

Article

An Artificial Plant Community with a Random-Pairwise Single-Elimination Tournament System for Conflict-Free Human–Machine Collaborative Manufacturing in Industry 5.0

Zhengying Cai , Xinfei Dou , Cancan He , Huiyan Deng  and Zhen Liu 

Hubei Province Engineering Technology Research Center for Construction Quality Testing Equipment, College of Computer and Information Technology, China Three Gorges University, Yichang 443002, China

* Correspondence: master_cai@163.com

Abstract

Human–machine collaborative manufacturing plays an important role in emerging Industry 5.0 and smart manufacturing. However, addressing the conflict-free human–machine collaborative manufacturing problem (CHMCMP) is extremely challenging because the cooperation and conflict between humans and machines are closely intertwined. This article examines the CHMCMP within the context of integrating the flexible job-shop scheduling problem (FJSP) and the flow-shop scheduling problem (FSP). Firstly, the CHMCMP was modeled as a job-flow-shop scheduling problem (JFSP), where machine processing is an FJSP and human operation is an FSP. Our goal is to complete all manufacturing jobs while pursuing multi-objective optimization, i.e., high manufacturing performance, conflict-free human–machine collaboration, and low no-load energy consumption. Secondly, an improved artificial plant community (APC) algorithm was developed to solve the NP-hard problem. A random-pairwise single-elimination tournament system is introduced for elite selection, with a time complexity of $O(S)$ linearly correlated with the population size (S), superior to the sorting-based elite selection used by most evolutionary algorithms with polynomial time complexity, i.e., $O(S^3)$ of the genetic algorithm (GA) and $O(S^2)$ of the non-dominated sorting genetic algorithm-II (NSGA-II). Thirdly, a medium-scale benchmark dataset was exploited according to a human–machine collaborative manufacturing scenario. The Gantt charts of machine processing and human operating reveal that the FJSP and the FSP are entangled and are interdependent on each other in the CHMCMP, and solving FJSP and FSP separately cannot eliminate the conflict between the two. Compared with other state-of-the-art algorithms, the APC algorithm improves the makespan by up to 11.38%, the total transfer time of humans by up to 14.09%, and the no-loaded processing energy consumption by up to 12.62% with conflict avoidance.



Academic Editor: Panagiotis Kyratsis

Received: 15 January 2026

Revised: 6 February 2026

Accepted: 6 February 2026

Published: 10 February 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: human–machine collaborative manufacturing (HMCM); Industry 5.0; smart manufacturing; flexible job-shop scheduling problem (FJSP); flow-shop scheduling problem (FSP); multi-objective optimization problem (MOP); conflict-free scheduling; artificial plant community (APC); metaheuristic algorithm

1. Introduction

Industry 5.0 and smart manufacturing represent an advanced, intelligent, automated manufacturing mode that is an intelligent human–machine system (HMS) with autonomous machine processing and robot transporting from raw materials to final products [1,2].

Among them, the human–machine collaborative manufacturing (HMCM) is particularly advantageous for distribution manufacturing systems handling a wide variety of small items and experiencing significant demand variability [3,4], as shown in Figure 1. Its implementation entails substantial human–machine collaboration, often tens of humans and machines or robots [5,6]. Therefore, a key challenge in the fields of Industry 5.0 and smart manufacturing is to optimize human–machine collaborative manufacturing [7], which is not an easy task and requires the integration and collaboration of humans, machines, and artificial intelligence (AI).

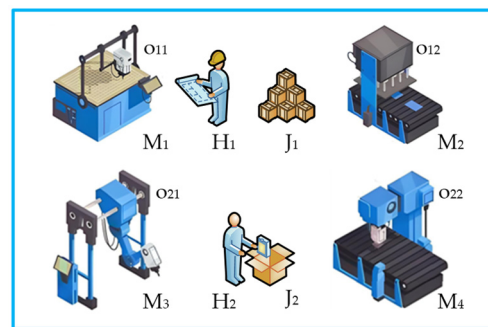


Figure 1. An example of human–machine collaborative manufacturing.

Rahman et al. [1] reviewed 187 references and analyzed the human–machine interfaces (HMIs) in enhancing additive processes. Vyavahare et al. [7] surveyed 211 references and synthesized literature on human–machine collaborative additive manufacturing (HMC-AM). The solution to the human–machine collaborative manufacturing problem (HMCMP) must be harmoniously aligned with the operations of manufacturing facilities and material resources [8]. The interdependence between human–machine scheduling and job scheduling is such that these processes must be executed in a synchronized manner. The concept of concurrent scheduling for humans and machines is a well-explored domain within job shop settings. The common solution is to model the problem as a job-shop scheduling problem (JSP) [9], which aims to minimize the maximum completion time across all jobs (CMAX), under the constraint that each job consists of a sequence of operations that must be performed on designated stations.

Another solution is the flexible job-shop scheduling problem (FJSP) [10,11], which is an extension of classical JSP, in which each process is allowed to be processed on any one of a set of available machines. Some scholars have also modeled it as the flow-shop scheduling problem (FSP) [12,13], where there is a set of machine tools with different functions, and the workpieces to be machined include multiple processes. In addition, many artificial intelligence algorithms have also been introduced by scholars to help solve the HMCMP, such as the non-dominated sorting genetic algorithm-II (NSGA-II) [8], ant colony optimization (ACO) [9], A*-inspired greedy search [10], particle swarm optimization (PSO) [11], grey wolf optimization (GWO) [11], proximal policy optimization (PPO) [12], Q-learning [12], Q-learning guided dual-population genetic algorithm (QGGA) [13], deep reinforcement learning (DRL) [14], federated learning (FL) [15], etc.

The existing research work mainly focuses on human–machine collaboration and pays little attention to human–machine conflicts, such as decision conflicts [16] or safety conflicts [17]. If the human–machine conflict cannot be well managed, it will naturally lead to the failure of human–machine cooperation, which is the research motivation of this article. Given the NP-hard nature of the conflict-free human–machine collaborative manufacturing problem (CHMCMP), developing a lightweight and efficient solving algorithm, i.e., artificial plant community (APC) algorithm [18], is also necessary and challenging.

Most evolutionary algorithms use sorting-based elite selection, and the time complexity shows a polynomial-level nonlinear correlation with the population size, resulting in a sharp increase in solution time as the population size grows.

This paper presents three primary contributions.

Firstly, unlike existing studies that separate FJSP and FSP, our work introduces a job-flow-shop scheduling problem (JFSP) to model the CHMCMP, integrating the FJSP of machine processing and the FSP of human operating. Our goal is to complete all manufacturing jobs while realizing multi-objective optimization (MOP), i.e., high manufacturing performance, conflict-free human-machine collaboration, and low no-load energy consumption. The JFSP considers the maximum completion time across all jobs, the no-load energy consumption, and the human-machine conflict time constraint.

Secondly, different from existing works that use sorting algorithms for elite selection, the proposed APC algorithm utilizes a random-pairwise single-elimination tournament system for elite selection, with a time complexity of $O(S)$ linearly correlated with the population size (S), better than $O(S^3)$ of GA [13] and $O(S^2)$ of NSGA-II [8]. The APC-based scheduling architecture includes four main steps, i.e., job analysis and manufacturing system evaluation, CHMCMP initialization, APC-based iterative search for the JFSP, and implementation of scheduling and manufacturing based on the optimal solution. It outperforms many existing benchmark methods in solving such NP-hard problems for Industry 5.0 and smart manufacturing.

Thirdly, a medium-scale benchmark test is developed that incorporates the maximum completion time, the conflict time, and the no-load energy consumption. Our work addresses the current limitation of only having small manufacturing instances available for study. The Gantt charts of machine processing FJSP and human operating FSP are provided, revealing the complex dependency relationships in human-machine collaborative manufacturing. Compared with other state-of-the-art algorithms, the APC algorithm improves the makespan by up to 11.38%, the total transfer time of humans by up to 14.09%, and the no-loaded processing energy consumption by up to 12.62% with conflict avoidance.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 illustrates the CHMCMP. Section 4 designs an APC-based scheduling architecture. Section 5 covers a series of benchmark experiments and comparative results, and Section 6 concludes this paper.

2. Related Work

Industry 4.0 can be likened to the Fourth Industrial Revolution [3]. To achieve automation and networking of Industry 4.0, more and more sensors, big data, and AI technologies are constantly penetrating into the human-machine collaborative manufacturing. Konstantinidis et al. [19] reviewed 93 references and brought out research gaps of human-machine interaction in Industry 4.0. In the context of Manufacturing 4.0, Habib et al. [20] proposed an experimental study to evaluate the impact of teamwork, human-machine cooperation, performance, and workload of the human operator in an intelligent manufacturing system made up of mobile robots and autonomous shuttles.

Industry 5.0 is not a chronological continuation or replacement of the existing Industry 4.0 paradigm, as it is the result of forward-looking work [21]. Pizon et al. [21] surveyed 60 references and carried out a human-machine collaborative manufacturing problem from the perspective of Industry 5.0. It describes the impact that the IoT will have on production organizations, thanks to the new interactions between humans and machines, as well as the new wave of digital applications in manufacturing. In Industry 5.0, the boundaries between different types of industrial workers are blurred [22]. The human-centric approach [15] in industry is not about starting from emerging technologies and examining their potential to

improve efficiency, but rather placing humans' core demands and interests at the core of the production process. Simeone et al. [23] explored the integration of neurodiverse individuals in the workforce through human-machine collaboration and reciprocal learning (RL), and used fuzzy logic inference to develop a cognitive load assessment procedure.

