

Article

Edge Computing and IoT Analytics for Agile Optimization in Intelligent Transportation Systems

Mohammad Peyman ^{1,*}, Pedro J. Copado ^{1,2}, Rafael D. Tordecilla ^{1,3}, Leandro do C. Martins ¹,
Fatos Xhafa ^{4,*} and Angel A. Juan ^{1,2}

¹ IN3—Computer Science Department, Universitat Oberta de Catalunya, 08018 Barcelona, Spain; pcopadom@uoc.edu (P.J.C.); rtordecilla@uoc.edu (R.D.T.); leandrocm@uoc.edu (L.d.C.M.);ajuanp@uoc.edu (A.A.J.)

² Department of Data Analytics&Business Intelligence, Euncet Business School, 08018 Barcelona, Spain

³ School of Engineering, Universidad de La Sabana, Chia 250001, Colombia

⁴ Computer Science Department, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain

* Correspondence: mpeyman@uoc.edu (M.P.); fatos@cs.upc.edu (F.X.)

Abstract: With the emergence of fog and edge computing, new possibilities arise regarding the data-driven management of citizens' mobility in smart cities. Internet of Things (IoT) analytics refers to the use of these technologies, data, and analytical models to describe the current status of the city traffic, to predict its evolution over the coming hours, and to make decisions that increase the efficiency of the transportation system. It involves many challenges such as how to deal and manage real and huge amounts of data, and improving security, privacy, scalability, reliability, and quality of services in the cloud and vehicular network. In this paper, we review the state of the art of IoT in intelligent transportation systems (ITS), identify challenges posed by cloud, fog, and edge computing in ITS, and develop a methodology based on agile optimization algorithms for solving a dynamic ride-sharing problem (DRSP) in the context of edge/fog computing. These algorithms allow us to process, in real time, the data gathered from IoT systems in order to optimize automatic decisions in the city transportation system, including: optimizing the vehicle routing, recommending customized transportation modes to the citizens, generating efficient ride-sharing and car-sharing strategies, create optimal charging station for electric vehicles and different services within urban and interurban areas. A numerical example considering a DRSP is provided, in which the potential of employing edge/fog computing, open data, and agile algorithms is illustrated.



Citation: Peyman, M.; Copado, P.J.; Tordecilla, R.D.; Martins, L.d.C.; Xhafa, F.; Juan, A.A. Edge Computing and IoT Analytics for Agile Optimization in Intelligent Transportation Systems. *Energies* **2021**, *14*, 6309. <https://doi.org/10.3390/en14196309>

Academic Editor: Sergio Saponara

Received: 2 August 2021

Accepted: 29 September 2021

Published: 2 October 2021

Keywords: fog; edge computing; Internet of Things; intelligent transportation systems; smart cities; machine learning; agile optimization

1. Introduction

In today's modern society, urban centers are facing the so-called booming of information. Due to the population growth in many countries around the globe, and recent innovations in information and telecommunication technologies, several activities and related challenges have jointly arisen. People are increasingly consuming more information through their mobile devices, vehicles are equipped with different intelligent systems, devices are distributed around the cities for gathering and generating information, and urban areas are continuously taking advantage of these information technologies and big data. Consequently, so-called smart cities have emerged, whose scope combines sustainable development with the intelligent management of gathered data in order to enhance the operation of different services within urban areas, such as waste collection management [1], car-sharing/ride-sharing activities [2], the optimal location of recharging stations for electric vehicles (EVs), among others. In this matter, during the past few years, the Internet of things (IoT) has become a popular term that plays a significant role to expand and produce a lot of data through sensors and allows citizens and things to be connected in any situation



Copyright: © 2021 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/>).

or with anyone [3]. Moreover, fog and cloud computing come to support IoT to manage the large amount of generated data [4]. In Figure 1, Roadside Units (RSU), cellphones, and vehicles share the fog layer (in green), whose devices are connected through the edge layer (connections) to the cloud/open data repository layer (in blue). These connections allow the interchanging of data among these agents. The open data server stores information stemming from IoT devices installed at the edge of the networks. The grabbed data are then processed in site, instead of being sent to the cloud directly by data analytics in the fog layer.

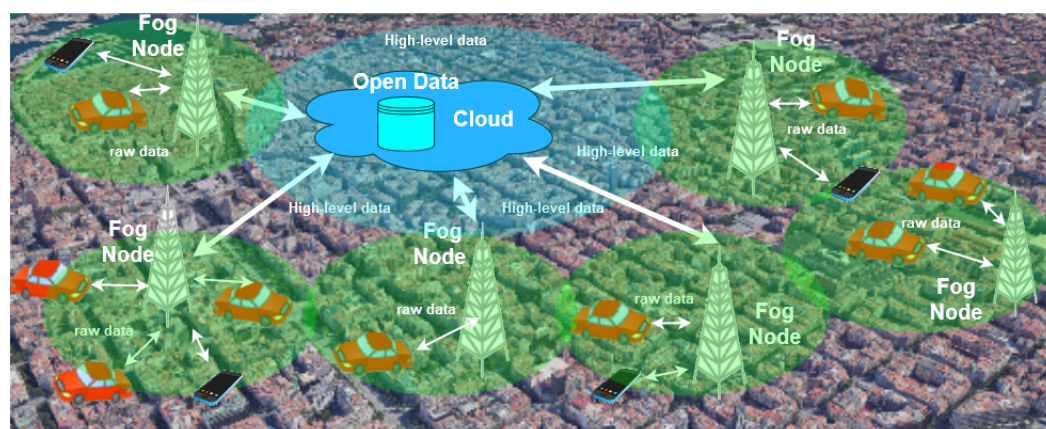


Figure 1. The fog-computing architecture deployed in this work, in which RSUs as are used as fog nodes and the Internet of Vehicles (IoV) as the communication protocol.

One of the main tasks when building smart cities is the development of intelligent transportation systems (ITS). These systems need to establish exact, effective, comprehensive, and real-time control systems, relying on IoT and capable of reducing or solving the phenomenon of mobility overcrowding [5,6]. Furthermore, integrating IoT and open data initiatives in smart cities allows governments, public, and private sectors to develop new services and applications by ensuring the effective handling and managing of data that are constantly shared among individual citizens and different industries [7]. For instance, sensing real-time traffic flow and mobility tracking data, such as vehicle states (e.g., location, speed, etc.), intersection information (e.g., the length of the queue waiting at the intersection, etc.), and the situation of the road (e.g., under construction, traffic accident, etc.) can be analyzed [8,9] to explain the dynamics of urban vehicles as micro and macroscopic simulations, traffic flow, and travel time estimations [10].

In this context, mobile internet technology is one of the actors which enables dynamic and on-demand sharing activities. In ride-sharing systems, for example, people are allowed to offer trips for riders by using their own private vehicles. These ride-sharing economy activities include highly dynamic systems in which drivers and riders receive matches through automated processes that are used by a ride-share provider [11]. Nowadays, the concept of ride-sharing plays an essential role in new transportation paradigms, holding multiple advantages for society in general, e.g., reducing traffic congestion, noise, and pollution; minimizing fares for passengers, and operational costs for drivers, etc. Hence, utilizing the cloud and edge computing helps to handle terabytes of data extracted from IoT devices—including information about vehicles' mobility and traffic conditions. Furthermore, by analyzing these data and combining them with the concept of ride-sharing, some urban mentioned problems can be reduced or even solved. In this context, optimization techniques, such as approximate methods—i.e., heuristics and metaheuristics—have proved to be both efficient and capable of generating high-quality solutions for large-scale and complex real-world problems [12]. This means that heuristics have a high potential to provide agility and real-time responses, which are necessary issues for a good performance of ITS and, in general, of this type of system. Nevertheless, after reviewing some related

work, very few articles combining heuristics with IoT analytics by utilizing the cloud and edge computing have been found.

Hence, to fill this gap, a dynamic ride-sharing problem (DRSP) is addressed in our work, where dynamic conditions usually encountered in modern urban centers affect the decision-making processes. In other words, the DRSP considers dynamic traffic conditions that might lead to several changes on the initially designed routes due to the incorporation of refreshed information, such as traffic conditions and vehicles states. In this problem, a set of routes must be designed so that the total reward collected by picking up passengers is maximized. A discrete-event-driven metaheuristic is proposed to solve this problem. This solution method is enhanced with biased-randomized techniques to provide an efficient exploration of the solution space. Furthermore, this paper has reviewed existing works in the context of IoT, edge/fog, cloud computing in terms of ITS. We highlight multiple challenges, opportunities, and usage of cloud, fog, and edge computing in ITS and discuss how to solve problems such as low latency, handling data, privacy protection, etc. in this area. Additionally, we discuss the role of IoT analytics in ITS and different techniques used for solving the problems. Some of our main goals are:

- To review accessible real open data repositories.
- To review approaches regarding optimization, simulation, machine learning, and agile optimization algorithms in ITS.
- To provide challenges and opportunities of cloud, fog, and edge computing and IoT analytics in ITS.
- To propose a methodology for solving the DRSP in the context of edge/fog computing.

The remaining sections of the paper are structured as follows: Section 2.1 reviews the open data initiatives for smart cities. Section 2.2 describes the optimization, simulation, and machine learning approaches in the ITS. In Section 2.3, the concept of agile optimization for dynamic and intelligent transportation systems is described. Section 3 presents a brief review of related work. We present a case study in Section 4. In Section 5, the discrete-event-based methodology is introduced. Section 6 discusses computational results. Finally, Section 7 summarizes our main insights and provides future research lines.

2. Fundamental Concepts

This section provides a theoretical background about the main concepts employed in this document. Firstly, an exposition about accessible real open data repositories in a group of smart cities is provided. Secondly, quantitative approaches regarding optimization, simulation, machine learning, and agile optimization algorithms in ITS are reviewed.

2.1. Open Data Initiatives for Smart Cities

During the last decade, there have been multiple initiatives involving big cities across the world in order to make ITS data available for the general public. These open-data are accessible to citizens and researchers [13]. The real potential of any small city relies not only on collecting data from sensors, but also on spreading these data to empower citizens, increase transparency, and enhance public services [3,14]. As mentioned by Granickas [15], in the European Union and in the year 2013, the direct benefit associated with the use of open data was approximately of 40 billion euros, which rises to 140 billion euros if we consider the entire set of European countries. There are many smart cities in the world, such as Barcelona (BCN), New York (NYC), Amsterdam, Helsinki, Chicago, Quebec City, Rio de Janeiro (Rio), Dublin, Nairobi, and Manchester. In many of these cities, open-data initiatives are growing fast [16]. Open data is known as one of the most significant and decisive elements of a smart city initiative, and open data allows for increasing access to information, enabling social inclusion and economic development of different smart city frameworks, such as environment, transportation, energy, governance, people and lifestyle, technology, and building infrastructure [17]. Table 1 identified several smart cities from the globe, and two open data sources that provide European Union territories. Moreover, it shows how often the data sets are updated, as well as the main format employed.

Table 1. Open data initiatives in smart cities by the type of format and frequency update.

