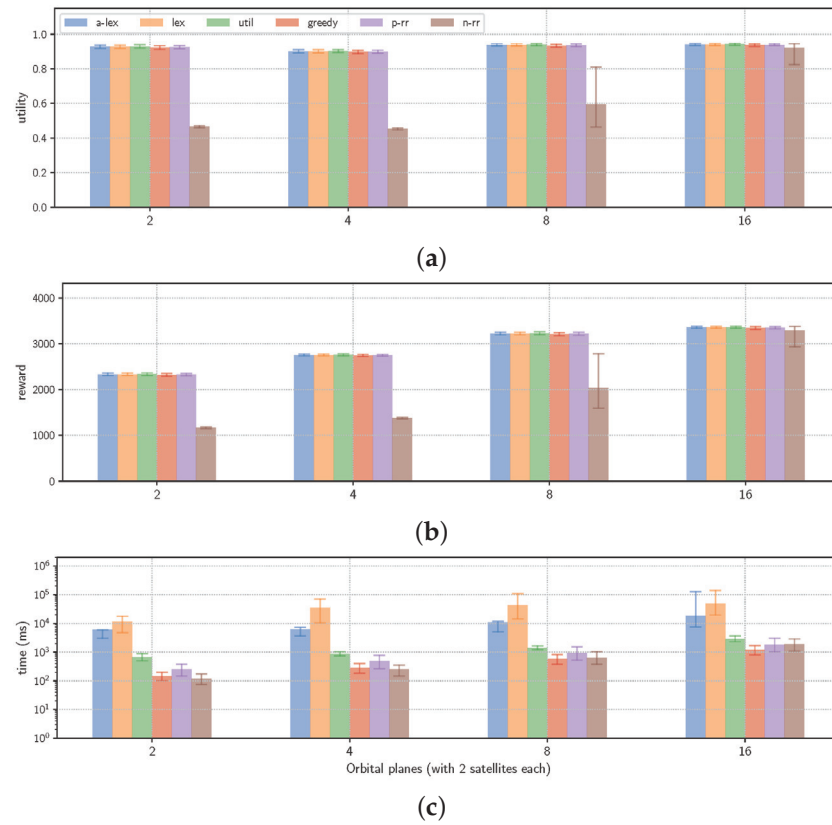


Figure 19. Performance metrics obtained by each algorithm for each constellation size, for flexible requests encoded as R-DPAP. (a) Normalized utility; (b) global reward; (c) computation time.

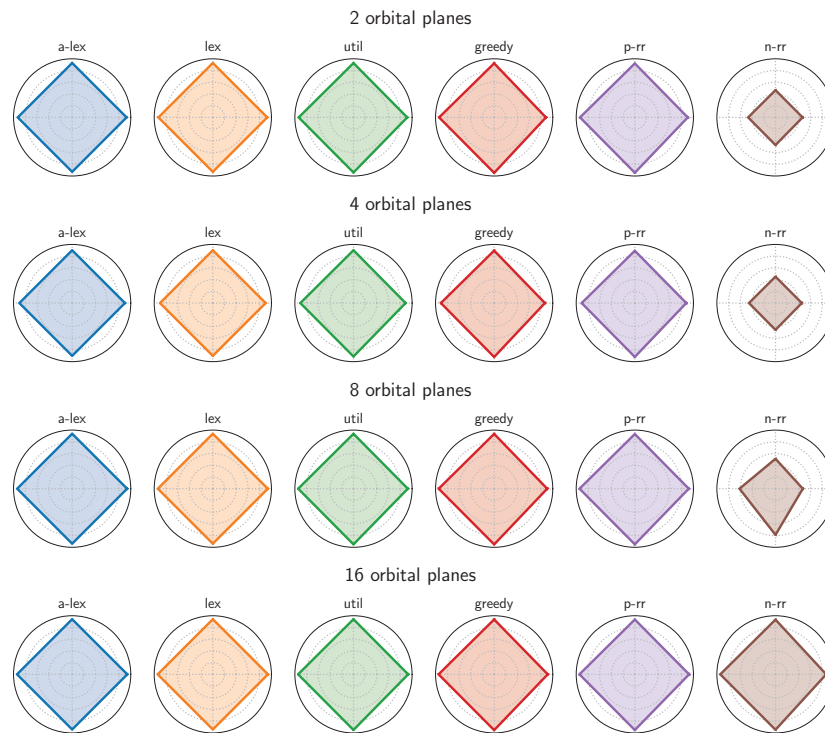


Figure 20. Average utility profiles (in leximin order) for each constellation size and each algorithm (south: best utility over all agents; west: second best utility; north: third best utility; east: worst utility), for partial request satisfaction mode and encoded as R-DPAP.