FJSP is a frequently used solution to address the human-machine collaborative manufacturing problem and is more difficult than traditional JSP because it introduces another decision content besides sorting, which is the job path [10,11]. Tarek et al. [11] noticed the hybrid human-machine collaborative flexible job shop scheduling problem (HHCFJSP) with random job arrivals, and put forward an A*-inspired greedy search strategy to deal with sudden job insertion demands during the scheduling process. Joo et al. [24] introduced a DRL-based approach in the process of job assignment to consider the unobservable characteristics of human operators and the stochastic nature of manufacturing systems. Zhou et al. [25] insisted that the dynamic flexible job shop scheduling problem (DFJSP) is challenging, due to its stochastic task arrivals, heterogeneous deadlines, and varied task types, and developed a genetic-algorithm-based hyperparameter optimization with proximal policy optimization (GA-HPO PPO) algorithm to enhance learning efficiency and scheduling performance. Manafi et al. [26] deal with a dual-resource flexible job shop scheduling (DRFJSS) problem, in which each operation requires two resources to be processed, i.e., a reconfigurable machine tool and a worker. It formulated the DRFJSS problem as a mixed-integer linear programming (MILP) model to minimize the makespan, and developed a self-learning whale optimization algorithm (SLWOA) to deal efficiently with such a difficult problem with medium-sized in-stances.

FSP is another solution to address the human-machine collaborative manufacturing problem, where each process is machined on one machine tool, and the machining routes for all workpieces are the same. There is a sequence constraint between each workpiece process. Wu et al. [12] considered that the hybrid flow-shop scheduling problem (HFSP) with energy efficiency and human-machine collaboration is significantly challenging and NP-hard. The study introduced an energy-efficient HFSP with human-machine collaboration (EHFSP-HMC) model, aiming to minimize both makespan and total energy consumption. It exploited a multi-objective evolutionary co-learning framework (MOECLF) that combines two proximal policy optimization (PPO) agents and a Q-learning agent for multi-agent hyper-heuristic search. Zuo et al. [13] proposed a Q-learning guided dual-population genetic algorithm and a state-dependent action adaptation mechanism to dynamically choose actions for distributed permutation flow-shop scheduling problem.

From recent research work, it can be seen that Industry 5.0 and human-machine collaborative manufacturing are important development directions for the future, and this problem is extremely challenging, attracting a large number of scholars to participate. Various artificial intelligence algorithms are constantly emerging to help us solve such challenges, such as NSGA-II [8], ACO [9], A*-inspired greedy search [10], PSO [11,17], GWO [11], PPO [12,25], GA [13,25], whale optimization algorithm (WOA) [26], APC [18,27], Q-learning [12,13], DRL [14,16,24], federated learning [15], machine learning [28], multifeature fusion multikernel learning [29], hierarchical aggregation graph neural network [30], etc.

However, existing research still has the following research gaps.

Firstly, few scholars have paid attention to the conflict-free human-machine collaborative manufacturing problem, but solving FJSP and FSP separately cannot eliminate the conflict between the two. Pan et al. [31] insisted that the effectiveness of human-machine teams is affected by conflicts. The cooperation and conflict between humans and machines are a contradictory unity, and conflict-free scheduling can greatly reduce the incidence of workplace accidents.

Secondly, there is a lack of efficient and lightweight solutions to solve such NP-hard problems. Due to the large number of IoT, sensor, and other chips used in Industry 5.0 and smart manufacturing, unlike traditional desktop computers and high-performance servers, there are limitations in computing power, storage resources, power consumption, bandwidth, etc. The sorting-based elite selection used by most evolutionary algorithms shows polynomial nonlinear correlation between the time complexity and the population size, which is not suitable for deployment on industrial edge computing platforms.

Thirdly, there is a lack of a medium-sized benchmark dataset for comprehensive testing of the conflict-free human-machine collaborative manufacturing problem.

3. Conflict-Free Human-Machine Collaborative Manufacturing Problem

Since solving FJSP and FSP separately cannot eliminate the conflict between the two, this chapter introduces a job-flow-shop scheduling problem with multi-objective optimization to jointly solve FJSP and FSP in human-machine collaborative manufacturing.

3.1. Basic Elements

In the real manufacturing scenario of Figure 1, a human-machine collaborative manufacturing environment integrates four basic elements, i.e., a group of humans, a collection of machines, and a set of jobs for part operations, where each job contains a series of operations.

Corresponding to the real manufacturing scenario shown in Figure 1, the CHMCMP is modeled as a job-flow-shop scheduling problem with nonlinear constraints, as shown in Figure 2. In Figure 2, human H_1 loads part J_1O_{11} on machine M_1 for processing, and then unloads it and transfers it to machine M_2 for the next processing. Similarly, human H_2 loads part J_2O_{21} on machine M_3 for processing, and then unloads it and transfers it to machine M_4 for the next processing. In this scenario, machines cannot do without human collaboration, but improper scheduling can also lead to human-machine conflicts.

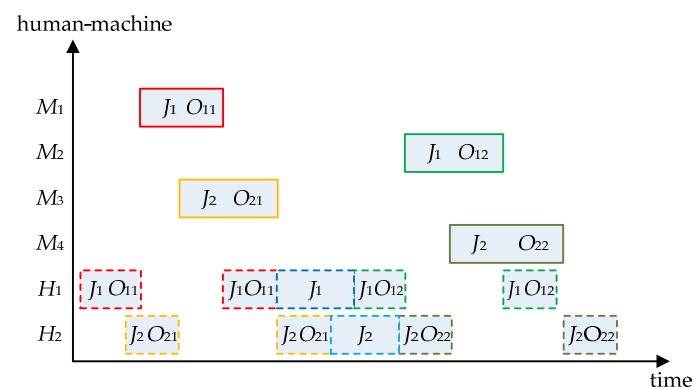


Figure 2. A job-flow-shop scheduling problem. Solid rectangles of different colors are used for processing on different machines, dashed rectangles of different colors are used for operations by different humans, and the same color is used for the same operation by humans on the same machine.

Definition 1. Conflict-free human-machine collaborative manufacturing refers to the mutual perception of the situation between humans and machines, through complementarity and assistance in the same production site, to complete all jobs while avoiding conflicts and achieving ultra flexible manufacturing.

Firstly, in a human-machine collaborative manufacturing environment, there is a group of identical humans, $H = \{H_1, H_2, \dots, H_h, \dots, H_{hmax}\}$, who are tasked with the loading, unloading, and transportation of parts between machines. Humans undertake two types of transfers: loaded transfers and empty transfers. A loaded transfer involves

the human transporting a part from the output buffer of one machine to the input buffer of another, to collect a part ready for transfer from machine m to machine m' , with the condition that $m \neq m'$. Conversely, during an empty transfer, the human transfers from machine m in an idle state, without a part. The transfer time between any two machines m and m' is denoted by $t_{m,m'}^h$, and it independent of the job and dependent on the operation rates of humans and the distances between machines involved.

The transfer time $t_{m,m'}^h$ of human H_h on the distance of $d_{m,m'}$ can be calculated as follows, where $d_{m,m'}$ and r_h are the distance between machines and the operation/transfer rate of the human involved, respectively:

$$t_{m,m'}^h = d_{m,m'} / r_h. \quad (1)$$

Secondly, there is a collection of machines, $M = \{M_1, M_2, \dots, M_m, M_{m'}, \dots, M_{mmax}\}$. Machines can process jobs or parts and are equipped with input and output buffers that temporarily hold parts before and after their processing, also serving as loading and unloading locations for the humans. A part to be processed requires human assistance to be loaded onto a machine for processing; similarly, a processed part requires human assistance to be unloaded from the machine for transfer. As parts proceed through various machines for different operations, this generates a series of loading/unloading requests for the humans.

For two machines with the co-ordinates $M_m(x_m, y_m)$ and $M_{m'}(x_{m'}, y_{m'})$, the distance between them can be obtained as

$$d_{m,m'} = \sqrt{(x_m - x_{m'})^2 + (y_m - y_{m'})^2}. \quad (2)$$

Machines undertake two types of processes: loaded processes and no-load processes. Each part of energy consumption can be calculated by using the processing power and the processing time. According to the loaded processing time, t_{jo}^{proce} , and the loaded processing power, p_m^{proce} , when operation O_{jo} is on machine M_m , we can obtain the loaded processing energy consumption, e_m^{proce} , of machine M_m . We have

$$e_m^{proce} = p_m^{proce} t_{jo}^{proce}. \quad (3)$$

For the no-loaded processing power, p_m^{noload} , and the no-loaded processing time, t_m^{noload} , when there is no operation is on machine M_m , the no-loaded processing energy consumption, e_m^{noload} , of machine M_m can be obtained.

$$e_m^{noload} = p_m^{noload} t_m^{noload} \quad (4)$$

Thirdly, a set of jobs $J = \{J_1, J_2, \dots, J_j, \dots, J_{jmax}\}$ are to be executed on one or more of these machines. Each job J_j is composed of a sequence of steps, $J_j = \{O_{j1}, O_{j2}, \dots, O_{jo}, \dots, O_{jomax}\}$, where the term operation (O_{jo}) is used to describe its combination of the steps.

The completion time, t_j^{cmax} , of job J_j is equal to the sum of the processing time, t_{jo}^{proce} , of all operations on each machine and the operating time, $t_{m,m'}^h$, of humans between machines. There is

$$t_j^{cmax} = \sum_{jo} t_{jo}^{proce} + \sum_h \sum_m t_{m,m'}^h. \quad (5)$$

The makespan of all jobs, t_{Σ}^{max} , depends on the maximum value of completion time, t_j^{max} , on a set of jobs $J = \{J_1, J_2, \dots, J_j, \dots, J_{jmax}\}$, that is,

$$t_{\Sigma}^{max} = \max_j \{t_j^{max}\}. \quad (6)$$

Fourthly, each operation O_{j_0} must be performed on one of several possible machines without interruption. No machine can handle more than one operation simultaneously. A human facilitates the movement of a part from one operation O_{j_0} to the subsequent operation $O_{j(o+1)}$, transferring the part from the source machine M_m to the target machine $M_{m'}$ for further processing.