Open Data Name			Format	Update Frequency
Name	City	Link		
Open Data BCN	Barcelona	https://opendata-ajuntament.barcelona.cat/data/en/organization/transport?page=2	All	availability/need
London data store	London	https://data.london.gov.uk/	All	Daily/ availability/need
Data.gov	United state	https://data.gov/	All	availability
NYC Open Data	New York City	https://data.cityofnewyork.us/	CSV and XLSX and XML & RDF & RSS	availability
Data and information	Amsterdam	https://data.amsterdam.nl/datasets/zoek/	CSV and XLSX and XLS and API	Daily
Overheid	Netherlands	https://data.overheid.nl/	All	Daily
Helsinki Region Info-share	Helsinki	https://hri.fi/en_gb/	All	availability
Chicago Data Portal	Chicago	https://data.cityofchicago.org/	CSV and XLSX and XML and RDF and RSS	availability
Quebec data	Quebec	https://www.donneesquebec.ca/	All	availability/need
Data.Rio	Rio	https://www.data.rio/	CSV and XLSX	availability
Dublinked	Dublin Region	https://data.smartdublin.ie/	All	availability/need
Data.gov	Ireland	https://data.gov.ie/	All	availability
Berlin Open Data	Berlin	https://daten.berlin.de/	CSV and PDF and WFS and HTML	availability
GovData	Germany	https://www.govdata.de/	XLSX and ZIP and PDF and CSV and WMS adn HTML	availability
European Union	several cities in Europe	https://data.europa.eu/euodp/en/data/	All	availability
European data portal	Several cities	https://www.europeandataportal.eu/en	All	availability

Public information has great potential value, which can be relative to different cities domains included in Economic and Business, Environment and Energy, Agriculture, Culture and Tourism, Education and Health. Table 2 considers various sectors that relate to the important area of smart cities, and some keywords are discussed for each. Thus, in the Health sector for instance, it is clear that they are concerned with measuring the impact of the COVID pandemic, the number of cases, or the status of each hospital. In addition, the Environment and Energy data sets cover air quality, wastewater, oil, gas, and energy usage. The data sets for Agriculture cover land use, animals, and a list of associations' equipment. However, some smart cities do not consider that information. Moreover, the Transportation and Traffic data sets are more active in those open data sources. They focus on traffic data, pedestrian facilities trails and paths, road infrastructure, and parking spaces.

Since cities are facing sustainable urban development challenges, innovative services and analytical capabilities are needed for resource optimization and value creation based on data. This information from cities open data is one of the most powerful resources [18]. Table 3 shows the availability of specific vehicles in each open data source. Most of the open data sources consider cars, taxis, buses, and metros in their transportation data sets. However, only a few of them address electric vehicles, unmanned aerial vehicles, airplanes, and ships. Some cities, such as Barcelona, London, New York City, and Dublin provide vehicles' information, while some cities in the United States provide information about unmanned aerial vehicles. Those open data sources create opportunities to improve many operational activities of the city by the anticipated effect on operating costs, public safety, transportation, and quality of municipal services.

Table 2. Sector availability data set in open data initiatives of smart cities.

Name	Sector Availability				
	Economic	Environment and Agriculture	Culture and Education	Transportation	Health
Open Data BCN	Trade, employment, science, and technology.	Air quality, green point, list of associations and environment equipment and activities.	Accommodation, regulated education, list of grants, tourism points, temporary exhibitions.	Parking spots, biking stations, bus stops, street sections, traffic incidence notices, transportation equipment.	Measures the impact of COVID, family planning, hospitals, protected housing, social services centers.
London data store	List of the number of jobs, data on broader economic conditions.	Recycling, household waste, carbon dioxide emission, employment by industry, domestic energy efficiency.	Number of visits by country, annual education, working-age population, international visitors by city.	Number of journeys, traffic flow of vehicles, number of bicycles, road information.	Annual London survey, physically active children, public health.
Data.gov	Loan data, retail, services' annual survey, the survey of business owners, construction price, economic indicators.	Oil and gas well map, clean air status, state soil geographic, earthquake locations, uranium location database, animal or plant diseases.	National register of historic districts, cultural resources, national automotive center, higher education school location, citizen participation.	Traffic data, railroad mileposts, airport runways, pedestrian crashes, bicycles, and pedestrian facilities.	Infant health indicators census tract and year, all live births in Illinois, human disease.
NYC Open Data	Civil service list, NYC jobs, BIC-issued violations, NYC free tax prep sites, payments received for DCA fines.	Water consumption, public recycling bins, natural gas consumption.	Latin cultural organizations, NYC museum recreation, physical education, bilingual education programs, art galleries, tourism grants.	Transportation sites and structure, new driver applications, subway entrances, real-time traffic speed.	Hospitals' facilities, mental health service, health center district.
Data and information	Income and expenditure, employment of Amsterdam, international trade.	—	Sports providers, running routes, primary education, culture in Amsterdam, museums and galleries, hotels.	Traffic forecasting traffic model, mobility of public transport, walking and cycling, metro and tram rail management.	Health districts, social care, city care districts.
Overheid	Income before inflow, income after outflow, purchase data, international trade, transit trade.	Electricity balance, supply, and consumption, supply of natural gas, public electricity network, noise pollution, fruit growing, history of agriculture.	Public libraries, history of education, education expenses, museums' size, class, visitors.	Disabled parking spaces, passenger mobility, travel characteristics, modes of transport, traffic performance.	Serious obesity in children, health expenditure, youth protection programs.
Helsinki Region Info share	Statistical yearbook of Helsinki, income by stage of life and area, income and consumption.	Nature data of Espoo city, energy consumption, district heat production, urban tree database of the city.	Places, events, and activities, grammar school, number of students, art and culture subsidies.	Intersections with traffic lights, signposts for bicycles, pedestrian traffic, parking payment zones.	Deaths and cause of death, comparison of child welfare, care for the mentally disabled.
Chicago Data Portal	Current employee information budget.	Energy usage, average electricity, green roofs map.	Individual Chicago landmarks, public library location, neighborhood boundaries.	Chicago street names, traffic crashes, taxi trips, average daily traffic counts, parking permit zones.	COVID cases, tests, and deaths, neighborhood health clinics, community service centers.
Quebec data	Business register, forest certification, pricing zone, list of work stoppages.	List of large parks, district parks and public spaces, public trees, food inspection, urban agriculture.	Places of interest of the city, all parks and green space, funding granted, administrative data, museum institutions.	Injuries suffered by accident, the road network of the city, sidewalks, and parking.	Daily number confirmed COVID cases, list of days of hospitalization.

Table 2. Cont.

Name	Sector Availability				
	Economic	Environment and Agriculture	Culture and Education	Transportation	Health
Data.Rio	Budget execution revenue.	Air quality, rainfall zones, hydrography.	Proportion of room, special education, the arrival of tourists, basic education development, ranking of national tourists.	Bike racks, cycle network, passenger movement.	Municipal health units, programmatic health areas, pay health insurance.
Dublinked	Location of enterprise centers' contact information.	Public lighting infrastructure, bin locations, local electoral areas, details of bathing water status, noise monitoring.	Location of sculptures, libraries, art centers, third level institutions, school warden duty points.	Bikeshare scheme, bicycle traffic volumes, road infrastructure, traffic congestion.	Locations health centers, contact information.
Data.gov	The location of enterprise centers, industrial estates, annual budget.	Wind energy development, winter service plan, water, power plants, number of farms, public slipways.	Tourism attractions, arts facilities, primary schools, the record of protected structures, number of registered teachers.	Parking of vehicles, road schedule, traffic congestion, traffic lights.	COVID-19 daily statistics, aftercare service, child welfare referrals.
Berlin Open Data	Gastronomy, shops, and other businesses.	List of street tree planting, used glass recycling.	List of memorial plaques, monuments of the state of Berlin.	Road traffic accidents, the volume of vehicles, parking space.	Covid number of cases, Covid number of indicators, staff in the public health service.
GovData	Foreign trade, import, export, employees gross wages.	Monitoring radioactivity, public wastewater treatment, pig and sheep population, land use and harvesting, electricity.	Tourist accommodation, the record of institute.	Wheel counting data, occupied parking spaces, road traffic accidents.	Daily alcohol consumption, diagnostic statistics.
European Union	Eu customs tariff, economic sentiment indicator.	European electricity market, European food consumption, number of dairy cows, greenhouse gas emissions, global surface water exploration.	Number of trips by country, world region of destination, tourism accommodation, classification of European skills.	Airport traffic data, the number of passengers.	Health programs, purity, and potency of drugs.
European data portal	Information of electronic address, goods decelerate, budget.	Land use, protected areas, oil, gas, water, pesticide sale, meat production.	List of courses, library collection, football data.	Public transport, schedule data, vehicles, passengers air transport.	Pharmacy type, number of confirmed COVID cases, deaths by week.

Table 3. Availability of open data initiatives in smart cities by each type of transportation.

Open Data Name	Type of Transport						
Name	Car-Taxi	Bus	Bike	Metro	Tram	EV/UAV	Airplane/Ship
Open Data BCN	YES	YES	YES	YES	YES	YES/NO	NO
London data store	YES	YES	YES	YES	YES	YES/NO	NO
Data.gov	YES	YES	YES	YES	YES	YES	YES
NYC Open Data	YES	YES	YES	YES	NO	Yes/NO	NO/ YES
Data and information	YES	YES	NO	YES	YES	NO	NO
Overheid	YES	YES	YES	YES	YES	NO	NO
Helsinki Region Infoshare	YES	YES	YES	YES	YES	NO	NO/YES
Chicago Data Portal	YES	YES	YES	NO	NO	NO	NO
Quebec data	YES	YES	NO	YES	NO	NO	NO
Data.Rio	YES	YES	YES	YES	NO	NO	NO/ YES
Dublinked	YES	YES	YES	NO	NO	NO	NO
Data.gov	YES	YES	YES	NO	NO	YES/NO	NO/ YES
Berlin Open Data	YES	YES	YES	NO	YES	NO	NO
GovData	YES	YES	YES	YES	YES	NO	NO/ YES
European Union	YES	YES	NO	YES	YES	NO	NO/ YES
European data portal	YES	YES	YES	YES	YES	YES/NO	NO/ YES

2.2. Optimization, Simulation and Machine Learning in ITS

Large amounts of data are generated daily by web-based services, mobile devices, and sensors in ITS. With that, new approaches for data-based transportation systems have emerged Vlahogianni [19]. In modern ITS, data play an important role in solving problems related to congestion control, peak load reduction, mobility management of EVs, etc. For instance, Saharan et al. [20] reviewed and analyzed dynamic pricing techniques in the ITS area.

Based on a multi-agent system, Satunin and Babkin [21] proposed a new approach to design a demand-responsive transport model, where the interests of the system stakeholders are represented by different independent agents. Moreover, according to combinatorial auctions, they proposed an algorithm that allows expression of commodities of multiple transportation scenarios with obvious means of offers. Furthermore, Shah et al. [22] developed and compared a mixed-integer programming model with a space–time network flow model for scheduling vehicles on a grid of intersecting roads. Then, they showed that the proposed space–time network model is more efficient than other approaches. Furthermore, using real-time data on travel times, Taniguchi and Shimamoto [23] showed that the total cost decreased by using their dynamic vehicle routing and scheduling model.

In the ITS area, one of the main goals is to support efficiency, safety, and eco-friendly transport networks that improve the quality of life. Based on a cellular automata, Marques and Neves-Silva [24] developed a traffic simulator called GESTRAF. It considers the individuality factor. By using data in Portugal, Ramos et al. [25] described and developed a useful framework that includes a core modeling and simulation platform. This framework can be adapted to the urban transportation system. Fernández-Isabel and Fuentes-Fernández [26] presented another simulation model for analysis of ITS. Their framework considers modeling languages, guidelines, and tools to develop ITS specifications and simulations.