These results suggest that in the case of R-DPAP with full request satisfaction mode, algorithm greedy offers the best trade-off in quality/time as it allows a fair allocation to be reached with a high global utility (as for other algorithms) but in much less time.

8. Conclusions

In this paper, we proposed several models for novel resource allocation problems where agents express their preferences over conflicting bundles of items as edge-weighted DAGs (DPAP). We particularly focused on conflicts on vertices (V-DPAP) and conflicts on resources (R-DPAP). We introduced and analyzed several solution methods (utilitarian, leximin, approximate leximin, greedy) against the classically used round-robin allocations from the utilitarianism and fairness perspectives. We evaluated these methods on large randomly generated instances of orbit slot allocation problems, where requests could be fully or partially fulfilled. We showed that when requests must be fully fulfilled, allowing resource sharing via R-DPAP encoding improves the performance of the system compared to V-DPAP with respect to normalized utility and global reward, while the computation times are equivalent or lower. When considering the request full satisfaction mode, problems encoded as V-DPAP are much more constrained with respect to the number of agents that can receive a non-empty allocation. Therefore, algorithm util is a relevant approach. In the case of R-DPAP, algorithm a-lex provides good results with respect to utility and is much fairer than other approaches, even if it requires a longer computation time. In the case of partial request satisfaction mode and V-DPAP problems, there is no clear winner on all metrics for small constellations: lex clearly returns fair allocations with a good global utility but requires a long computation time. On the other hand, algorithm util is faster but not as fair. For large constellations, algorithm util allows us to reach the fairest allocations and is, therefore, the most suitable. Finally, when offering even more flexibility, i.e., allowing partial request fulfilling, the performances become even better, to a point where, for larger constellations, all the algorithms reach the same optimal and fair allocations. This highlights that adding request flexibility eases the allocation process, whilst the problems remain NP-hard in general. In such a case, non-exact algorithms such as greedy offer the best trade-off with respect to utility, fairness, and computation time.

We identify several tracks for future investigations. First, as DPAPs are strongly constrained by conflicts, we aim to explore minimum conflict heuristics to improve our algorithms. Secondly, we believe DPAP and its variants have great potential to be used in a variety of domains, and we thus aim to evaluate the proposed techniques on problems coming from other application fields, such as the NFV domain (function chains modeled as graphs and incompatibilities controlling the access to nodes) or the multi-agent path finding domain (path preferences modeled as graphs and incompatibilities, imposing that two agents cannot occupy the same position at the same time). Depending on the targeted application, other ways for expressing conflicting bundles could be explored. For instance, one could consider that items can consume resources with capacity. Finally, in the Earth observation domain, once the slots have been allocated, the agents have to plan their own observations within the allocated slots, and may have to interact to accept external observations. Such a coordination scheme has been investigated [3], but we aim to evaluate the whole chain (slot allocation followed by coordinated observation scheduling) on realistic data.

Author Contributions: Conceptualization, S.R., G.P., C.P. and S.M.; data curation, S.R., G.P., C.P. and S.M.; formal analysis, S.R., G.P. and C.P.; funding acquisition, S.R. and C.P.; investigation, S.R., G.P., C.P. and S.M.; methodology, S.R., G.P., C.P. and S.M.; project administration, S.R., G.P. and C.P.; resources, S.R., G.P., C.P. and S.M.; software, S.R., G.P., C.P. and S.M.; supervision, S.R., G.P., C.P. and S.M.; validation, S.R., G.P., C.P. and S.M.; visualization, S.R., G.P. and C.P.; writing—original draft, S.R., G.P. and C.P.; writing—review and editing, S.R., G.P., C.P. and S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been performed with the support of the French government in the context of the “Programme d’Investissements d’Avenir”, namely, by the BPI PSPC LiChIE project (Lion Chaîne Image Elargie), coordinated by Airbus Defence and Space.