If the end time of operation O_{j_0} on machine M_m is $t_{j_0}^{end}$ and the start time of operation $O_{j(o+1)}$ on machine $M_{m'}$ is $t_{j(o+1)}^{start}$, and the difference between the two times is less than the operating time, $t_{m,m'}^h$, on humans, a human–machine conflict will occur. Then, the human–machine conflict time, t_j^{conf} , of job J_j can be given as follows:

$$t_j^{conf} = \sum_{j_0} \left[\left| t_{j(o+1)}^{start} - t_{j_0}^{end} \right| - t_{m,m'}^h \right] \quad (7)$$

3.2. Main Assumptions

The CHMCMP can be defined as the task of determining the operation sequence, as well as the start and end times for each job at the respective machines, alongside assigning human transfers between these machines, with the goal of completing all jobs in a manner that improves manufacturing performance, conflict-free human–machine collaboration, and lowers no-load energy consumption.

This formulation is predicated on the following main assumptions:

- The number of humans is known, and they initially start from the stocker;
- Loading, and unloading times are deterministic and known;
- The number of machines is known, and each machine has sufficient input/output buffer space to prevent deadlocks;
- Processing times are deterministic and known;
- The human transfers and machine operations are non-preemptive;
- All humans are homogeneous, all machines are homogeneous, all jobs are homogeneous, without considering the preferences of different humans and machines for processing jobs;
- Neither the impact of human diseases and fatigue on operation nor machine failures are considered.

Additionally, it is posited that the presence of a human is requisite to facilitate the load/unload process when a machine undertakes the processing of a job for any two successive operations.

3.3. Conflict-Free Decision Variables

According to the human–machine conflict time in Equation (7), it is necessary to arrange human operations and machine processing reasonably to avoid conflicts between the two. Therefore, three types of decision variables are defined here, i.e., the Boolean variable, the interval variable, and the sequential variable.

A Boolean variable is a type of logical data type that contains only two possible values: true (1) or false (0). It can be used to choose specific decisions or not.

An interval variable is a type of decision variable whose domain is a subset of $\{t_{j_0} = (t_{j_0}^{start}, t_{j_0}^{end})\}$, where $t_{j_0}^{start}$ and $t_{j_0}^{end}$ denote the start and end times of the operation interval, respectively. This variable can help us arrange operations to eliminate the

need for multiple decision variables in binary integer programming. In this context, interval variables are employed to characterize both human operations and machine processing.

A sequential variable is constructed from a collection of interval variables, representing a specific ordering of these intervals. This variable's value is a permutation that includes all the intervals that are present. For example, given a set $t1 = \{t_{jo}, t_{j(o+1)}\}$ where interval variable $t_{jo} = (t_{jo}^{start}, t_{jo}^{end})$ is required and $t_{j(o+1)} = (t_{j(o+1)}^{start}, t_{j(o+1)}^{end})$ is optional, the sequential variable defined on $t1$ would have two possible permutations in its domain: $\{(t_{jo}), (t_{jo}, t_{j(o+1)})\}$. Hence, this sequential variable can compile and sequence the interval variables that are allocated to various machines and humans.

In addressing the CHMCMP, the following decision variables are employed based on above three types of decision variables:

$\alpha_{jo,h}, J_j \in J, O_{jo} \in J_j, H_h \in H$: A Boolean variable donates that operation O_{jo} is currently assigned to human H_h .

$\beta_{jo,m}, J_j \in J, O_{jo} \in J_j, M_m \in M$: A Boolean variable indicates that operation O_{jo} is currently assigned to machine M_m .

$t_{j,h,m'}^{load}, J_j \in J, H_h \in H, M_m \in M$: An interval variable signifies the loading task for operation O_{jo} and it represents the time interval within which human H_h loads job J_j finished in the previous operation from the input buffer on machine M_m for the ongoing operation.

$t_{jo,h,m'}^{unld}, J_j \in J, H_h \in H, M_m \in M$: An interval variable denotes the unloading task for operation O_{jo} , delineating the timeframe during which human H_h unloads job J_j to the output buffer on machine M_m for subsequent transfer.

$t_{jo,h,m,m'}^{trans}, J_j \in J, O_{jo} \in J_j, H_h \in H, M_m, M_{m'} \in M$: An interval variable for the transfer state of operation O_{jo} on human H_h from the output buffer on machine M_m to the input buffer on machine $M_{m'}$, and $M_m \neq M_{m'}$.

$t_{jo,m}^{proc}, O_{jo} \in J_j, M_m \in M$: An interval variable for operation O_{jo} being processed on machine M_m .

$\{t_{jo,h,m'}^{load}, t_{jo,m}^{proc}, t_{jo,h,m}^{unld} | O_{jo} \in J_j, H_h \in H, M_m \in M\}$: A sequential variable that encapsulates all possible permutation sequences of the interval variables $(t_{jo,h,m'}^{load}, t_{jo,m}^{proc}, t_{jo,h,m}^{unld})$ allocated to the operations on machine M_m .

$\{t_{jo,h,m,m'}^{unld}, t_{jo,h,m,m'}^{trans}, t_{jo,h,m}^{load} | O_{jo} \in J_j, H_h \in H, M_m, M_{m'} \in M\}$: A sequential variable that captures all potential permutation sequences of the interval variables $(t_{jo,h,m,m'}^{unld}, t_{jo,h,m,m'}^{trans}, t_{jo,h,m}^{load})$ assigned to human H_h .

These conflict-free decision variables should be reasonably arranged to avoid conflicts between human operations and machine processing. To maintain a record of each human's final location after completing a transfer, it is essential to incorporate a tracer to track the last machine associated with each human's transfer activity.

3.4. Multi-Objective Function

Now the CHMCMP can be modeled as a job-flow-shop scheduling problem with nonlinear constraints. Our goal is to complete all manufacturing jobs while pursuing high manufacturing performance, conflict-free human-machine collaboration, and low no-load energy consumption. The manufacturing performance can be expressed through the makespan of all jobs, t_{Σ}^{cmax} , in Equation (6). The no-load energy consumption can be determined using the no-loaded processing energy consumption, e_m^{noload} , in Equation (4).

Then, the multi-objective function of the CHMCMP can be formulated into *JFSP* as follows:

$$JFSP = \left\{ \min(t_{\Sigma}^{cmax}), \min\left(\sum_m e_m^{noload}\right) \right\} \quad (8)$$

In Equation (8), the first part of *JFSP* aims to minimize the total time required to complete all jobs, known as the makespan, seeking higher manufacturing performance, and the second part focuses on low no-load energy consumption. These two goals are both intertwined and conflicting with each other.

3.5. Constraints

The CHMCMP also incorporates various constraints, mainly divided into two categories, i.e., mutually exclusive constraints and temporal constraints.

The mutually exclusive constraint refers to the mutually exclusive relationship between various variables and is utilized to allocate operations or transfers to a set of mutually exclusive resources. If operation O_{jo} is scheduled, it can only be in one of four states: loaded by human H_h , transferred by human H_h , unloaded by human H_h , and processed by machine M_m . This constraint restricts an operation to be processed only by one human or one machine, ensuring that only one resource is engaged for each operation or transfer. Therefore, an interval variable should coincide with the start and end times.

The temporal constraints are utilized to dictate the temporal relationship between two interval variables and ensure that the loading, processing, unloading, and transfer of a part are carried out in sequential order. For example, a part can only be processed by a machine when it is loaded onto the machine from the input buffer; and a processed part can only be transferred after being unloaded from the machine to the output buffer by a human.

Based on the above two types of constraints, the main constraints are as follows:

$$\alpha_{jo,h} \in [0, 1], J_j \in J, O_{jo} \in J_j, H_h \in H \quad (9)$$

$$\beta_{jo,m} \in [0, 1], J_j \in J, O_{jo} \in J_j, M_m \in M \quad (10)$$

$$\text{no_duplicates}(O_{jo}), O_{jo} \in J_j \quad (11)$$

$$0 \leq \text{count}(O_{jo} \text{ on } H_h) \leq 1, J_j \in J, H_h \in H \quad (12)$$

$$0 \leq \text{count}(O_{jo} \text{ on } M_m) \leq 1, J_j \in J, M_m \in M \quad (13)$$

$$t_{jo}^{\text{start}} < t_{jo}^{\text{end}}, O_{jo} \in J_j \quad (14)$$

$$t_{jo}^{\text{end}} < t_{j(o+1)}^{\text{start}}, O_{jo}, O_{j(o+1)} \in J_j \quad (15)$$

$$t_{jo,h,m}^{\text{load}} < t_{j(o+1),h,m'}^{\text{load}}, O_{jo}, O_{j(o+1)} \in J_j, H_h \in H, M_m, M_{m'} \in M, M_m \neq M_{m'} \quad (16)$$

$$t_{jo,h,m}^{\text{load}} < t_{jo}^{\text{start}}, O_{jo} \in J_j, H_h \in H, M_m \in M \quad (17)$$

$$t_{jo}^{\text{end}} < t_{jo,h,m}^{\text{unld}}, O_{jo} \in J_j, H_h \in H, M_m \in M \quad (18)$$

$$t_{jo,h,m}^{\text{load}} < t_{jo,m}^{\text{proc}} < t_{jo,h,m'}^{\text{unld}}, O_{jo} \in J_j, H_h \in H, M_m \in M \quad (19)$$

$$t_{jo,h,m}^{\text{unld}} < t_{j(o+1),h,m,m'}^{\text{trans}} < t_{j(o+1),h,m'}^{\text{load}}, O_{jo}, O_{j(o+1)} \in J_j, H_h \in H, M_m, M_{m'} \in M, M_m \neq M_{m'} \quad (20)$$

$$0 < t_{j,m}^{\text{proc}} < t_{jo}^{\text{end}} - t_{jo}^{\text{start}}, O_{jo} \in J_j, M_m \in M \quad (21)$$