Using simulation for car-sharing management dates back to the 70s, due to the fuel crisis in the United States and the lack of federal funding for new urban transportation facilities. Hence, Kornhauser et al. [27] developed a simulation for the cities of Trenton and New Jersey, in order to evaluate the productivity potential of dynamic ride-sharing

systems on a hypothetical automated guide-way transit network. According to the number of specific origins and destinations that vehicles can travel to at any time, different policies were tested. Agent-based and dynamic simulation have been the most frequently used methods to deal with car-sharing challenges. Based on agent-based modeling, by using New York city fleet data, Lokhandwala and Cai [28] considered implicit objectives such as reducing the fleet size, increasing the occupancy rate, decreasing the total travel distance, and reducing carbon emissions. Moreover, their simulations demonstrated that the total travel distance was decreased by up to 55%. In terms of using shared autonomous vehicles, Fagnant and Kockelman [29] used dynamic ride-sharing with aim of the optimizing fleet sizing, improve the model's capabilities, and deliver a benefit–cost analysis for fleet operators.

ITS management has become more efficient due to the application of deep learning and machine learning techniques that perfectly complement other analytical and statistical techniques. This, in turn, has facilitated traffic management and traffic planning, enhanced safety and security in transit roads, reduced maintenance costs, and optimized public transportation as well as ride-sharing performance [30]. Thus, for example, Fang et al. [31] proposed a support vector machine to classify user transportation and vehicular modes after considering different machine learning methods. In addition, Said et al. [32] applied Q-Learning and reinforcement learning techniques by using a support vector machine to select the best transportation route based on CO₂ emissions, travel duration, ticket tariff, waiting connection time to catch transport means, and connection time between the different transport means to reach the destination. Nguyen et al. [33] developed a useful review and showed which deep learning applications are more efficient in transportation networks, such as traffic flow forecasting, traffic signal control, automated vehicle detection, travel demand prediction, autonomous driving, and driver behavior analysis. Moreover, Li and Xu [34] classified vehicles using different classifiers, such as Adaboost, support vector machines, reinforcement learning, support vector regression algorithms. They used the latter to improve the accuracy of short-term traffic flows. Likewise, Karami and Kashef [35] introduced intelligent planning data, methods, and models for transportation, and discussed clustering techniques that automatically group more accurate information, and tested some useful machine learning methods in time series prediction: ARIMA, Kalman filtering, Holt winters' exponential smoothing, random walking, KNN algorithms, and deep learning. Similarly, Boukerche and Wang [36] reviewed different machine learning methods in the field of traffic prediction. With the rapid rise of traffic monitoring, significant challenges arise regarding the storage, communication, and processing of traditional transportation systems based on cloud computing. For instance, Chen et al. [37] propose a traffic flow detection scheme based on deep learning on the edge nodes.

Based on K-fold cross-validation and out-of-bag error, Jahangiri and Rakha [38] modeled a selection process for extracting data from smartphone sensors. This process considered different supervised machine learning models, such as K-nearest neighbor, support vector machines, decision tree, bagging, and random forest. The authors developed multi-class classifiers that identify the transportation model, e.g., driving a car, using a bus, riding a bicycle, walking, and running. By using some information of public transportation and geographical location from daily mobility, Omrani [39] was able to predict the travel mode employed by citizens in Luxembourg city. Furthermore, after considering several machine learning methods, they selected artificial neural networks for the job. In a similar way, Gal et al. [40] considered both historical and real-time data associated with the bus network system in the city of Dublin to develop a hybrid method combining queuing theory and machine learning. They employ this method to predict travel times in scheduled bus routes.

2.3. Agile Optimization Algorithms in ITS

Many real-life applications in ITS require real-time decision-making. Especially in transportation, urban centers are continuously exposed to plenty of dynamic situations,

which highlights the need for smart and agile decision making. Examples of these situations occur under the rupture of such systems, where intelligent systems must be able to quickly react and provide users with smart alternatives to the previously computed ones.

During the last few decades, many solution methods have been proposed for tackling many stochastic and non-stochastic combinatorial optimization problems. From the deterministic world, the use of approximate solution approaches—e.g., heuristic and metaheuristics—has gained singular notoriety due their capability for providing near-optimal (or even optimal) solutions in a reasonable amount of computational time [41]. Alternatively, simulation–optimization approaches have proved satisfactory when dealing with stochastic optimization problems. These algorithms facilitate the consideration of risk and reliability analysis during the assessment of alternative high-quality solutions [42,43].

Obtaining high-quality solutions for large-scale problems in real-time is a major challenge. It sometimes assumes the re-optimization of the model, as inputs and constraints are modified dynamically due to the incorporation of new data or changes in the environmental conditions. Heuristics are extremely fast solution approaches that are designed to solve a specific problem. These deterministic procedures can be extended in a probabilistic algorithm using biased-randomization techniques [12]. These techniques consist in employing skewed (non-symmetric) probability distributions to smooth the taking of decisions during the construction of a solution. Under these circumstances, the resulting biased-randomized algorithm (BRA) can be executed multiple times, thus generating alternative solutions of similar quality. Moreover, BRAs can be used to extend and enhance the performance of many classical metaheuristics approaches [44].

The BRAs are extremely fast and only require up to a single parameter to be calibrated. In this way, they can be naturally executed in parallel. Accordingly, the idea behind agile optimization (AO) algorithms relies on the employment of multiple CPU/GPU cores to concurrently run a large number of threads, each being responsible for virtually executing a run of the BRA (Figure 2). Consequently, multiple runs of a BRA are simultaneously processed (at virtually the same time as the one required by a single execution of the original heuristic). As a result, a pool of multiple alternative solutions is generated, being, finally, the best solution returned by this procedure. By taking advantage of both the optimization and parallel computing worlds, AO strategies allow for: (i) the finding of efficient solutions to large-scale and NP-hard optimization problems in real-time; as well as (ii) periodically re-optimizing the model as the inputs and constraints are dynamically modified due to the arrival of new data or to changes in the environmental conditions. This is the case, for instance, of goods transportation in humanitarian logistics, where routing plans must be provided in real-time in order to save lives [45]. Another example refers to connecting vehicles in motion to roadside units, where the dynamic movement of vehicles requires their re-assignment to these units in real-time [46]. Similar requirements can be also found in ride-sharing operations in smart cities, where requests, demands, and traffic conditions alter the expected natural operation of this environment [2].

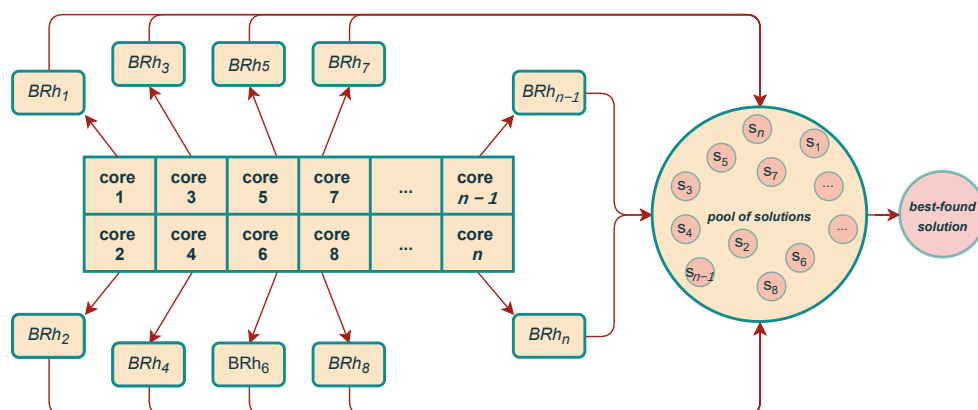


Figure 2. The concept of agile optimization.

3. Related Work

This section reviews some related work in the context of cloud, fog, and edge computing, as well as on the use of data analytics in ITS.

3.1. Cloud, Fog, and Edge Computing in ITS

When applied in the context of ITS, cloud, fog, and edge computing techniques facilitate the transmission and processing of terabytes of data in real-time. These data are provided by smart sensors networked with IoT devices, through which they are transmitted to the decision-making system unit in the cloud, where many issues arise. Mainly, a huge load on servers and cloud systems is created. To reduce this overloading on the cloud, technologies such as edge and fog computing were developed in various manners. For instance, fog computing provides a better infrastructure for the interaction between cloud computing and IoT devices, thus promoting their spread, data mining, and analysis in order to optimize the use of these resources [47]. Some characteristics of fog computing were discussed in Bonomi et al. [48], such as low latency, extensive distribution, mobility, and a wide range of nodes that could create a new type of application and services of IoT in connected vehicles, smart grid, and smart cities. In this regard, Peter [49] found the ability of fog computing as a suitable platform for IoT to handle the data overflowing and resolve the problem related to congestion and latency. Later, Bierzynski et al. [50] considered the possible way to combine fog and cloud computing to support the solution of IoT challenges. In addition, in this context, Chen et al. [51] used an edge-computing system for IoT based on smart grids, fully realizing the demand for high bandwidth with a low latency problem. They proposed the privacy protection strategy through edge computing, a data prediction strategy, and a pre-processing strategy of hierarchical decision-making based on task grading.

Allocation of applications to fog and edge nodes is one of the big challenges in this area. Some applications may require larger computing capacity. Hence, techniques for grouping and classifying resources in virtual nodes with large computational capacity are necessary for the co-existence of computational tools with different computations. Due to this issue, based on combining semantic description of resources with semantic clustering techniques, Xhafa et al. [52] present some clustering techniques for creating virtual computing nodes from fog/edge nodes and, by using these clusters via heuristics and linear programming, creating an optimal allocation of applications to virtual computing nodes. Additionally, due to the growth of IoT and 5G telecommunication networks, Gohar and Nencioni [53] provide an overview of the benefits of 5G in an economic and technological context, and discuss impacts and concepts of 5G for ITS from various dimensions.

Apart from providing support to efficiently manage data, cloud computing and IoT hold the potential of improving problems such as security, privacy, scalability, reliability, and quality of service in cloud and vehicular networks. Particularly, Yan et al. [54] introduced challenges in security and privacy in their work. For instance, challenges of high-mobility vehicles' verification, scalability, mixed identities and locations, and the complexity of establishing trust relationships between multiple players due to the intermittent short-range communications were addressed. Moreover, to show the suitable security architecture that manages many of the challenges in vehicular clouds, they provided a directional security architecture. Wang et al. [55] provided a collaborative vehicular edge computing for a qualified vehicular network that has the flexibility to support the collaboration between both horizontal and vertical dimensions. This flexibility can be distributed more efficiently at network edges. Furthermore, based on cloud computing and IoT technologies, He et al. [56] presented a multilayered vehicular data cloud platform by proposing new capable software architecture for vehicular data cloud in the IoT environment. The objective is to integrate numerous devices available within vehicles and devices in the road infrastructure. On this matter, Dobre and Xhafa [57] provided services based on context-aware and data-intensive applications to support challenges that are tied to big data management to foster a better understanding of traffic problems in

large cities. Moreover, Xiao and Zhu [58] presented an intuitive theory about vehicular fog computing that transforms connected vehicles into mobile fog nodes and used the mobility of vehicles—such as buses and taxis—to offer cost-effective and deliver computed resources to anywhere it is requested.

According to the rapid rise of data produced by devices and sensors, Tanganelli et al. [59] presented a distributed hash table that is executed to design discovery services for fog computing platforms with mobile nodes. This is done by creating multiple attributes and range queries in order to use their storage and computing capabilities in ITS. Furthermore, Badidi et al. [60] reviewed current service delivery models in edge and fog computing in smart cities, and proposed a fog-based data pipeline for IoT data management messaging systems. Zhang et al. [61] considered developing and deploying data-driven ITS that has the capability of vision, multi-sources, and learning to optimize its performance. They also have identified some issues such as missing values, data cleaning, dimension reduction, sparse learning, and heterogeneous learning for further research to improve the transportation systems. Minh et al. [62] suggested efficient decentralization of internet-connected vehicle data and services at different levels of intelligence and complexity in the fog computing model and optimized the edge-based virtualized resources. Through integrated intelligent computing, real-time data analytics, and the internet of vehicles, Darwish and Abu Bakar [63] proposed a new architecture and reviewed the challenges and opportunities of implementing fog computing and real-time data analytics in the area of the Internet of vehicles. Raza et al. [64] developed a vehicular edge computing architecture to support a high level of scalability and a suitable model for vehicles to reduce waiting time for services that require real-time decision making. Brennand et al. [65] used fog computing in traffic services to control congestion in order to reduce the problem caused by the traffic jam. They proposed a distributed approach based on a classification system to suggest new routes to vehicles.