Data Availability Statement: Instances used in the experimental evaluation are available for the community, and accessible on Zenodo (<https://doi.org/10.5281/zenodo.7669379>).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DAG	Directed acyclic graph
DPAP	Directed path allocation problem
V-DPAP	Vertex-constrained DPAP
R-DPAP	Resource-constrained DPAP
PADAG	Path allocation in directed acyclic graph
ILP	Integer linear programming
MILP	Mixed integer linear programming
POI	Point of interest
RT	Request time
lex	Leximin solver
a-lex	Approximate leximin solver
greedy	Greedy solver
util	Utilitarian MILP solver
p-rr	Path round-robin solver
n-rr	Node round-robin solver

Notes

- ¹ A Nash equilibrium is an allocation in which the modification of a path for a single agent does not improve its associated utility.
- ² We have removed the worst 5% of values and the best 5% of values for the indicated range.

References

- Yang, S.; Li, F.; Trajanovski, S.; Yahyapour, R.; Fu, X. Recent Advances of Resource Allocation in Network Function Virtualization. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 295–314. [[CrossRef](#)]
- Lemaître, M.; Verfaillie, G.; Fargier, H.; Lang, J.; Bataille, N.; Lachiver, J.M. Equitable Allocation of Earth Observing Satellites Resources. In Proceedings of the 5th ONERA-DLR Aerospace Symposium (ODAS’03), Toulouse, France, 4–6 June 2003.
- Picard, G. Auction-based and Distributed Optimization Approaches for Scheduling Observations in Satellite Constellations with Exclusive Orbit Portions. In Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS-22), Virtual Event, New Zealand, 9–13 May 2022.
- Maqrot, S.; Roussel, S.; Picard, G.; Pralet, C. Bundle Allocation with Conflicting Preferences Represented as Weighted Directed Acyclic Graphs—Application to Orbit Slot Ownership. In *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation. The PAAMS Collection, Proceedings of the 20th International Conference, PAAMS 2022, L’Aquila, Italy, 13–15 July 2022*; Dignum, F., Mathieu, P., Corchado, J.M., De la Prieta, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2022; Volume 13616, pp. 280–293. [[CrossRef](#)]
- Bouveret, S.; Cechlárová, K.; Elkind, E.; Igarashi, A.; Peters, D. Fair Division of a Graph. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, 19–25 August 2017; pp. 135–141. [[CrossRef](#)]
- Igarashi, A.; Peters, D. Pareto-Optimal Allocation of Indivisible Goods with Connectivity Constraints. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, HI, USA, 27 January–1 February 2019; pp. 2045–2052. [[CrossRef](#)]
- Immorlica, N.; Karger, D.R.; Nikolova, E.; Sami, R. First-Price Path Auctions. In Proceedings of the Proceedings 6th ACM Conference on Electronic Commerce (EC-2005), Vancouver, BC, Canada, 5–8 June 2005; pp. 203–212. [[CrossRef](#)]
- Du, Y.; Sami, R.; Shi, Y. Path Auctions with Multiple Edge Ownership. *Theor. Comput. Sci.* **2010**, *411*, 293–300. [[CrossRef](#)]
- Zhang, L.; Chen, H.; Wu, J.; Wang, C.; Xie, J. False-Name-Proof Mechanisms for Path Auctions in Social Networks. In Proceedings of the Twenty-Second European Conference on Artificial Intelligence, The Hague, The Netherlands, 29 August–2 September 2016; Volume 285, pp. 1485–1492. [[CrossRef](#)]
- Megiddo, N. Optimal Flows in Networks with Multiple Sources and Sinks. *Math. Program.* **1974**, *7*, 97–107. [[CrossRef](#)]