$$0 < t_{j(o+1),h,m,m'}^{\text{trans}} < t_{j(o+1),h,m'}^{\text{load}} - t_{jo,h,m}^{\text{unld}}, O_{jo}, O_{j(o+1)} \in J_j, H_h \in H, M_m, M_{m'} \in M, M_m \neq M_{m'} \quad (22)$$

$$t_{jo,h,m}^{\text{load}} + t_{jo,m}^{\text{proc}} + t_{jo,h,m}^{\text{unld}} < t_{j(o+1),h,m,m'}^{\text{trans}}, O_{jo}, O_{j(o+1)} \in J_j, H_h \in H, M_m, M_{m'} \in M, M_m \neq M_{m'} \quad (23)$$

$$t_{jo,h,m}^{\text{unld}} + t_{jo,h,m,m'}^{\text{trans}} + t_{j(o+1),h,m'}^{\text{load}} < t_{j(o+1),h,m'}^{\text{load}}, O_{jo}, O_{j(o+1)} \in J_j, H_h \in H, M_m, M_{m'} \in M, M_m \neq M_{m'} \quad (24)$$

$$t_j^{\text{conf}} = 0 \quad (25)$$

Constraints (9) and (10) limit the logical data types of Boolean variables to have only two possible values: true (1) or false (0). Constraint (11) confines a job to always have at most one operation in progress. Constraint (12) prevents more than one job from being scheduled as a loading/unloading/transfer task on the same human. Constraint (13) ensures that a machine can only process at most one operation at a time. Constraint (14) limits the end time of each operation to not be earlier than its start time. Constraint (15) establishes the necessary sequencing of operations for each job, ensuring that they follow the correct order. Constraint (16) ensures that each operation is assigned to one specific machine from a set of possible machines. Constraint (17) assigns each loading task to a human before the start time of the operation. Constraint (18) stipulates that the unloading task can only be done after the end time of the operation. Constraint (19) limits the processing time to only after the loading time and before the unloading time. Constraint (20) maintains the unloading time of the previous operation and the loading time of the next operation, which must be synchronized with the human transfer operation. Constraint (21) determines the minimum and maximum bounds of processing time. Constraint (22) indicates the lower and upper bounds of transfer time. Constraint (23) ensures that the sequential variable indicates that encapsulates all possible permutation sequences of the interval variables $(t_{jo,h,m}^{load}, t_{jo,m}^{proc}, t_{jo,h,m}^{unld})$ allocated to the operations on machine M_m . Constraint (24) accounts for the sequential variable that captures all potential permutation sequences of the interval variables $(t_{jo,h,m'}^{unld}, t_{jo,h,m,m'}^{trans}, t_{jo,h,m}^{load})$ assigned to human H_h . Constraint (25) realizes conflict-free human-machine collaboration.

4. Methodology

Given that FJSP and FSP are both NP-hard, jointly solving FJSP and FSP in Section 3 is also NP-hard, and traditional methods struggle to solve it. This section developed an improved APC algorithm to efficiently achieve optimal results which employs a random-pairwise single-elimination tournament system.

4.1. APC-Based Scheduling Architecture

The APC-based scheduling architecture employs a metaheuristic search strategy that integrates seeding, growing, and fruiting to efficiently guide the global search, local search, and fitness convergence process, as shown in Figure 3.

In Figure 3, the APC-based scheduling architecture defines the conflict-free human-machine collaborative manufacturing problem as a job-flow-shop scheduling problem with a lot of constraints. Our goal is to complete all manufacturing jobs while pursuing higher manufacturing performance, conflict-free human-machine collaboration, and low no-load energy consumption. The solution process is mainly implemented through the following four main steps.

Step 1: Job analysis and manufacturing system evaluation. This step involves systematic job analysis, data collection, organization, modeling, and mining centered around customer orders to support scientific decision-making in manufacturing, sales, inventory, operations, and customer management for enterprises. After evaluation and analysis, a job set, $J = \{J_1, J_2, \dots, J_j, \dots, J_{jmax}\}$, a human set, $H = \{H_1, H_2, \dots, H_h, \dots, H_{hmax}\}$, and a machine set, $M = \{M_1, M_2, \dots, M_m, M_{m'}, \dots, M_{mmax}\}$, will be generated, where each job J_j contains a series of operations, $J_j = \{O_{j1}, O_{j2}, \dots, O_{j0}, \dots, O_{jmax}\}$.

Step 2: CHMCMP initialization.

The parameters of the CHMCMP will be initialized at first, including the total number of humans that can be scheduled, h_{max} , the operation/transfer rate of each human, $\{r_h\}$, the total number of machines that can be scheduled, m_{max} , the distance between machines, $\{d_{m,m'}\}$, the loaded processing power of each machine, $\{p_m^{proce}\}$, the no-loaded processing

power, $\{p_m^{noload}\}$, the total number of jobs that can be scheduled, j_{max} , the total number of operations in each job, $\{j_{omax}\}$, and the loaded processing time of each job, $\{t_{jo}^{proce}\}$. The conflict-free human-machine collaboration can be determined by the human-machine conflict time, t_j^{conf} , in Equation (7).

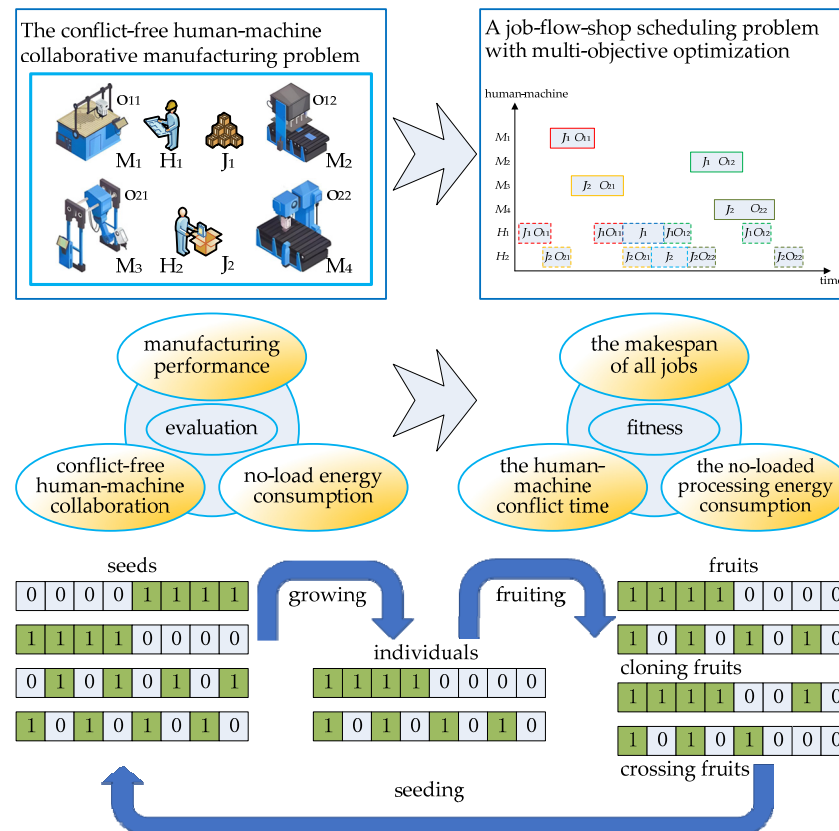


Figure 3. APC-based scheduling architecture.

The manufacturing performance can be measured through the makespan of all jobs, t_{Σ}^{cmax} , in Equation (6). The no-load energy consumption can be evaluated using the no-loaded processing energy consumption, e_m^{noload} , in Equation (4).

The parameters of the APC will be initialized also, i.e., the population size S of the APC, the seeding probability p_{seed} , the growing probability p_{grow} , and the fruiting probability p_{fruit} . Each APC individual is encoded as a binary string x :

$$x = \{\alpha_{jo,h}, \beta_{jo,m}\} \quad (26)$$

The fitness of the APC can be calculated using $JFSP$ in Equation (8), but the larger the value, the higher the fitness, as follows:

$$fitness = -JFSP \quad (27)$$

The end condition of the APC iteration also needs to be initialized. When the intermediate calculation results meet the end condition, the iteration can be terminated and the solution with the best fitness will be output. The commonly used conditions for ending iterations are the maximum number of iterations and the iteration error threshold. They determine whether the APC iteration continues or ends.

Step 3: APC-based iterative search for the JFSP.

As a metaheuristic algorithm, the optimal solution of the APC is not constrained by initial values, but depends on the heuristic learning process.

Step 3-1: Seeding operation to generate random feasible solutions.

It is to randomly produce APC seeds in each iteration for global search, which is decided by the seeding probability $p_{seed} \in [0, 1]$. The seeding population $\{x_{seed}(k)\}$ in iteration k includes two parts: the random produced seeds $\{x_{seed}^{rand}(k)\}$ by p_{seed} and the fruits $\{x_{seed}^{fruit}(k-1)\}$ in the previous iteration by $(1 - p_{seed})$. There is

$$\{x_{seed}(k)\}_S = \{x_{seed}^{rand}(k)p_{seed}, x_{seed}^{fruit}(k-1)(1 - p_{seed})\}_S. \quad (28)$$

It should be noted that, there are no fruits ($\{x_{seed}^{fruit}(k-1)\} = \emptyset$) in the first iteration ($k = 0$), and all seeds $\{x_{seed}^{rand}(k=0)\}$ are randomly produced.

Step 3-2: Growing operation to select elite feasible solutions.