Fog computing supports a variety of growing applications, including those in IoT and new generations of wireless systems (5G/6G), which are recognized as an important area of future technology. In this regard, Chiang and Zhang [66] showed the opportunities and challenges associated with the use of fog computing in the networking context of IoT. In addition, as cloud-based 5G infrastructure is not efficient for some high-demand applications such as transportation, tactile Internet, and augmented reality, and because of the extra cost for more latency of computation and storage, Tufail et al. [67] proposed multi-access edge computing technology for smart cities' environments. Based on vehicular networks and features of device-to-device communications, Cheng et al. [68] developed a feasible study of device-to-device for ITS. They also improved general system performance in ITS by considering the interference control mechanism, predictive resource allocation methods, and roadside unit cooperative scheduling. Camacho et al. [69] inspected the 5G architecture designed with software-defined networking and presented potential challenges in wireless technologies to provide vehicle connectivity such as connected vehicles or self-driving cars. For a proper allocation of resources, Lee et al. [70] proposed an auction-based scheme that consists of a blockchain mechanism. Among its advantages, there is the consideration of a fog-enable ITS model to allow vehicles on the Internet of vehicle networks to use the services provided by roadside units. Moreover, by carrying out an ordering service at the data center with the hyper-ledger fabric platform, they prevented the overuse of computing resources for proof-of-work and validation. In addition, Raza et al. [71] proposed a vehicle-to-everything communication model based on ultra-high-speed and ultra-low latency of integrated networking technologies—such as mobile edge computing, fog, and cloud computing—for managing traffic and monitoring systems sustained by a full automated ITS environment.

3.2. *IoT Analytics in ITS*

IoT analytics is one of the core tasks in any ITS, since it takes care of gathering a large amount of data and transforming these data into useful information, developing

predictive models, and supporting intelligent decision making [72]. IoT develops digital services and functions for different groups of users and creates smarter cities and villages. Analyzing the mobility of digital devices, such as phones and vehicles, allows for the understanding of relationships between sensors and energy, and between sensors and data storage. Due to the strong use of IoT and Big data in ITS, and based on analyzing different road traffic problems, Dai and Ma [73] helped to improve transportation management by optimizing the traffic. Using vehicles and communication within entities in road traffic scenarios generates huge raw data in ITS. In order to understand raw sensor data in ITS, Swarnamugi and Chinnaiyan [74] address the enforceability of techniques for modeling and the reasoning approaches in ITS. They also identify the used machine learning and deep learning techniques in the reasoning phase of ITS. Mohandu and Kubendiran [75] reviewed data analytics insight for the transport and mobility industry, ITS implementations, threshold instances, and in some use cases, including routing, planning, platform architecture, among others. Calabrese et al. [76] addressed techniques to extract efficient mobility statistics for transportation research. When integrated with statistical analysis, mobile phone traces demonstrate a reasonable proxy for individual mobility that can help to understand the intra-urban variation of mobility and the non-vehicular component of overall mobility. In two different analysis levels, such as GPS-data traces and street segments, Jiménez-Meza et al. [10] used GPS-data (data-time, latitude, and longitude) to estimate travel time, distance, and speed. Then, they characterized street segments by calculating the level of services by using the average speed. Later, Zanella et al. [77] reviewed and analyzed the ability technology, protocols, and architecture for urban IoT and tried to create a block to find out an integrated urban scale of information and communication technology platform by using practical cases in Padova, Italy.

Mahdavinejad et al. [78] evaluated different machine learning methods to deal with IoT data challenges in smart cities. They classified machine learning algorithms and explained how different techniques can be applied to extract more accurate information. Graser [79] presented a new python library to concern movement data based on the pandas' data analysis library and the GeoPandas extension. They analyzed existing frameworks and implementations to define the main function required for a movement data analysis library and show its usefulness in stand-alone python scripts. Pappalardo et al. [80] presented data cleaning difficulties related to raw spatio-temporal trajectories, and created an algorithm to synthetically generate trajectories able to reproduce the realistic law of human mobility. Bao and Liu [81] mathematically analyzed the transportation system and explained how to build a multi-agent deep deterministic policy gradient system to optimize real-time signal control policies in emerging large-scale ITS. Moreover, Teng et al. [82] proposed a low-cost code dissemination scheme which called vehicles as code mules in a smart city, where code stations are deployed to obtain the updated code from the cloud data center and send it to the code mules. They optimized the code selection scheme and greedy deployment scheme to maximize coverage of code dissemination over the city with low cost and duration time.

Cao et al. [83] proposed an edge computing platform by using descriptive analysis to discover significant patterns from real-time bus transit data streams. These authors found the potential of applying this platform on applications such as autonomous vehicles, smart intersections, and smart traffic light systems. By considering bus information, Yongjun et al. [5] designed a system based on IoT and GPS, quickly collecting accurate data with the collector, sending it to the centralized dispatch center and the corresponding database system. The system then reports information, such as current urban traffic conditions, present bus location, arrival time, and bus line in the short run. Hence, based on real-time traffic conditions, the monitor center can complete the bus schedule. Panadero et al. [84] solved the stochastic version of the team orienteering problem, which is related to unmanned aerial vehicles. These authors proposed a simheuristic algorithm [43], which combines a biased-randomized heuristic [85] with simulation techniques to provide an 'agile' optimization methodology [46]. Moreover, Adi et al. [86] reviewed the process of machine learning analysis in IoT data generation, and developed a framework to make

suitable applications to learn from other IoT applications. Finally, Wei et al. [87] considered bus line optimization based on the metro–bus relationship and competition. They also developed a quantitative methodology to evaluate their approach.

Table 4 shows a comparison between different characteristics addressed by the reviewed literature in the context of cloud, fog, and edge computing, as well as in the use of data analytics in ITS. Most of the reviewed works focus on the concepts of cloud, fog, and edge computing to manage the processing of huge amounts of data, and improve their performance to provide better services. Furthermore, only a few works address optimization and ML techniques by considering open data servers to solve real life problems in ITS. In order to narrow this gap, we address the usage of a combination of fog and edge computing in ITS by using open repositories. This is useful to provide real-time information, which can feed our developed algorithms for solving the DRSP.

Table 4. Comparison of the reviewed work.

References	Year	Use Case	4G/5G/6G	IoT	Fog	Cloud	ITS	Opt	ML	Data Analysis
[47]	2019	Self-driving vehicles, sensors raw data		X	X	X	X		X	X
[48]	2012	Fog characteristics		X	X					
[49]	2015	Analyzed Fog Computing and its real time applications		X	X	X				X
[50]	2017	Combination of cloud, fog and edge computing		X	X	X				
[51]	2019	Edge computing system for IoT-based		X		X				
[52]	2021	Optimization methods, semantic clustering		X	X	X	X	X		
[53]	2021	Various dimensions of 5G on ITS	X	X		X	X			
[54]	2012	Analyze security challenges, potential privacy threats in vehicular clouds				X	X			X
[55]	2018	Collaborations different edge computing, vehicular edge computing	X		X	X				
[56]	2014	Multilayered vehicular data cloud, cloud computing and IoT technologies		X		X	X	X		X
[57]	2014	Analysis challenges of next-generation Big Data services CAPIM platform, Context-Aware Framework		X		X	X			
[58]	2017	Connect vehicles into mobile fog nodes	X	X	X	X				
[59]	2017	Service for fog computing with mobile nodes			X		X			
[60]	2020	Fog-based data pipeline	X	X	X	X	X			
[61]	2011	Multifunctional data-driven intelligent transportation system					X	X	X	
[62]	2018	Context-aware fog computing, multiple intelligent		X	X	X	X			
[63]	2018	Real-time ITS big data analytics in the Internet of vehicles		X	X	X	X			X
[64]	2019	Vehicular Edge Computing architecture, smart vehicle	X	X	X	X	X			
[65]	2019	Traffic service, control congestion, roads classification	X		X	X				
[66]	2016	Challenges of fog and IoT	X	X	X	X				
[67]	2021	Multi-access edge computing	X	X	X	X				
[68]	2015	Device-to-device communications. vehicular networks					X			
[69]	2018	Wireless technologies, vehicle-to-x connectivity	X	X			X			
[70]	2020	Vickrey–Clarke–Groves auction mechanism, road side unit			X	X	X			
[71]	2018	Vehicular-to-Everything, cellular 5G technologies, roadside infrastructure	X	X	X	X	X			
[72]	2019	IoT-ITS system, multiple regression analysis, multiple discriminant analysis and logistic regression		X			X			X
[73]	2021	Road traffic, traffic information		X			X	X		X
[74]	2021	Context-awareness in the Intelligent Transportation System		X	X	X	X		X	X
[75]	2021	ITS, Big data analytics		X		X	X			X
[76]	2013	Mobile phone data, mobility information		X			X			X
[10]	2013	Vehicles, travel time, distance					X			X
[77]	2014	Urban IoT system		X		X	X			X

Table 4. Cont.

References	Year	Use Case	4G/5G/6G	IoT	Fog	Cloud	ITS	Opt	ML	Data Analysis
[78]	2018	Smart cities IoT data	X	X	X	X		X	X	X
[79]	2019	Movement data, python library								X
[80]	2019	Mobility analysis							X	X
[81]	2019	Spatial influence-based communication, multi-agent Deep Deterministic Policy Gradient					X	X		X
[82]	2019	Mobile vehicles, taxi trajectory		X		X	X	X		X
[83]	2017	Mobile edge nodes, transit bus data	X		X	X				X
[5]	2012	Vehicle subsystem, the station subsystem and the monitoring center					X			X
[84]	2018	Team orienteering problem, simheuristic algorithm, biased-randomized heuristic, simulation techniques					X	X		
[86]	2020	Data generating for IoT, IoT applications		X	X	X	X		X	X
[87]	2020	Bus line optimization, metro integration						X		

4. An Illustrative Case Study

We have developed a case study in order to illustrate the previously described concepts. Our example addresses a DRSP, in which events such as traffic conditions, new service requests, or service cancellations can change the originally designed routes after the vehicles have already departed from their origins. The core idea of the ride-sharing problem is to foster that personal private vehicles are shared by a group of people, instead of being used only by the car owner. Nowadays, the massive use of apps and interconnected smartphones facilitates the immediate contact between drivers and users for sharing trips. Furthermore, ride-sharing activities provide multiple benefits for drivers, users, and the entire community [88], such as the reduction in costs, pollution, and traffic congestion.

The static version of the ride-sharing problem [2] consists of a finite set of capacitated heterogeneous vehicles, each one driven by an individual owner, who offers empty seats to users with similar itineraries. Each user requests a service, providing their current location, and drivers pick them up in these locations. This means that drivers have some kind of flexibility to adapt their routes so they visit the pickup point. Moreover, the vehicle capacity allows for more than one user to be transported. Hence, the route performed by each vehicle consists of an origin point (each driver's home), a set of locations where the driver picks up the users, and an arrival point. We assume that a driver can pick up each user only if the destination of all of them is the same. However, the destination points can be the same or different for each vehicle. Since the vehicle capacity and the number of vehicles are limited, not all users requesting a service can be picked up. Hence, the challenge is not only to design the routes, but also to select the users that will be picked up. This selection process is carried out based on both the distance between the user location and the driver's origin and destination points, as well as the fee that is paid by the user to the driver for being transported. The objective of the ride-sharing problem is then to maximize the total collected fee. Figure 3 displays an example of a complete solution for the static version of the addressed problem. Connected houses represent the users who are served by the vehicles, whereas non-connected houses represent the non-served users.