11. Hughes, M.S.; Lunday, B.J.; Weir, J.D.; Hopkinson, K.M. The Multiple Shortest Path Problem with Path Deconffliction. *Eur. J. Oper. Res.* **2021**, *292*, 818–829. [[CrossRef](#)]
12. Nisan, N.; Roughgarden, T.; Tardos, E.; Vazirani, V.V. *Algorithmic Game Theory*; Cambridge University Press: Cambridge, NY, USA, 2007.
13. Panagopoulou, P.N.; Spirakis, P.G. Algorithms for Pure Nash Equilibria in Weighted Congestion Games. *ACM J. Exp. Algorithmics* **2007**, *11*, 2.7-es. [[CrossRef](#)]
14. Caragiannis, I.; Kurokawa, D.; Moulin, H.; Procaccia, A.D.; Shah, N.; Wang, J. The Unreasonable Fairness of Maximum Nash Welfare. *ACM Trans. Econ. Comput.* **2019**, *7*, 1–32. [[CrossRef](#)]
15. Riera, J.F.; Escalona, E.; Batallé, J.; Grasa, E.; García-Espín, J.A. Virtual Network Function Scheduling: Concept and Challenges. In Proceedings of the 2014 International Conference on Smart Communications in Network Technologies (SaCoNeT), Vilanova i la Geltru, Spain, 18–20 June 2014; pp. 1–5. [[CrossRef](#)]
16. Maqrot, S.; Roussel, S.; Picard, G.; Pralet, C. Orbit Slot Allocation in Earth Observation Constellations. In *PAIS 2022, Proceedings of the 11th Conference on Prestigious Applications of Artificial Intelligence, Vienna, Austria, 25 July 2022*; Passerini, A., Schiex, T., Eds.; co-located with IJCAI-ECAI 2022; Frontiers in Artificial Intelligence and Applications; IOS Press: Amsterdam, The Netherlands 2022; Volume 351, pp. 3–16. [[CrossRef](#)]
17. Wang, X.; Wu, G.; Xing, L.; Pedrycz, W. Agile Earth Observation Satellite Scheduling over 20 years: Formulations, Methods and Future Directions. *IEEE Syst. J.* **2021**, *15*, 3881–3892. [[CrossRef](#)]
18. Tangpattanakul, P.; Jozefowicz, N.; Lopez, P. A Multi-Objective Local Search Heuristic for Scheduling Earth Observations Taken by an Agile Satellite. *Eur. J. Oper. Res.* **2015**, *245*, 542–554. [[CrossRef](#)]
19. Bianchessi, N.; Cordeau, J.F.; Desrosiers, J.; Laporte, G.; Raymond, V. A Heuristic for the Multi-Satellite, Multi-Orbit and Multi-User Management of Earth Observation Satellites. *Eur. J. Oper. Res.* **2007**, *177*, 750–762. [[CrossRef](#)]
20. Eddy, D.; Kochenderfer, M.J. A Maximum Independent Set Method for Scheduling Earth-Observing Satellite Constellations. *J. Spacecr. Rocket.* **2021**, *58*, 1416–1429. [[CrossRef](#)]
21. Lenstra, J.; Rinnooy Kan, A.; Brucker, P. Complexity of Machine Scheduling Problems. In *Studies in Integer Programming*; Hammer, P., Johnson, E., Korte, B., Nemhauser, G., Eds.; Elsevier: Amsterdam, The Netherlands, 1977; Volume 1, pp. 343–362. . [[CrossRef](#)]
22. Moulin, H. *Fair Division and Collective Welfare*; MIT Press: Cambridge, MA, USA, 2003.
23. Kurokawa, D.; Procaccia, A.D.; Shah, N. Leximin Allocations in the Real World. *ACM Trans. Econ. Comput.* **2018**, *6*, 1–24. [[CrossRef](#)]
24. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2001.
25. OpenStreetMap Points of Interest (on French Territory). 2021. Available online: <https://www.data.gouv.fr/fr/datasets/points-dinterets-openstreetmap/> (accessed on 30 August 2021).
26. Wang, J.; Song, G.; Liang, Z.; Demeulemeester, E.; Hu, X.; Liu, J. Unrelated Parallel Machine Scheduling with Multiple Time Windows: An Application to Earth Observation Satellite Scheduling. *Comput. Oper. Res.* **2023**, *149*, 106010. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland
www.mdpi.com

Systems Editorial Office
E-mail: systems@mdpi.com
www.mdpi.com/journal/systems



Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Academic Open
Access Publishing

mdpi.com

ISBN 978-3-0365-9308-1