A random-pairwise single-elimination tournament system is introduced for elite selection, with a time complexity of $O(S)$, better than $O(S^3)$ of GA [13] and $O(S^2)$ of NSGA-II [6]. It is to provide the fitness selection for the APC individuals, which is determined by a growing probability $p_{grow} \in [0, 1]$. According to the fitness function in Equation (27), the optimal seeds $\{x_{seed}(k)\}$ will be selected by p_{grow} and the growing population size will decrease to Sp_{grow} . We can obtain

$$\{x_{grow}(k)\}_{Sp_{grow}} = \{x_{seed}(k) | \max_{Sp_{grow}} fit(x_{seed}(k))\}_{Sp_{grow}}. \quad (29)$$

Step 3-3: Fruiting operation to allow elites to produce more feasible solutions.

It is to produce more APC fruits in each iteration for local search, which is decided by a fruiting probability of $p_{fruit} \in [0, 1]$. The growing individuals $\{x_{grow}(k)\}$ have opportunities to produce parthenogenesis fruits $\{x_{fruit}(k)\}$ by cloning and the crossing fruits by string concatenation, where p_{fruit} determines how much genetic material a growing individual $\{x_{grow}(k)\}$ can retain. Assuming two parents, $x_{grow}(k)$ and $x'_{grow}(k)$, they can produce two fruits as follows:

$$\begin{cases} x_{fruit}^{cross}(k) = \{x_{grow}(k)_{p_{fruit}} // x'_{grow}(k)_{1-p_{fruit}}\} \\ x_{fruit}^{cross'}(k) = \{x'_{grow}(k)_{p_{fruit}} // x_{grow}(k)_{1-p_{fruit}}\} \end{cases} \quad (30)$$

where $//$ is the string concatenation.

A set of fruits can be randomly selected from Equation (30), $x_{fruit}^{cross}(k)$ or $x_{fruit}^{cross'}(k)$. Hence, the total fruiting population size has doubled as follows:

$$\{x_{fruit}(k)\}_{2Sp_{grow}} = \left\{ \{x_{grow}(k)\}_{Sp_{grow}}, \{x_{fruit}^{cross}(k)\}_{Sp_{grow}} \right\} \quad (31)$$

Step 3-4: End judgment to determine whether to terminate the search for feasible solutions.

End judgment can be based on the end conditions initialized in step 2, such as a predefined maximum number of iterations or the iteration error threshold. If the end conditions are not fit, the APC will return to step 3-2 for the next iterative seeding. Otherwise, the best solution can be output, and the iterative calculation terminated.

Step 4: Implementation of scheduling and manufacturing based on the optimal solution.

Publish the optimal solution to each workshop, clarify the tasks, processes, working hours, and quality standards for both humans and machines. Real-time tracking of progress using information management or data systems can be employed to ensure opti-

mal execution of human–machine scheduling. Monitor and define anomaly types (such as human–machine conflicts, equipment failures, material shortages, quality issues, etc.), and respond quickly through manufacturing anomaly reports. When necessary, readjust scheduling, call backup resources, and even re-optimize human–machine scheduling or manufacturing resources.

4.2. Flowchart

APC-based iterative search for the JFSP is shown in Figure 4, where seeding, growing, and fruiting construct a large loop. Before the loop, there is the parameter initialization and random solution generation. The seeding step is to search for as many feasible solutions as possible through global search, with the population size, S . The growing step is to select optimal APC individuals for fitness convergence, with a decreased population size, Sp_{grow} . The fruiting step is to search for more optimal solutions through local search, with an increased population size, $2Sp_{grow}$. After multiple iterations, it is possible to find the optimal solutions to the conflict-free human–machine collaborative manufacturing problem.

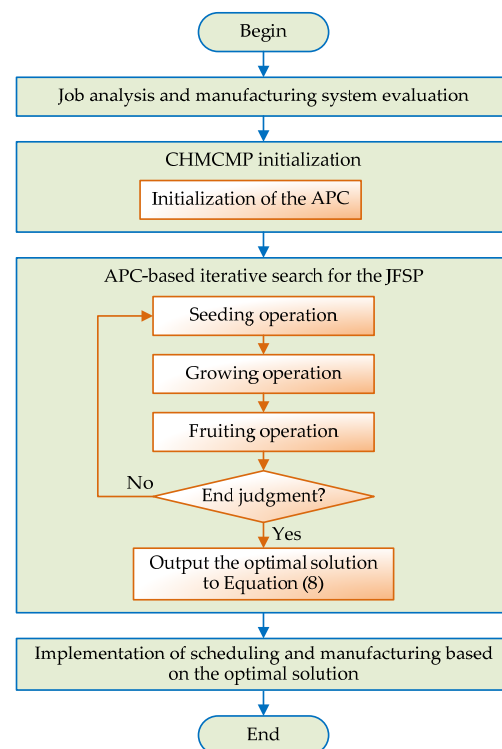


Figure 4. APC-based iterative search for the JFSP.

According to Figure 4, the computational complexity of the APC-based scheduling architecture is linearly influenced by the population size (S), number of iterations (k_{max}), and the cost of the fitness function evaluation $C_{fitness}(h_{max}, m_{max}, j_{max})$. The cost of the fitness function evaluation depends on the total number of humans (h_{max}), the total number of machines (m_{max}), and the total number of jobs (j_{max}). Therefore, the time complexity of the proposed algorithm can be calculated as $O(S k_{max} C_{fitness}(h_{max}, m_{max}, j_{max}))$.

4.3. Pseudocode

Algorithm 1 illustrates the pseudocode for the APC-based scheduling architecture. Before line 1, the inputs and outputs are set. Line 1 shows Step 1 job analysis and manufacturing system evaluation. Lines 2 through 6 are Step 2 CHMCMP initialization, including the objective function, constraints, APC parameter set, and the randomly initialized solution. Lines 7–24 determine Step 3 APC-based iterative search for the JFSP. Line 9 is the

seeding operation, Line 10 is a random-pairwise single-elimination tournament system, and Line 11 is the growing operation and. Line 17 calculates the fitness, and Lines 19~20 are the fruiting operation. Line 21 checks whether the feasible solution meets the constraints in Equations (9)–(25), and invalid solutions that do not meet the constraints in Line 4 will be discarded. Line 22 implements the end judgment. If the end conditions are not fit, the APC will return to Line 8 for the next iterative seeding. Otherwise, the best solution can be output and the iterative calculation can be terminated. Lines 25~27 conduct Step 4, implementation of scheduling and manufacturing based on the optimal solution.

Algorithm 1: APC-based scheduling architecture

Input: $J = \{J_1, J_2, \dots, J_j, \dots, J_{jmax}\}$, $H = \{H_1, H_2, \dots, H_h, \dots, H_{hmax}\}$,
 $M = \{M_1, M_2, \dots, M_m, \dots, M_{mmax}\}$,
 $J_j = \{O_{j1}, O_{j2}, \dots, O_{jo}, \dots, O_{jmax}\}$, $\{r_h\}$, $\{d_{m,m'}\}$, $\{p_m^{proce}\}$, $\{p_m^{noload}\}$, $\{t_{jo}^{proce}\}$.
 Output: $x = \{\alpha_{jo,h}, \beta_{jo,m}\}$

- 1: Step 1: job analysis and manufacturing system evaluation
- 2: Step 2: CHMCMP initialization
- 3: Objective function: Equations (4), (6) and (8)
- 4: Constraints: Equations (9)–(25)
- 5: Set: S , p_{seed} , p_{grow} , p_{fruit} , and $fitness$.
- 6: $x(0) = rand\{\alpha_{jo,h}, \beta_{jo,m}\}$
- 7: Step 3: APC-based iterative search for the JFSP
- 8: for $k = 1 : k_{max}$
- 9: $\{x_{seed}(k)\}_S = \{x_{seed}^{rand}(k)p_{seed}, x_{seed}^{fruit}(k-1)(1-p_{seed})\}_S$
- 10: a random-pairwise single-elimination tournament system
- 11: $\{x_{grow}(k)\}_{Sp_{grow}} = \left\{x_{seed}(k) | \max_{Sp_{grow}} fit(x_{seed}(k))\right\}_{Sp_{grow}}$
- 12: $t_j^{cmax} = \sum_{jo} t_{jo}^{proce} + \sum_h \sum_m t_{m,m'}^h$
- 13: $t_{\sum}^{cmax} = \max_j \{t_j^{cmax}\}$
- 14: $t_j^{conf} = \sum_j o \left[\left| t_{j(o+1)}^{start} - t_{jo}^{end} \right| - t_{m,m'}^h \right]$
- 15: $e_m^{noload} = p_m^{noload} t_m^{noload}$
- 16: $JFSP = \left\{ \min(t_{\sum}^{cmax}), \min(\sum_j t_j^{conf}), \min(\sum_m e_m^{noload}) \right\}$
- 17: $fitness = -JFSP$
- 18: $\{x(k)\} = \{x(k) | \max\{fit(x(k))\}\}_S$
- 19: $x_{fruit}^{cross}(k) = \left\{ x_{grow}(k) p_{fruit} / x'_{grow}(k)_{1-p_{fruit}} \right\}$
- 20: $\{x_{fruit}(k)\}_{2Sp_{grow}} = \left\{ \{x_{grow}(k)\}_{Sp_{grow}}, \{x_{fruit}^{cross}(k)\}_{Sp_{grow}} \right\}$
- 21: invalid solutions that do not meet the constraints in Line 5 will be discarded
- 22: if $|\max(fitness(x(k))) - \max(fitness(x(k-1)))| \geq e_{th}$, then return to line 8
- 23: end for
- 24: output the best solution $x^*(k)$
- 25: Step 4: implementation of scheduling and manufacturing based on the optimal solution
- 26: assign jobs $J = \{J_1, J_2, \dots, J_j, \dots, J_{jmax}\}$ to humans $H = \{H_1, H_2, \dots, H_h, \dots, H_{hmax}\}$
- 27: assign jobs $J = \{J_1, J_2, \dots, J_j, \dots, J_{jmax}\}$ to machines $M = \{M_1, M_2, \dots, M_m, \dots, M_{mmax}\}$

Algorithm 1 aims to complete all manufacturing jobs, $J = \{J_1, J_2, \dots, J_j, \dots, J_{jmax}\}$, while aiming to improve manufacturing performance ($\min(t_{\sum}^{cmax})$), realize conflict-free human-machine collaboration ($\sum_j t_j^{conf} = 0$), and minimize no-load energy consumption ($\min(\sum_m e_m^{noload})$). The time complexity of Algorithm 1 can be calculated as $O(S k_{max} C_{fitness}(h_{max}, m_{max}, j_{max}))$, is linearly related to the population size (S), number

of iterations (k_{max}), and the cost of the fitness function evaluation $C_{fitness}(h_{max}, m_{max}, j_{max})$. The cost of the fitness function evaluation depends on the total number of humans (h_{max}), the total number of machines (m_{max}), and the total number of jobs (j_{max}).