The static version of the ride-sharing problem assumes that, once all routes have been designed, they cannot be further modified. Nevertheless, real-world events, such as traffic conditions, new orders, or cancellations, may lead to make changes in the original route plans. Since these events occur frequently when vehicles are already in route, a DRSP allows for the design of new routes, which include only those users who have not yet been picked up. This redesign process is performed in discrete time intervals. Formally speaking, the DRSP can be defined on a directed graph $G(N, E)$, where N is the set of nodes, and E is the set of edges linking these nodes, i.e., $E \subseteq N \times N = \{(i, j) \mid i \in N, j \in N, i \neq j\}$. Three subsets of nodes are considered, such that $N = I \cup O \cup A$. I is the subset of nodes where the users are located, O is the subset of drivers' / vehicles' origin nodes, and A is the subset

of final destination nodes. Each pickup point $i \in I$ has a known fee f_i , which is paid by each user for being transported. Traversing each edge $(i, j) \in E$ has a deterministic cost c_{ij} . Routes are performed by a set K of vehicles. The capacity b_k of each vehicle $k \in K$ is known as well. Each pickup point $i \in I$ must be visited only once, and each vehicle $k \in K$ is assigned to only one route. Only a subset of nodes $J \subset I$ can be visited. Each route starts in an origin node $o \in O$, traverses a subset of nodes $H \subset J$, and finishes in a destination node $a \in A$. If d_h is the demand of each node $h \in H$, then $\sum_{h \in H} d_h \leq b_k$. If t is the time interval set to recalculate the routes, then this process is always performed in the time $\tau = nt$, where $n \in \mathbb{N}$. This recalculation is performed iteratively until all vehicles have arrived to their respective destinations. Furthermore, if L_n is the subset of non-visited nodes in the period n , such that $L_n \subseteq J$, then the route recalculation is performed only for the nodes in both L_n and A . This assumption is helpful to respect the commitment of serving the originally selected customers, taken in the period $n = 0$. We assume that routes are only affected by traffic conditions, i.e., new orders and cancellations are not allowed in our addressed version of the problem. Hence, our problem consists in designing a set of $|K|$ routes that meet the aforementioned constraints, such that the total collected fee is maximized.

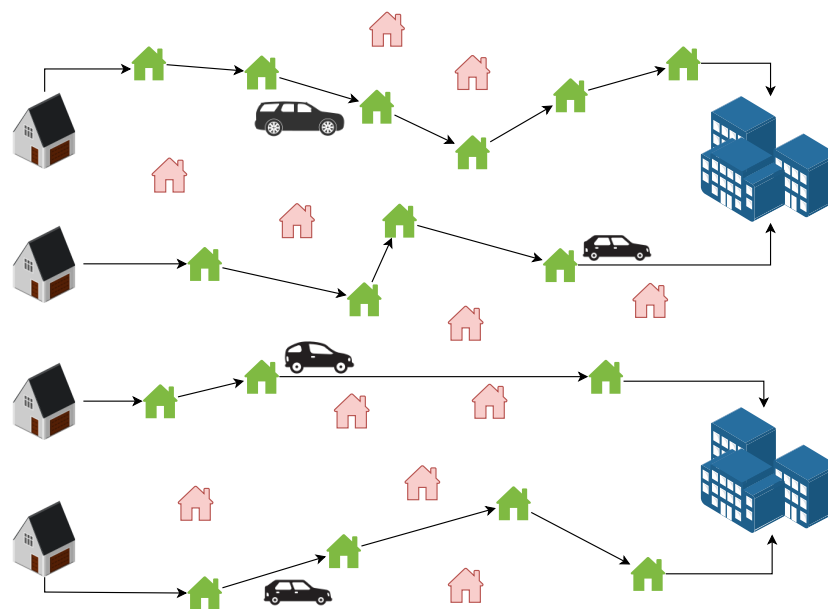


Figure 3. Representation of a complete solution for a static RSP.

5. Solution Approach

In this section, we describe our proposed methodology for solving the DRSP. This methodology is based on a discrete-event heuristic [89], which generates promising solutions according to events that occur over time. These events are related to circumstances in which the system has to react appropriately, i.e., changes in traffic conditions involve re-planning routes to contemplate them. Discrete-event constructive heuristics are based on the use of discrete-event simulation to handle time dependencies that arise as the solutions are constructed. In our case, the basic idea is to complete a discrete-event (time-based) simulation of arrival, departures, and re-planning of routes, so that vehicles can re-plan routes according to the traffic conditions provided by the open data server to minimize the travel time towards the final destination. This re-planning procedure is performed at each time interval t . Hence, any event can belong to one of the following types: vehicle delivery, vehicle arrival, and traffic update. Each event is associated with a vehicle and a trigger time. Likewise, the vehicle is assigned to a current trip (between two customers) and the current route that is covered.

The flowchart of our solving approach is presented in Figure 4. At the beginning (period $n = 0$), the algorithm produces an initial static planning without considering any traffic conditions, that is, all the information employed during this stage is not modified. In addition, the list of events is initialized adding one departure event for each available vehicle, and a traffic update event. Departure events are programmed to occur at the period $n = 0$, while the traffic data event arises in the period $n = 1$. Each period n lasts t time units, which is established as an input parameter. The main loop iterates over a list of events that occur during the execution of the routes until this list is empty. At each iteration, the algorithm takes the first event and proceeds according to the event type. In the case of a departure event, the vehicle located at the customer departs to the next destination of the current trip, hence, a new arrival event is scheduled, considering the travel time and the current traffic congestion. If the event is an arrival event, two possibilities could be given: the vehicle arrives either to a customer or to the final destination. In the case of the former, the vehicle must stop to perform a pick-up action. Then, the vehicle capacity availability is updated with the passenger demand. In addition, the algorithm creates a new vehicle departure event from this customer. In the case of the latter, the vehicle arrives to the final destination and, thus, no further action is required. Finally, for the case of the traffic update event, which occurs in each period n , the algorithm re-plans the vehicles' routes, considering the current traffic data, from the next stop to the final destination.

Algorithm 1 outlines the heuristic for solving the DRSP in a given period n . This approach is a two-stage heuristic algorithm capable of providing a good trade-off between solution quality and required computational effort. The input parameters are: a list of customers, where each customer consists of location coordinates, passenger demand, and fee to pay; a list of vehicles, where each vehicle is composed of the coordinates of the origin and final destination, and seats capacity; and the α and β parameters for computing the savings list and the biased-randomized heuristic, respectively. In the first stage, the original problem is divided into small sub-problems (clusters) according to origin points and destination points. Notice that different sub-problems might share some of the pick-up locations. In the second stage, each cluster is solved by applying a savings-based heuristic proposed by Panadero et al. [90]. This heuristic involves the following steps: (i) generation of a dummy solution where each pick-up point (node) is connected by one route (vehicle) with the origin and the final destination, and (ii) construction of the enriched savings list of edges, where each savings value is related to an edge that links whichever location such as origin, destination, and pick-up point. This enriched savings value is computed as $s_{ij} = \alpha(c_{im} + c_{oj} - c_{ij}) + (1 - \alpha)(\mu_i + \mu_j)$. The input parameter α is set within $(0, 1)$; c_{ij} denotes the traveling time between i and j . Likewise, 0 and m are the origin and destination nodes, respectively. Finally, μ_i and μ_j are the assigned fees at each node. These savings values consider both the traveling time and the aggregated fee collected by visiting both locations i and j . The main loop iterates while the list is not empty. For each iteration, the edge at the top of the list is chosen, then, the associated routes are merged if and only if the resulting route does not exceed the vehicle capacity; otherwise, the edge is rejected.

This heuristic is deterministic because the merging process always selects the first element of the savings list. We extend this heuristic introducing a biased-randomization process [85] in order to produce a variety of solutions without losing the logic behind the original heuristic. As explained in Section 2.3, we employed the geometric probability distribution, $\text{Geom}(\beta)$ with $\beta \in (0, 1)$, to introduce a biased-randomized behavior. In Algorithm 2, the biased-randomized heuristic is executed for a maximum number of iterations or computational time, resulting in a multi-start approach [91]. Therefore, several feasible and promising solutions are generated, and the one with the highest collected fee is returned. The input parameters for Algorithm 2 are the same as those for Algorithm 1.

Algorithm 1 Two-stage approach algorithm for solving a static RSP

```

1: input:
2:   customers: list of customers
3:   vehicles: list of vehicles
4:    $\alpha$ : parameter for computing the savings list
5:    $\beta$ : parameter for the geometric distribution
6: end input
7: function Biased-Randomized Algorithm(customers,vehicles, $\beta,\alpha$ )
8:   clusters  $\leftarrow$  computeClusters(customers,vehicles)
9:   for  $cluster_k$  in clusters do
10:     $sol_k \leftarrow$  dummySolution( $cluster_k$ )
11:    savings  $\leftarrow$  genSavingsList( $cluster_k,\alpha$ )
12:    while savings  $\neq \emptyset$  do
13:       $edge_{ij} \leftarrow$  pick(savings, $\beta$ )
14:       $r_i, r_j \leftarrow$  getRoutes( $edge_{ij}$ )
15:      if checkMerging( $r_i, r_j, vehicle(cluster_k)$ ) then
16:         $route_i \leftarrow$  merge( $route_i, route_j$ )
17:         $sol_k \leftarrow$  replace( $sol_k, route_i$ )
18:      end if
19:      savings  $\leftarrow$  remove(savings, $edge_{ij}$ )
20:    end while
21:    sol  $\leftarrow$  add(sol, $sol_k$ )
22:  end for
23: return sol
24: output:
25:   sol: a solution
26: end output

```

Algorithm 2 Multi-Start Approach for solving RSP

```

1: input:
2:   customers: list of customers
3:   vehicles: list of vehicles
4:    $\alpha$  and  $\beta$ : parameters required by the BRA heuristic
5: end input
6: function Multi-Start(customers,vehicles, $\beta,\alpha$ )
7:   bestSol  $\leftarrow$  Heuristic(customers,vehicles)
8:   while end not reached do
9:     sol  $\leftarrow$  BRA(customers,vehicles, $\beta,\alpha$ )
10:    if fee(sol) > fee(bestSol) then
11:      bestSol  $\leftarrow$  sol
12:    end if
13:  end while
14: return bestSol
15: output:
16:   bestSol: the best solution
17: end output

```

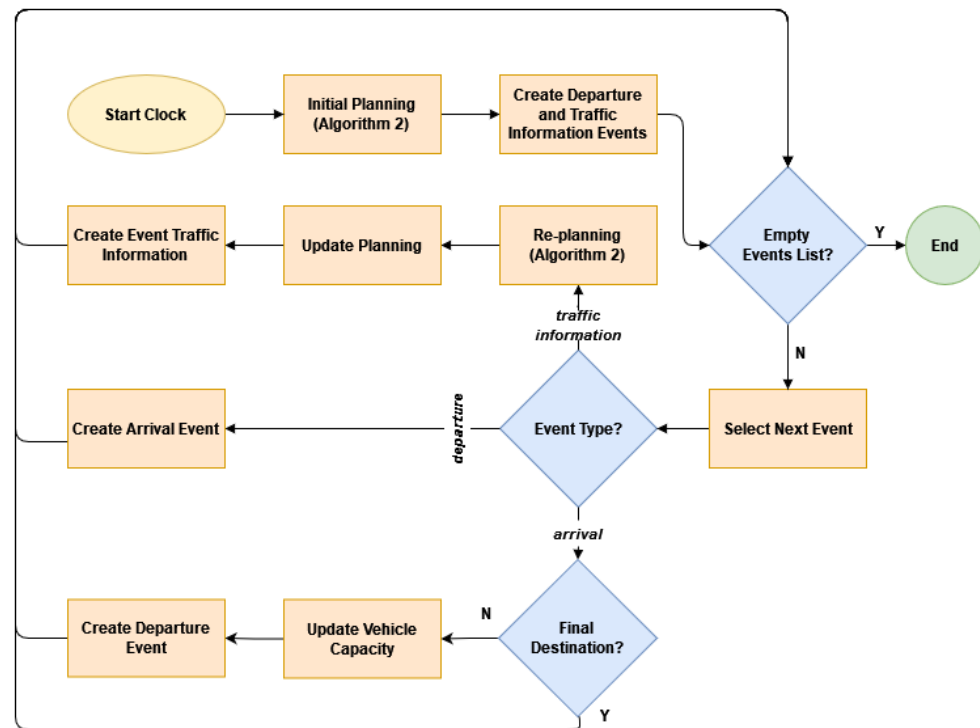


Figure 4. Discrete-Event Algorithm flowchart.

6. Computational Experiments and Results

A series of numerical experiments were designed to test our approach. A total of 27 instances with different characteristics were tested. The instance name in Table 5 sets both the number of customers requesting a service and the number of available vehicles. For example, *drsp63x6-1* is the first instance in the list considering 63 potential customers and 6 vehicles. Hence, three groups of instances with different sizes are tested. Each potential customer's demand and location were generated randomly. Available vehicles are heterogeneous in each instance, with capacities varying between 4 and 8 users. The aggregated capacity of all vehicles is proportional to the total demand. Only for experimental purposes have the traffic conditions also been generated randomly for each edge $(i, j) \in E$ in each period n . These conditions are represented by a coefficient w_{ij}^n , which was generated according to a uniform probability distribution, such that $w_{ij}^n \sim U(0, 1)$. For real-world cases, w_{ij}^n can be computed after retrieving the corresponding traffic information from open data repositories. w_{ij}^n affects the cost c_{ij} incurred when traversing this edge. For instance, if c_{ij}^0 is the time spent for going from customer i to customer j in the period 0, i.e., before the vehicles' departure, then the simulated real time for traversing the edge (i, j) in the period n , affected by the traffic conditions, is computed according to Equation (1). Finally, the time interval that triggers the re-computation of routing plans is set in $t = 10$ time units. Notice that the number of times that routes are re-computed is instance-dependent, since these new computations are performed iteratively until all vehicles have arrived at their corresponding final destinations. The algorithm was implemented in Python 3.8. The experiments were carried out on an i7-8750 CPU at 2.20 GHz with 16 GB of RAM memory installed. The time limit for the biased-randomization process was set as 1 s (in order to keep the real-time condition).

$$c_{ij}^n = c_{ij}^0(1 + w_{ij}^n) \quad (1)$$

Table 5. Obtained results by our algorithm for a DRSP.

Instance	Served Customers	Total Collected Fee	OBS Cost	OBD Cost	Gap
drsp43x4-1	17	237	595.80	384.14	−35.53%
drsp43x4-2	18	295	587.88	492.58	−16.21%
drsp43x4-3	20	373	512.51	398.23	−22.30%
drsp43x4-4	22	394	502.89	402.97	−19.87%
drsp43x4-5	15	223	582.33	374.95	−35.61%
drsp43x4-6	22	347	528.00	528.70	0.13%
drsp43x4-7	16	261	497.84	367.13	−26.26%
drsp43x4-8	23	468	528.30	455.39	−13.80%
drsp43x4-9	18	313	650.22	418.83	−35.59%
Average	19.00	323.44	553.97	424.77	−22.78%
drsp63x6-1	33	676	685.87	635.04	−7.41%
drsp63x6-2	34	678	777.73	735.61	−5.42%
drsp63x6-3	31	494	811.37	726.05	−10.51%
drsp63x6-4	35	700	786.22	721.99	−8.17%
drsp63x6-5	30	515	714.26	665.46	−6.83%
drsp63x6-6	33	560	827.48	847.60	2.43%
drsp63x6-7	30	506	799.50	654.05	−18.19%
drsp63x6-8	28	383	873.49	791.88	−9.34%
drsp63x6-9	34	608	782.46	650.52	−16.86%
Average	32.00	568.89	784.26	714.24	−8.92%
drsp83x8-1	40	582	1114.75	1020.69	−8.44%
drsp83x8-2	38	722	1411.50	1040.39	−26.29%
drsp83x8-3	39	660	1310.40	1173.44	−10.45%
drsp83x8-4	43	740	1050.79	1051.51	0.07%
drsp83x8-5	34	516	1155.35	882.54	−23.61%
drsp83x8-6	39	661	1092.90	1019.06	−6.76%
drsp83x8-7	41	730	1124.13	1019.06	−9.35%
drsp83x8-8	40	726	1315.08	1340.66	1.95%
drsp83x8-9	39	668	1212.02	1052.88	−13.13%
Average	39.22	667.22	1198.55	1066.69	−10.67%

Table 5 shows the results obtained after running our approach for the considered instances. The limited number of vehicles and their limited capacities make it infeasible to visit all the potential customers. Hence, the maximum number of served customers is 23, 35, and 43 for each group of instances, respectively. The total collected fee for serving these customers is also shown. Then, we employ our algorithm to generate two different solutions: our best static solution (OBS) and our best dynamic solution (OBD). The OBS is a solution generated in the period 0 (an initial solution) that cannot be further modified, i.e., the vehicles perform the same static routes regardless of the traffic conditions. Alternatively, the OBD is a solution that is recomputed each 10 time units and, therefore, it adapts to the changing traffic conditions. This recalculation is performed in the same way as the initial solution, but considering only the remaining customers to serve and the cost per edge given by Equation (1). The total collected fee is not different for the OBS and the OBD since the customers selected to be served by the initial solution are respected in all the recalculations performed to obtain the OBD. Only the routes can change in this case. Finally, gaps in Table 5 represent the percentage difference between the costs attained by the OBD and the ones attained by the OBS. A negative gap means that the OBD obtains a lower cost than the OBS. Average gaps are always negative regardless of the instance size. Only 4 out of 27 instances reach a positive gap, with a maximum value of only 2.43% for the instance *drsp63x6-6*. Conversely, the most negative gap reaches a value of −35.61% for the instance *drsp43x4-5*. These results indicate that, in terms of costs, our dynamic approach always outperforms the scenario in which the solution is not adaptable to external changing conditions.

Figure 5 shows a series of box plots displaying the attained costs by the OBS (pink) and the OBD (green). Each box plot depicts the group of instances classified according to their size. Additionally, a crossed circle indicates the mean of each group of data. This figure shows the natural cost increase when the instance size grows. Nevertheless, this increase is made up for the proportional rise in the total collected fee, as Table 5 shows. Furthermore, Figure 5 indicates graphically the cost savings attained after solving this problem with our dynamic approach, instead of employing a static metaheuristic.

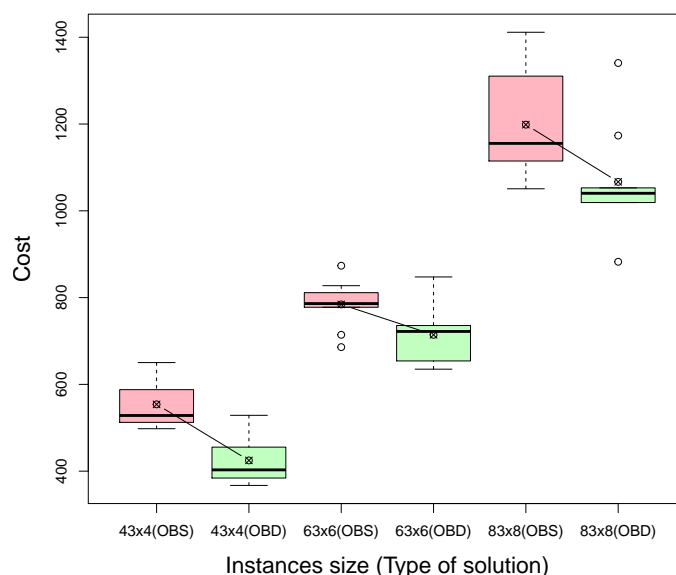
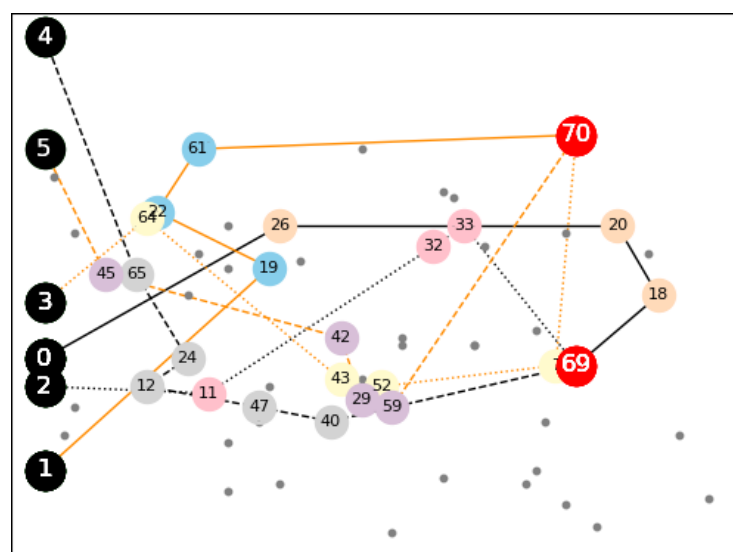
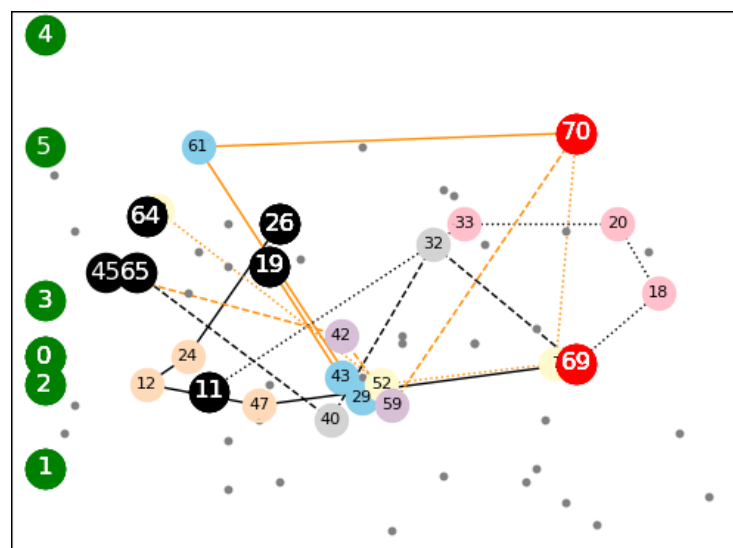


Figure 5. Costs obtained by each type of solution for different instance sizes.