The random-pairwise single-elimination tournament system is shown in Algorithm 2. The inputs and outputs are set before Line 1. Lines 1 to 8 form a large loop with $S(1 - p_{grow})$ iterations. Therefore, the complexity of this algorithm depends on how many feasible solutions are eliminated, rather than selecting the optimal batch of solutions. On Line 2, two different seeds are randomly selected to form a pair. From Lines 3 to 7, these two seeds engage in fitness competition and elimination, with winner-up and loser-out. The seed with high fitness is retained in the seed population, while the other seed with low fitness is eliminated. After a round of competition, the size of the seed population is reduced by one. After $S(1 - p_{grow})$ of comparisons, the seed population size decreased to Sp_{grow} , and they became elite individuals, as shown in Lines 9–10.

Algorithm 2: A random-pairwise single-elimination tournament system

Input : $\{x_{seed}(i)\}_S$, $S_{temp} = S$
Output : $\{x_{grow}(i)\}_{Sp_{grow}}$

```

1:  for  $i = 1 : S(1 - p_{grow})$ 
2:    randomly select two seeds for a pair :  $x_{seed}(rand(0)) \neq x_{seed}(rand(1))$ 
3:    if  $fitness(x_{seed}(rand(0))) \geq fitness(x_{seed}(rand(1)))$ 
4:      winner-up :  $x_{seed}(rand(0))$  is reserved
5:      loser-out :  $x_{seed}(rand(1))$  is eliminated
6:       $S_{temp} = S_{temp} - 1$ 
7:    end if
8:  end for
9:   $\{x_{grow}(i)\}_{Sp_{grow}} = \{x_{seed}(i)\}_{S_{temp}}$ 
10: output the elite selection  $\{x_{grow}(i)\}_{Sp_{grow}}$ 

```

Algorithm 2 is committed to quickly eliminating solutions with low fitness while ensuring that the solution with the highest fitness can be retained. The time complexity of Algorithm 2 can be calculated as $O(S(1 - p_{grow}))$ or $O(S)$, and is linearly related to the population size (S).

For ease of understanding, comparison of APC with GA [13,25] and NSGA-II [8] is shown in Table 1. It can be seen that the proposed APC algorithm has made significant improvements in global search capability and elite selection, abandoning the sorting-based elite selection algorithms. Our algorithm cannot guarantee that all selected elite individuals are the best, but it has a linear time complexity and the best individual can be retained by 100%.

Table 1. Comparison of APC with GA and NSGA II.

	GA	NSGA II	APC
global search capability	-	-	random seeds in each iteration
local search capability	crossover and mutation	elitist strategy and crowded-comparison	clone fruits and variant fruits
convergence capability	selection sorting algorithm	a fast and elitist non-dominated sorting algorithm	a random-pairwise single-elimination tournament system
computational cost for elite selection	$O(S^3)$	$O(S^2)$	$O(S)$ or $O(S(1 - p_{grow}))$

5. Benchmark Experiments

In this section, medium-scale benchmark dataset is developed and a set of benchmark experiments is implemented to validate and compare the APC-based scheduling architecture and some baseline algorithms. The Gantt charts of machine processing FJSP and human operating FSP are provided.

5.1. Medium-Scale Benchmark Dataset

The benchmark data set is based on data in reference [5], and is extended to a larger scale to test the conflict-free human–machine collaborative manufacturing problem. The benchmark roadmap is a human–machine collaborative workshop, as shown in Figure 5, where there are fifteen humans $H = \{H_1, H_2, \dots, H_{15}\}$ and fifteen machines $M = \{M_1, M_2, \dots, M_{15}\}$. All machines are evenly distributed on a rectangular workshop measuring 16 m by 8 m, with a distance of 4.0 m between adjacent machines.

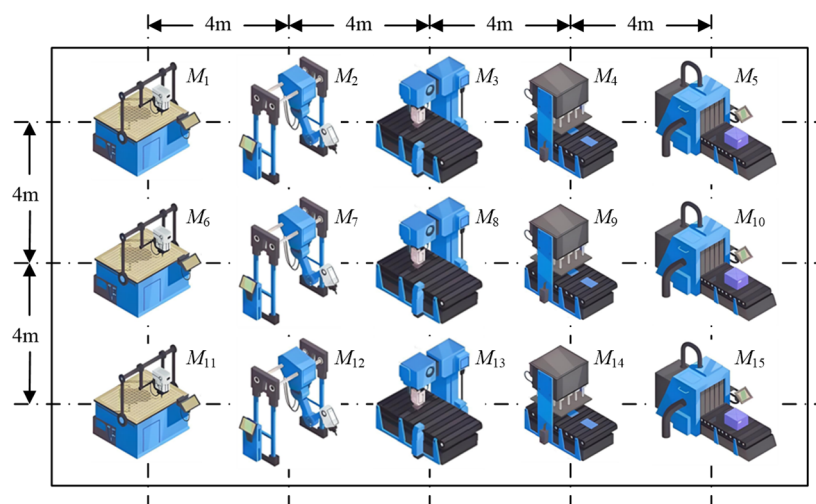


Figure 5. Benchmark roadmap of human–machine collaborative manufacturing.

It is assumed that the first machine, M_1 , is the start node, and all jobs are set on machine M_1 from the beginning. The human–machine collaboration continues until all jobs are completed. Based on Figure 5, the distance between machines in the benchmark roadmap, $d_{m,m'}$, can be calculated, as shown in Table 2.

Table 2. The distance between machines in the benchmark roadmap (meters).

Machines	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}
M_1	0	4	8	12	16	4	8	12	16	20	8	12	16	20	24
M_2	4	0	4	8	12	8	4	8	12	16	12	8	12	16	20
M_3	8	4	0	4	8	12	8	4	8	12	16	12	8	12	16
M_4	12	8	4	0	4	16	12	8	4	8	20	16	12	8	12
M_5	16	12	8	4	0	20	16	12	8	4	24	20	16	12	8
M_6	4	8	12	16	20	0	4	8	12	16	4	8	12	16	20
M_7	8	4	8	12	16	4	0	4	8	12	8	4	8	12	16
M_8	12	8	4	8	12	8	4	0	4	8	12	8	4	8	12
M_9	16	12	8	4	8	12	8	4	0	4	16	12	8	4	8
M_{10}	20	16	12	8	4	16	12	8	4	0	20	16	12	8	4
M_{11}	8	12	16	20	24	4	8	12	16	20	0	4	8	12	16
M_{12}	12	8	12	16	20	8	4	8	12	16	4	0	4	8	12
M_{13}	16	12	8	12	16	12	8	4	8	12	8	4	0	4	8
M_{14}	20	16	12	8	12	16	12	8	4	8	12	8	4	0	4
M_{15}	24	20	16	12	8	20	16	12	8	4	16	12	8	4	0

In the benchmark data set, there are fifteen jobs, $J = \{J_1, J_2, \dots, J_{15}\}$ are to be executed on one or more of these machines, and each job is composed of a sequence of operations, $J_j = \{O_{j1}, O_{j2}, \dots, O_{j15}\}$. The loaded processing time of each job, $\{t_{jo}^{proce}\}$ is shown in Table 3. The loaded processing time of each job is in the range of [5, 24] min.

Table 3. The loaded processing time (minutes) of each job.

Operation	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}	J_{11}	J_{12}	J_{13}	J_{14}	J_{15}
O_{j1}	15	7	6	11	18	8	21	15	9	14	12	17	16	20	10
O_{j2}	21	16	10	12	8	17	19	22	20	6	5	15	7	14	13
O_{j3}	17	23	8	16	13	6	22	7	11	21	20	14	15	18	12
O_{j4}	12	22	23	15	11	20	10	9	6	14	18	13	16	7	8
O_{j5}	13	14	8	17	19	10	16	5	7	6	15	12	21	24	11
O_{j6}	6	9	17	12	7	22	11	21	10	15	16	13	19	14	5
O_{j7}	20	12	5	15	11	16	13	10	9	14	8	7	10	6	23
O_{j8}	11	9	21	18	12	17	15	8	14	16	7	5	6	10	19
O_{j9}	16	5	7	6	15	12	21	23	11	13	14	8	17	9	10
O_{j10}	5	6	11	8	9	13	18	24	14	12	17	16	22	10	7
O_{j11}	10	15	8	14	16	5	7	6	10	24	11	9	14	13	12
O_{j12}	13	6	18	7	9	14	5	19	15	17	12	10	20	8	16
O_{j13}	24	11	5	21	5	6	14	17	13	16	7	8	12	22	15
O_{j14}	7	10	15	16	13	18	14	25	6	11	19	12	9	23	24
O_{j15}	14	17	11	22	12	15	9	13	8	7	10	6	24	21	5

The processing power (kW) for each machine is provided in Table 4, which comes from a set of machines. The loaded processing power of each machine, $\{p_m^{proce}\}$, is within the range of [1.5, 5.0] (kW), as shown in the first line in Table 4. The no-loaded processing power, $\{p_m^{noload}\}$, consumed by the machines is within the range of [0.3, 1.2] (kW), as shown in the second line in Table 4.