Figure 6 depicts an example of the execution of our algorithm in two consecutive periods for the instance *drsp63x6-2*. Big black nodes indicate the origin points, big red nodes indicate the destination points, medium-sized numbered nodes represent the served customers, and small unnumbered gray points represent the non-served customers. Routes are depicted by lines of different colors and styles. Figure 6a shows the initial best-found solution (BFS), generated in the period $n = 0$. If this set of routes is not further modified, then we obtain the OBS, at a cost of 777.73 (Table 5). Alternatively, Figure 6b shows the BFS in the period $n = 1$ considering dynamic conditions and, therefore, the originally designed routes have changed to eventually obtain the OBD. At this moment, each vehicle has already arrived to the location of its respective first customer. Our algorithm takes these locations as new origin points; hence, the former origins are represented by green nodes in Figure 6b. Notice that routes after the black nodes in this figure are different from those of Figure 6a. For instance, the black dashed route in Figure 6a follows the sequence 4-65-24-12-47-40-69. In the period $n = 1$, the corresponding vehicle has already traveled from node 4 to node 65. Nevertheless, the dynamic traffic conditions makes the original planned route change and the vehicle follows the new sequence 4-65-40-32-69. The total customers selected in the period $n = 0$ remain the same, i.e., the commitment of serving them is respected, although the routes are different now. Given the limited space, the new re-computed routes for the rest of the periods are not displayed; however, this process is repeated until every vehicle has arrived at nodes 69 and 70, respectively. All these successive recalculations lead the attainment of the OBD, at a cost of 735.61 (Table 5).

(a) Period $n = 0$.(b) Period $n = 1$.**Figure 6.** Example of the BFS for the instance *drsp63x6-2* in the periods $n = 0$ and $n = 1$.

7. Conclusions and Future Research

This paper has discussed how the combination of cloud with fog and edge computing can influence modern transportation systems, especially in the context of smart cities, where open data initiatives are growing. These open repositories can be employed to provide real-time information to users of public and private transportation systems. When processed by machine learning methods and optimization algorithms, this information can become in valuable knowledge that empowers citizens and support intelligent mobility decisions.

The paper also discussed the necessity for developing agile algorithms, capable not only of providing high-quality solutions to large-scale vehicle routing and other transportation-related problems, but also to do so in real-time and every few minutes, as new data are gathered via IoT and open repositories. This agile optimization algorithms are based on the parallelization of biased-randomized algorithms, which rely on introducing an oriented random behavior into an already fast constructive heuristic.

A numerical case study regarding ride-sharing mobility contributes to illustrate these concepts. In this case study, a more traditional static scenario is compared against a dynamic one. In the former, the transportation system is optimized just at the beginning,

while in the latter new data are employed to re-optimize the system every now and then. The computational experiments show the benefits of our proposed dynamic approach, which clearly outperform the traditional one in terms of costs. Furthermore, our experiments include instances with multiple sizes; hence, our algorithm has been proven to be flexible and scalable, i.e., it can be easily adapted to cope with even bigger instances. Nevertheless, it is worth mentioning that we have not included the occurrence of events such as new requests, cancellations, or destination modifications. These events are present in real-life situations and, therefore, they can be included in future work.

As additional future research lines, the following are highlighted: (i) to employ our agile optimization methodology in a scenario using data directly obtained from an open-data repository in the city of Barcelona; (ii) to extend the proposed methodology into a simheuristic algorithm, capable of considering random travel times, random rewards, as well as other stochastic components; and (iii) to consider a different case study based on a car-sharing problem, which complements the ride-sharing example provided in this paper.

Author Contributions: Conceptualization, F.X. and A.A.J.; methodology, P.J.C., L.d.C.M. and R.D.T.; validation, M.P., P.J.C. and R.D.T.; formal analysis, P.J.C., M.P. and L.d.C.M.; writing—original draft preparation, R.D.T., L.d.C.M. and M.P.; writing—review and editing, P.J.C. and A.A.J. 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: Not applicable.

Acknowledgments: This work was partially supported by the Spanish Ministry of Science (PID2019-111100RB-C21/AEI/10.13039/501100011033, RED2018-102642-T), and the Erasmus+ program (2019-I-ES01-KA103-062602).

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

References

1. Gruler, A.; Quintero-Araújo, C.L.; Calvet, L.; Juan, A.A. Waste collection under uncertainty: A simheuristic based on variable neighbourhood search. *Eur. J. Ind. Eng.* **2017**, *11*, 228–255.
2. Martins, L.d.C.; de la Torre, R.; Corlu, C.G.; Juan, A.A.; Masmoudi, M.A. Optimizing ride-sharing operations in smart sustainable cities: Challenges and the need for agile algorithms. *Comput. Ind. Eng.* **2021**, *153*, 107080.
3. Beneicke, J.; Juan, A.A.; Xhafa, F.; Lopez-Lopez, D.; Freixes, A. Empowering citizens' cognition and decision making in smart sustainable cities. *IEEE Consum. Electron. Mag.* **2019**, *9*, 102–108.
4. Aazam, M.; Huh, E.N. Fog computing and smart gateway based communication for cloud of things. In Proceedings of the 2014 International Conference on Future Internet of Things and Cloud, Barcelona, Spain, 27–29 August 2014; pp. 464–470.
5. Zhu, Y.; Zhu, X.; Zhu, S.; Guo, S. Intelligent transportation system based on Internet of Things. In Proceedings of the World Automation Congress 2012, Puerto Vallarta, Mexico, 24–28 June 2012; pp. 1–3.
6. Bojan, T.M.; Kumar, U.R.; Bojan, V.M. An internet of things based intelligent transportation system. In Proceedings of the 2014 IEEE International Conference on Vehicular Electronics and Safety, Hyderabad, India, 16–17 December 2014; pp. 174–179.
7. Ahlgren, B.; Hidell, M.; Ngai, E.C.H. Internet of things for smart cities: Interoperability and open data. *IEEE Internet Comput.* **2016**, *20*, 52–56.
8. Tang, C.; Xia, S.; Zhu, C.; Wei, X. Phase timing optimization for smart traffic control based on fog computing. *IEEE Access* **2019**, *7*, 84217–84228.
9. Aazam, M.; Huh, E.N. Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT. In Proceedings of the 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, Gwangju, Korea, 24–27 March 2015; pp. 687–694.
10. Jiménez-Meza, A.; Arámburo-Lizárraga, J.; de la Fuente, E. Framework for estimating travel time, distance, speed, and street segment level of service (los), based on GPS data. *Procedia Technol.* **2013**, *7*, 61–70.
11. Agatz, N.; Erera, A.L.; Savelsbergh, M.W.; Wang, X. Dynamic ride-sharing: A simulation study in metro Atlanta. *Procedia Soc. Behav. Sci.* **2011**, *17*, 532–550.
12. Grasas, A.; Juan, A.A.; Faulin, J.; de Armas, J.; Ramalhinho, H. Biased randomization of heuristics using skewed probability distributions: A survey and some applications. *Comput. Ind. Eng.* **2017**, *110*, 216–228.

13. Rodríguez-Bolívar, M.P. *Transforming City Governments for Successful Smart Cities*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 1–185.
14. Janssen, M.; Charalabidis, Y.; Zuiderwijk, A. Benefits, Adoption Barriers and Myths of Open Data and Open Government. *Inf. Syst. Manag.* **2012**, *29*, 258–268.
15. Granickas, K. *Understanding the Impact of Releasing and Re-Using Open*; European Public Sector Information Platform: Brussels, Belgium, 2013; pp. 1–29.
16. Solar, M.; Concha, G.; Meijueiro, L. A model to assess open government data in public agencies. In Proceedings of the International Conference on Electronic Government, Kristiansand, Norway, 3–6 September 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 210–221.
17. Ojo, A.; Curry, E.; Zeleti, F.A. A tale of open data innovations in five smart cities. In Proceedings of the Annual Hawaii International Conference on System Sciences, Kauai, HI, USA, 5–8 January 2015; Volume 2015, pp. 2326–2335.
18. Abella, A.; Ortiz-de Urbina-Criado, M.; De-Pablos-Heredero, C. A model for the analysis of data-driven innovation and value generation in smart cities' ecosystems. *Cities* **2017**, *64*, 47–53.
19. Vlahogianni, E.I. Computational intelligence and optimization for transportation big data: Challenges and opportunities. In *Engineering and Applied Sciences Optimization*; Springer: Cham, Switzerland, 2015; pp. 107–128.
20. Saharan, S.; Bawa, S.; Kumar, N. Dynamic pricing techniques for Intelligent Transportation System in smart cities: A systematic review. *Comput. Commun.* **2020**, *150*, 603–625.
21. Satunin, S.; Babkin, E. A multi-agent approach to intelligent transportation systems modeling with combinatorial auctions. *Expert Syst. Appl.* **2014**, *41*, 6622–6633.
22. Shah, N.; Kumar, S.; Bastani, F.; Yen, I.L. Optimization models for assessing the peak capacity utilization of intelligent transportation systems. *Eur. J. Oper. Res.* **2012**, *216*, 239–251.
23. Taniguchi, E.; Shimamoto, H. Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times. *Transp. Res. Part C Emerg. Technol.* **2004**, *12*, 235–250.
24. Marques, M.C.; Neves-Silva, R. Traffic simulation for intelligent transportation systems development. In Proceedings of the 2005 IEEE Intelligent Transportation Systems, Vienna, Austria, 16 September 2005; pp. 320–325.
25. Ramos, A.L.; Ferreira, J.V.; Barceló, J. Modeling & simulation for intelligent transportation systems. *Int. J. Model. Optim.* **2012**, *2*, 274.
26. Fernández-Isabel, A.; Fuentes-Fernández, R. Analysis of intelligent transportation systems using model-driven simulations. *Sensors* **2015**, *15*, 14116–14141.
27. Kornhauser, A.L.; Mottola, P.; Stephenson, B. Transportation efficiency and the feasibility of dynamic ride sharing. In Proceedings of the 56th Annual Meeting of the Transportation Research Board, Washington, DC, USA, 24–28 January 1977; Transportation Research Record Serial.
28. Lokhandwala, M.; Cai, H. Dynamic ride sharing using traditional taxis and shared autonomous taxis: A case study of NYC. *Transp. Res. Part C Emerg. Technol.* **2018**, *97*, 45–60.
29. Fagnant, D.J.; Kockelman, K.M. Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin, Texas. *Transportation* **2018**, *45*, 143–158.
30. Haghighat, A.K.; Ravichandra-Mouli, V.; Chakraborty, P.; Esfandiari, Y.; Arabi, S.; Sharma, A. Applications of deep learning in intelligent transportation systems. *J. Big Data Anal. Transp.* **2020**, *2*, 115–145.
31. Fang, S.H.; Liao, H.H.; Fei, Y.X.; Chen, K.H.; Huang, J.W.; Lu, Y.D.; Tsao, Y. Transportation modes classification using sensors on smartphones. *Sensors* **2016**, *16*, 1324.
32. Said, A.M.; Abd-Elrahman, E.; Afifi, H. A comparative study on machine learning algorithms for green context-aware intelligent transportation systems. In Proceedings of the 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA 2017), Ras Al Khaimah, United Arab Emirates, 21–23 November 2017; Volume 2018, pp. 1–5.
33. Nguyen, H.; Kieu, L.M.; Wen, T.; Cai, C. Deep learning methods in transportation domain: A review. *IET Intell. Transp. Syst.* **2018**, *12*, 998–1004.
34. Li, C.; Xu, P. Application on traffic flow prediction of machine learning in intelligent transportation. *Neural Comput. Appl.* **2021**, *33*, 613–624.
35. Karami, Z.; Kashef, R. Smart transportation planning: Data, models, and algorithms. *Transp. Eng.* **2020**, *2*, 100013.
36. Boukerche, A.; Wang, J. Machine Learning-based traffic prediction models for Intelligent Transportation Systems. *Comput. Netw.* **2020**, *181*, 107530.
37. Chen, C.; Liu, B.; Wan, S.; Qiao, P.; Pei, Q. An edge traffic flow detection scheme based on deep learning in an intelligent transportation system. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 1840–1852.
38. Jahangiri, A.; Rakha, H.A. Transportation Mode Recognition Using Mobile Phone Sensor Data. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2406–2417.
39. Omrani, H. Predicting travel mode of individuals by machine learning. *Transp. Res. Procedia* **2015**, *10*, 840–849.
40. Gal, A.; Mandelbaum, A.; Schnitzler, F.; Senderovich, A.; Weidlich, M. Traveling time prediction in scheduled transportation with journey segments. *Inf. Syst.* **2017**, *64*, 266–280.
41. Talbi, E.G. *Metaheuristics: From Design to Implementation*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 74.