Table 4. The processing power (kW) for each machine.

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}
p_m^{proce}	1.5	2.0	3.0	5.0	4.0	1.5	2.0	3.0	5.0	4.0	1.5	2.0	3.0	5.0	4.0
p_m^{noload}	0.3	0.4	0.5	1.2	1.0	0.3	0.4	0.5	1.2	1.0	0.3	0.4	0.5	1.2	1.0

Based on the loaded processing power, p_m^{proce} , in Table 4, and the loaded processing time, t_{jo}^{proce} , the loaded processing energy consumption, e_m^{proce} , of machine m can be calculated through Equation (3). According to the no-loaded processing power, p_m^{noload} , in Table 4, and the no-loaded processing time, t_m^{noload} , the no-loaded processing energy consumption, e_m^{noload} , can be obtained through Equation (4).

When a human walks between machines, the operation/transfer rate of each human, $\{r_h\}$, is within the range of [20, 50] m/min. Hence, the transfer time $t_{m,m'}^h$ of human H_h on the distance of $d_{m,m'}$ can be calculated through Equation (1). For ease of analysis, assuming that the first 10% part of the transfer time is the unloading time, $t_{jo,h,m'}^{unld}$, the second 70% part is walking time, and the third 20% part is loading time, $t_{j,h,m}^{load}$.

Considering that the number of people is 15, the number of machines is 15, and the number of tasks is 15, each task includes 15 operations. Therefore, the scale of this benchmark test set is $15 \times 15 \times 15 \times 15 = 50,625$, which is a medium-sized test set. Due to these benchmark data being derived and modified from existing literature and data in reference [5], they are reasonable and suitable for testing CHMCMP.

5.2. Benchmark Results

This section provides benchmark results of the proposed APC-based scheduling architecture. The experimental platform contains an AMD Ryzen 3 4300U with Radeon Graphics 2.70 GHz CPU, 8.00 GB RAM, 64-bit Windows 10 operating system, and MATLAB R2018a simulation software. The parameters of the APC-based scheduling architecture include the population size = 1000, the seeding probability $p_s = 0.2$, the growing probability $p_g = 0.4$, and the fruiting possibility $p_f = 0.8$. The maximum number of iterations $k_{max} = 1000$, and the maximum computation time for termination = 3.0 h.

Figure 6 shows the Gantt chart of the proposed APC-based scheduling architecture. The horizontal axis indicates the machine processing time in minutes, the vertical axis indicates the machine number. The title gives the makespan value $t_{\Sigma}^{max} = 591$ min at the top of Figure 6. The fractions of the three numbers on the right side of Figure 6 label the human transfer time ($t_{m,m'}^h$) in minutes, the machine processing time (t_{jo}^{proce}) in minutes, and the maximum completion time (t_j^{cmax}) in minutes of each machine, respectively. The different colors of progress bars indicate different jobs. The three numbers enclosed in parentheses marked on each progress bar show the job number J_j , operation number O_{jo} , and the machine processing time in minutes, t_{jo}^{proce} , respectively. From Figure 6, it can be seen that a certain operation can be selected for processing on multiple machines, and the processing sequence of different jobs on the machines can be completely different, which is the flexible job-shop scheduling problem.

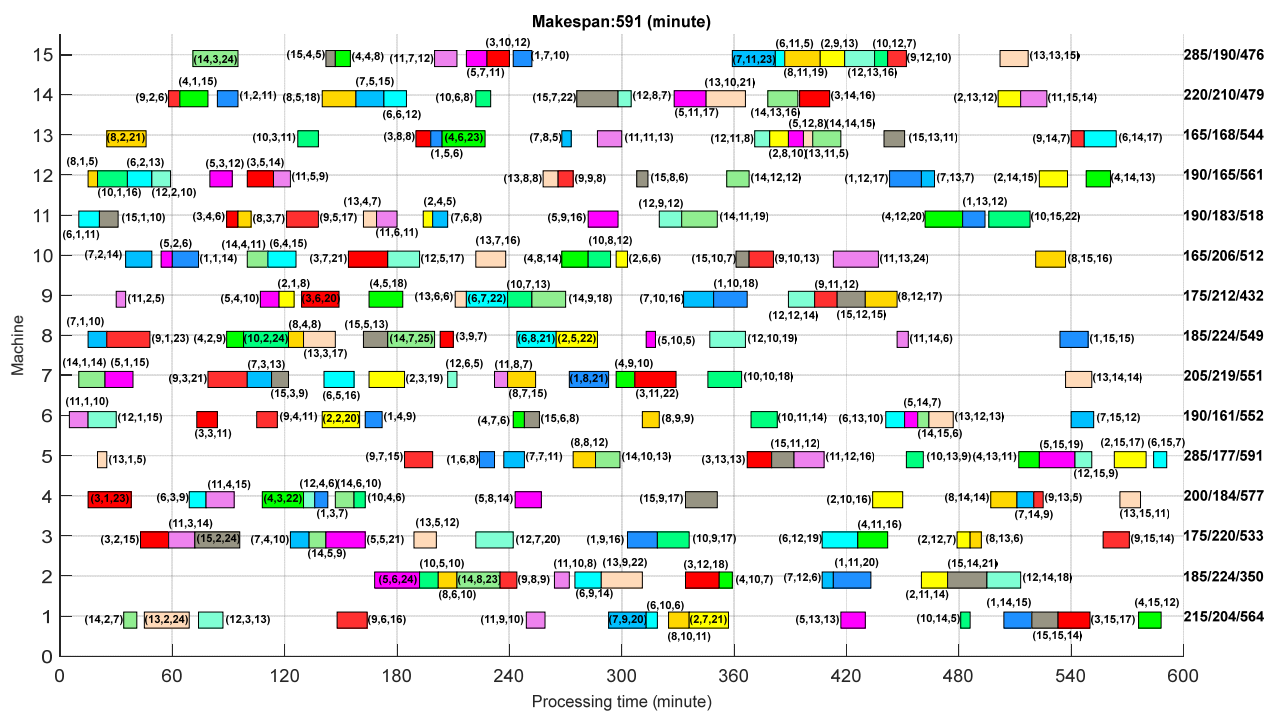


Figure 6. The Gantt chart of the machines. The different colors of progress bars indicate different jobs.

Furthermore, according to Figure 6, the processing time for each operation of each machine can be used to obtain the processing time for all operations of each machine. This is $\{t_{jo}^{proce}\} = \{190, 210, 168, 165, 183, 206, 212, 224, 219, 161, 177, 184, 220, 224, 204\}$. Therefore, the loaded processing energy consumption of machines can be obtained by multiplying the processing time and the processing power, p_m^{proce} , in Table 4 to be $\{e_m^{proce}\} = \{12.667, 17.5, 8.4, 5.5, 4.575, 13.733, 17.667, 11.2, 7.3, 4.025, 11.8, 15.333, 11, 7.467, 5.1\}$. Similarly, the no-loaded processing time, t_m^{noload} , can be calculated by subtracting the processing time (t_{jo}^{proce}) from the maximum completion time (t_j^{cmax}) in minutes of each

machine, $\{t_m^{noload}\} = \{286, 269, 376, 396, 335, 306, 220, 325, 332, 391, 414, 393, 313, 126, 360\}$. Hence, the no-loaded processing energy consumption of machines, e_m^{noload} , can be obtained by multiplying the no-loaded processing time and no-loaded processing power, p_m^{noload} , in Table 4 to be $\{e_m^{noload}\} = \{4.767, 5.38, 3.133, 2.64, 1.675, 5.1, 4.4, 2.708, 2.213, 1.955, 6.9, 7.86, 2.608, 0.84, 1.8\}$. Hence, the total energy consumption $\{e_m^{proce} + e_m^{noload}\}$ of all machines can be calculated at $153.267 + 53.980 = 207.247$ kWh.

Figure 7 displays the Gantt chart of the transfer time of humans, where the vertical axis indicates the human number, the horizontal axis indicates the transfer time of humans in minutes. The title labels the human makespan value of 584 min at the top of Figure 7. The fractions of the three numbers on the right side of Figure 7 show the human transfer time ($t_{m,m'}^h$) in minutes, and the maximum completion time (t_j^{cmax}) in minutes of each human, respectively. The different colors of progress bars indicate different jobs and operations. The three numbers enclosed in parentheses marked on each progress bar show the job number J_j , operation number O_{jo} , and the human transfer time ($t_{m,m'}^h$), respectively. From Figure 7, it can be seen that every human follows the exact same sequence of operations, which is a typical feature of assembly line production, namely the flow-shop scheduling problem.