42. Juan, A.A.; Faulin, J.; Grasman, S.E.; Rabe, M.; Figueira, G. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Oper. Res. Perspect.* **2015**, *2*, 62–72.
43. Chica, M.; Juan, A.A.; Christopher, B.; Oscar, C.; W. David, K. Why simheuristics? Benefits, limitations, and best practices when combining metaheuristics with simulation. *Stat. Oper. Res. Trans.* **2020**, *44*, 311–334.
44. Ferone, D.; Gruler, A.; Festa, P.; Juan, A.A. Enhancing and extending the classical GRASP framework with biased randomisation and simulation. *J. Oper. Res. Soc.* **2019**, *70*, 1362–1375.
45. Martins, L.d.C.; Hirsch, P.; Juan, A.A. Agile optimization of a two-echelon vehicle routing problem with pickup and delivery. *Int. Trans. Oper. Res.* **2021**, *28*, 201–221.
46. Martins, L.d.C.; Tarchi, D.; Juan, A.A.; Fusco, A. Agile optimization for a real-time facility location problem in Internet of Vehicles networks. *Networks* **2021**, doi:10.1002/net.22067.
47. Kumar, K.V.; Kumar, K.D.; Poluru, R.K.; Basha, S.M.; Reddy, M.P.K. Internet of things and fog computing applications in intelligent transportation systems. In *Architecture and Security Issues in Fog Computing Applications*; IGI Global: London, UK, 2019; pp. 131–150.
48. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog computing and its role in the internet of things. In Proceedings of the MCC'12—Proceedings of the 1st ACM Mobile Cloud Computing Workshop, Helsinki, Finland, 13–17 August 2012; pp. 13–15.
49. Peter, N. FOG Computing and Its Real Time Applications. *Int. J. Emerg. Technol. Adv. Eng. FOG Comput. Its Real Time Appl.* **2015**, *5*, 266.
50. Bierzynski, K.; Escobar, A.; Eberl, M. Cloud, fog and edge: Cooperation for the future? In Proceedings of the 2017 2nd International Conference on Fog and Mobile Edge Computing (FMEC 2017), Valencia, Spain, 8–11 May 2017; pp. 62–67.
51. Chen, S.; Wen, H.; Wu, J.; Lei, W.; Hou, W.; Liu, W.; Xu, A.; Jiang, Y. Internet of Things Based Smart Grids Supported by Intelligent Edge Computing. *IEEE Access* **2019**, *7*, 74089–74102.
52. Xhafa, F.; Aly, A.; Juan, A.A. Allocation of applications to Fog resources via semantic clustering techniques: With scenarios from intelligent transportation systems. *Computing* **2021**, *103*, 361–378.
53. Gohar, A.; Nencioni, G. The Role of 5G Technologies in a Smart City: The Case for Intelligent Transportation System. *Sustainability* **2021**, *13*, 5188.
54. Yan, G.; Wen, D.; Olariu, S.; Weigle, M.C. Security challenges in vehicular cloud computing. *IEEE Trans. Intell. Transp. Syst.* **2012**, *14*, 284–294.
55. Wang, K.; Yin, H.; Quan, W.; Min, G. Enabling Collaborative Edge Computing for Software Defined Vehicular Networks. *IEEE Netw.* **2018**, *32*, 112–117.
56. He, W.; Yan, G.; Xu, L.D. Developing vehicular data cloud services in the IoT environment. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1587–1595.
57. Dobre, C.; Xhafa, F. Intelligent services for Big data science. *Future Gener. Comput. Syst.* **2014**, *37*, 267–281.
58. Xiao, Y.; Zhu, C. Vehicular fog computing: Vision and challenges. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops 2017), Kona, HI, USA, 13–17 March 2017; pp. 6–9.
59. Tanganelli, G.; Vallati, C.; Mingozzi, E. A fog-based distributed look-up service for intelligent transportation systems. In Proceedings of the 18th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2017—Conference, Macau, China, 12–15 June 2017; pp. 4–9.
60. Badidi, E.; Mahrez, Z.; Sabir, E. Fog computing for smart cities' big data management and analytics: A review. *Future Internet* **2020**, *12*, 1–29.
61. Zhang, J.; Wang, F.Y.; Wang, K.; Lin, W.H.; Xu, X.; Chen, C. Data-driven intelligent transportation systems: A survey. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 1624–1639.
62. Minh, Q.T.; Kamioka, E.; Yamada, S. CFC-ITS: Context-Aware Fog Computing for Intelligent Transportation Systems. *IT Prof.* **2018**, *20*, 35–44.
63. Darwish, T.S.; Abu Bakar, K. Fog Based Intelligent Transportation Big Data Analytics in The Internet of Vehicles Environment: Motivations, Architecture, Challenges, and Critical Issues. *IEEE Access* **2018**, *6*, 15679–15701.
64. Raza, S.; Wang, S.; Ahmed, M.; Anwar, M.R. A survey on vehicular edge computing: Architecture, applications, technical issues, and future directions. *Wirel. Commun. Mob. Comput.* **2019**, *2019*, doi:10.1155/2019/3159762.
65. Brennand, C.A.; Filho, G.P.; Maia, G.; Cunha, F.; Guidoni, D.L.; Villas, L.A. Towards a fog-enabled intelligent transportation system to reduce traffic jam. *Sensors* **2019**, *19*, 3916.
66. Chiang, M.; Zhang, T. Fog and IoT: An overview of research opportunities. *IEEE Internet Things J.* **2016**, *3*, 854–864.
67. Tufail, A.; Namoun, A.; Alrehaili, A.; Ali, A. A Survey on 5G Enabled Multi-Access Edge Computing for Smart Cities: Issues and Future Prospects. *Int. J. Comput. Sci. Netw. Secur.* **2021**, *21*, 107–118.
68. Cheng, X.; Yang, L.; Shen, X. D2D for intelligent transportation systems: A feasibility study. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1784–1793.
69. Camacho, F.; Cárdenas, C.; Muñoz, D. Emerging technologies and research challenges for intelligent transportation systems: 5G, HetNets, and SDN. *Int. J. Interact. Des. Manuf.* **2018**, *12*, 327–335.
70. Lee, Y.; Jeong, S.; Masood, A.; Park, L.; Dao, N.N.; Cho, S. Trustful Resource Management for Service Allocation in Fog-Enabled Intelligent Transportation Systems. *IEEE Access* **2020**, *8*, 147313–147322.

71. Raza, N.; Jabbar, S.; Han, J.; Han, K. Social Vehicle-To-Everything (V2X) communication model for Intelligent Transportation Systems based on 5G scenario. In Proceedings of the International Conference on Future Networks and Distributed Systems, Amman, Jordan, 26–27 June 2018; ACM International Conference Proceeding Series.
72. Muthuramalingam, S.; Bharathi, A.; Gayathri, N.; Sathiyaraj, R.; Balamurugan, B. IoT based intelligent transportation system (IoT-ITS) for global perspective: A case study. In *Internet of Things and Big Data Analytics for Smart Generation*; Springer: Cham, Switzerland, 2019; pp. 279–300.
73. Dai, X.; Ma, T. IoT perception and public transportation network optimization based on big data algorithms. *Pers. Ubiquitous Comput.* **2021**, 1–12, doi:10.1007/s00779-021-01634-0.
74. Swarnamugi, M.; Chinnaiyan, R. Modelling and Reasoning Techniques for Context-Aware Computing in Intelligent Transportation System. *arXiv* **2021**, arXiv:2107.14374.
75. Mohandu, A.; Kubendiran, M. Survey on Big Data Techniques in Intelligent Transportation System (ITS). *Mater. Today Proc.* **2021**, 47, 8–17.
76. Calabrese, F.; Diao, M.; Di Lorenzo, G.; Ferreira, J.; Ratti, C. Understanding individual mobility patterns from urban sensing data: A mobile phone trace example. *Transp. Res. Part C Emerg. Technol.* **2013**, 26, 301–313.
77. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of things for smart cities. *IEEE Internet Things J.* **2014**, 1, 22–32.
78. Mahdavinejad, M.S.; Rezvan, M.; Barekatain, M.; Adibi, P.; Barnaghi, P.; Sheth, A.P. Machine learning for internet of things data analysis: A survey. *Digit. Commun. Netw.* **2018**, 4, 161–175.
79. Graser, A. MovingPandas: Efficient structures for movement data in Python. *GIForum* **2019**, 1, 54–68.
80. Pappalardo, L.; Simini, F.; Barlacchi, G.; Pellungrini, R. scikit-mobility: A Python library for the analysis, generation and risk assessment of mobility data. *arXiv* **2019**, arXiv:1907.07062.
81. Bao, W.; Liu, X.Y. Spatial Influence-aware Reinforcement Learning for Intelligent Transportation System. *arXiv* **2019**, arXiv:1912.06880.
82. Teng, H.; Liu, Y.; Liu, A.; Xiong, N.N.; Cai, Z.; Wang, T.; Liu, X. A novel code data dissemination scheme for Internet of Things through mobile vehicle of smart cities. *Future Gener. Comput. Syst.* **2019**, 94, 351–367.
83. Cao, H.; Wachowicz, M.; Cha, S. Developing an edge computing platform for real-time descriptive analytics. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 4546–4554.
84. Panadero, J.; Freixes, A.; Mozos, J.M.; Juan, A.A. Agile Optimization for Routing Unmanned Aerial Vehicles under Uncertainty. In Proceedings of the XIII Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2017), Granada, Spain, 23–26 October 2018; pp. 635–640.
85. Quintero-Araujo, C.L.; Caballero-Villalobos, J.P.; Juan, A.A.; Montoya-Torres, J.R. A biased-randomized metaheuristic for the capacitated location routing problem. *Int. Trans. Oper. Res.* **2017**, 24, 1079–1098.
86. Adi, E.; Anwar, A.; Baig, Z.; Zeadally, S. Machine learning and data analytics for the IoT. *Neural Comput. Appl.* **2020**, 32, 16205–16233.
87. Wei, J.; Long, K.; Gu, J.; Ju, Q.; Zhu, P. Optimizing bus line based on metro-bus integration. *Sustainability* **2020**, 12, 1493.
88. Bistaffa, F.; Blum, C.; Cerquides, J.; Farinelli, A.; Rodríguez-Aguilar, J.A. A computational approach to quantify the benefits of ridesharing for policy makers and travellers. *IEEE Trans. Intell. Transp. Syst.* **2019**, 22, 119–130.
89. Fikar, C.; Juan, A.A.; Martinez, E.; Hirsch, P. A discrete-event driven metaheuristic for dynamic home service routing with synchronised trip sharing. *Eur. J. Ind. Eng.* **2016**, 10, 323–340.
90. Panadero, J.; Juan, A.A.; Bayliss, C.; Currie, C. Maximising reward from a team of surveillance drones: A simheuristic approach to the stochastic team orienteering problem. *Eur. J. Ind. Eng.* **2020**, 14, 485–516.
91. Martí, R. Multi-start methods. In *Handbook of Metaheuristics*; Springer: Boston, MA, USA, 2003; pp. 355–368.