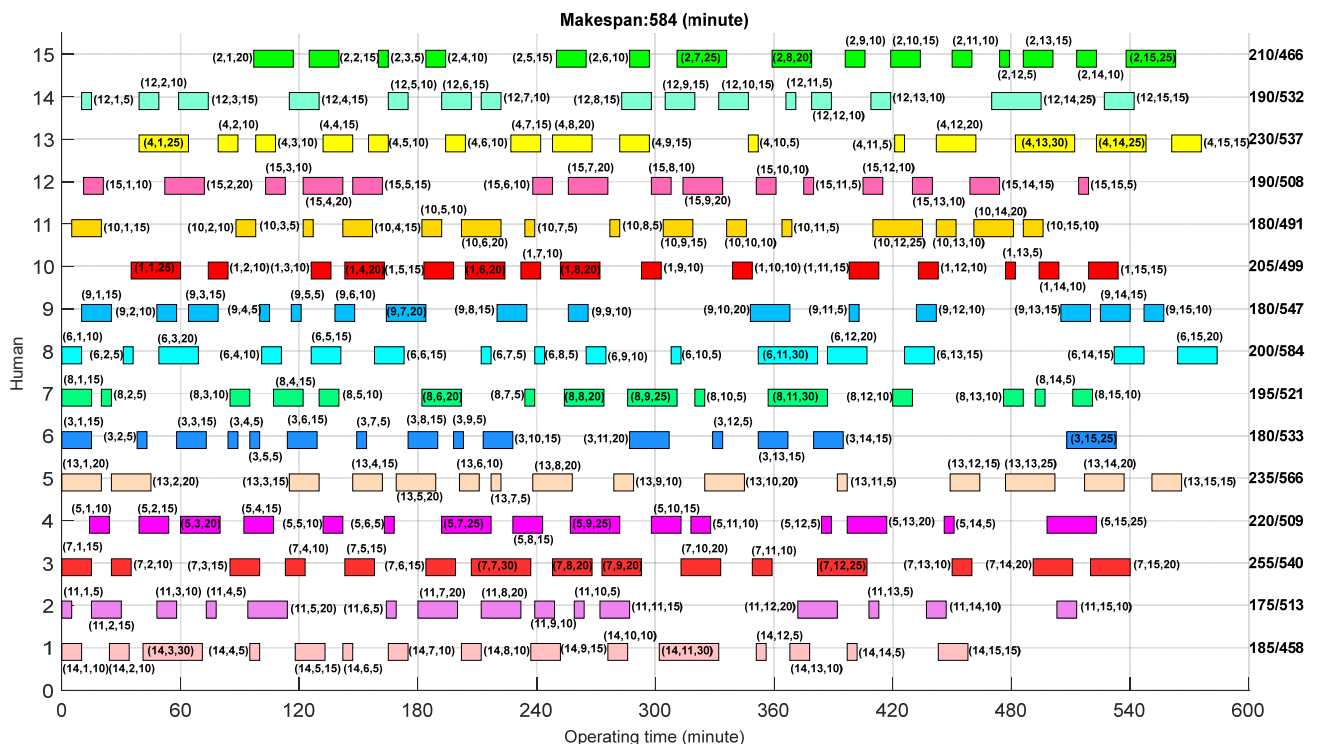


Figure 7. The Gantt chart of the humans. The different colors of progress bars indicate different jobs.

From Figures 6 and 7, it can be seen that the FJSP and the FSP are entangled and influence each other in the CHMCMP, and solving FJSP and FSP separately cannot eliminate the conflict between the two. Furthermore, according to Figures 6 and 7, the human-machine conflict time of all jobs $\{J_j\}$ can be calculated as $\sum_j t_j^{conf} = 0.0$ min through Equation (7). Hence, the benchmark results of the proposed APC-based scheduling architecture realize conflict-free human-machine collaborative manufacturing.

Figure 8 shows the convergence curves of the APC algorithm to solve the JFSP. In Figure 8, the optimal solution of the proposed APC algorithm can keep stable after about 741 iterations with a maximum iteration of 2000. This corresponds to the Gantt charts in Figures 6 and 7. The average fitness curve shows a fluctuating upward trend, reflecting that the population is searching for better feasible solutions in each iteration. The best fitness

curve shows a monotonic increase, but tends to stabilize after 741 iterations, reflecting the strong convergence ability of the population. Therefore, the APC algorithm has good global search capability and fast convergence performance.

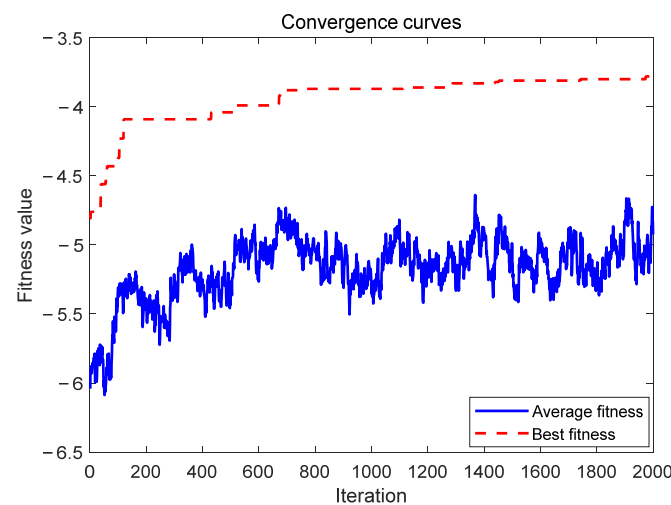


Figure 8. The convergence curves of the APC algorithm.

Table 5 analyzes the statistical results of the proposed APC-based scheduling architecture on the benchmark dataset in 200 experiments, including the makespan of all jobs, t_{Σ}^{max} , in Equation (6), the human–machine conflict time, t_j^{conf} , in Equation (7), the no-loaded processing energy consumption, e_m^{noload} , in Equation (4), etc. Table 5 shows the minimum value of makespan t_{Σ}^{max} has a 17.573% improvement compared to the maximum value, so the manufacturing performance has improved. At the same time, the benchmark results of the proposed APC-based scheduling architecture realize conflict-free human–machine collaborative manufacturing, since the human–machine conflict time of all jobs is $\sum_j t_j^{conf} = 0.0$ min. The no-loaded processing energy consumption, e_m^{noload} , of all machines has a 28.174% improvement compared to the maximum value, meaning less no-loaded processing time. Hence, the proposed APC-based scheduling architecture is efficient in solving the conflict-free human–machine collaborative manufacturing problem.

Table 5. Statistical results of benchmark testing.

Metrics	Solutions		
	Minimum	Average	Maximum
The makespan t_{Σ}^{max} of all jobs (min)	591.00	652.34	717.00
The human–machine conflict time t_j^{conf} (min)	0.00	0.00	0.00
The no-loaded processing energy consumption e_m^{noload} (kWh)	53.980	64.636	75.153
The loaded processing energy consumption e_m^{proce} (kWh)	153.267	153.267	153.267
The total energy consumption $\{e_m^{proce} + e_m^{noload}\}$ (kWh)	207.247	217.903	228.420
The transfer time $t_{m,m'}^h$ of humans (min)	2955	3001.25	3055
The unloading time $t_{jo,h,m}^{unld}$ of humans (min)	295.5	300.125	305.5
The loading time $t_{j,h,m}^{load}$ of humans (min)	591	600.25	611
The lowest transfer rate of humans (m/min)	4.00	4.00	4.00

5.3. Comparison and Discussions

Some baseline algorithms were selected for the comparative experiments, i.e., ACO [9], PSO [11,17], GWO [11], GA [13,25], and DRL [14,16,24]. The algorithm parameters in comparative experiments were set as shown in Table 6.

Table 6. Algorithm parameters in comparative experiments.

Algorithm	Population Size	Other Parameters
APC	$P = 1000$ of the APC individuals	The seeding probability $p_{seed} = 0.2$, the growing probability $p_{grow} = 0.4$, the fruiting probability $p_{fruit} = 0.8$.
ACO [9]	$P = 1000$ the of ants	The importance of heuristic factors $h = 5.0$, the pheromone volatilization factor $p = 0.1$, and the pheromone importance $phi = 1.0$.
PSO [11,17]	$P = 1000$ of the particles	The location limitation $loc = 0.5$, the speed limitation $sp = [-0.5, 0.5]$, the self-learning factor $c1 = 1.5$, and the social learning factor $c2 = 1.5$.
GWO [11]	$P = 1000$ of the gray wolves	Problem dimension $dim = 2$, and initial positions of the wolf leader (alpha), wolf deputy (beta), and wolf advisor (delta) $pos = rand(dim) \times 10^{-5}$.
GA [13,25]	$P = 1000$ of the chromosomes	The chromosome length $Lind = 32$, the crossover probability $px = 0.7$, and the mutation probability $pm = 0.01$.
DRL [14,16,24]	$P = 1000$ of the neurons in each hidden layer	A state embedding network comprised a shared stack with two hidden layers and multiple heads. The head block contained 100 ReLU units followed by $ v + 1$ units with zero-centered tanh activation.

Two scenarios are considered for comparison, i.e., different machine numbers and different human numbers, as shown in Tables 7 and 8, respectively. The number of machines increases from $m_{max} = 1, 3, 5, \dots$ to 15, and the number of humans increases from $h_{max} = 1, 3, 5, \dots$ to 15. For fair comparison, all algorithm runs on the same platform for the maximum number of iterations 200, and their values were normalized in the range of 0 and 1.

Table 7 displays the comparison results of six algorithms on the same benchmark data with different machine numbers. The first column in Table 7 shows the methods, i.e., ACO [9], PSO [11,17], GWO [11], GA [13,25], and DRL [14,16,24], where each method is on the basic version. The first line in Table 7 presents the number of machines, where there are nine cases in total, including the mean value. Based on the scheduling information in Table 7, more machines can help reduce the makespan and improve production efficiency, but increase the machine energy consumption. Although different algorithms perform differently on five metrics, the proposed APC technique is easier to search for the optimal solution in a short time. Compared with other state-of-the-art algorithms, the APC algorithm can improve the makespan by up to 11.38%, the total transfer time of humans by up to 10.39%, and the no-loaded processing energy consumption by up to 10.26% with conflict avoidance.

Table 8 reveals comparison results of six heuristic algorithms on the same benchmark data with different human numbers. The first column in Table 8 lists the methods, i.e., ACO [9], PSO [11,17], GWO [11], GA [13,25], and DRL [14,16,24], where each method is on the basic version. The first line in Table 8 lists the number of humans, where there are nine cases in total, including the mean value. Table 8 verifies that more humans can help reduce the makespan and the machine energy consumption, but increase the human walking distance. Compared with other state-of-the-art algorithms, the APC algorithm improves the makespan by up to 11.11%, the total transfer time of humans by up to 14.09%, and the no-loaded processing energy consumption by up to 12.62% with conflict avoidance. Although different algorithms perform differently on five metrics, the proposed APC technique has once again been proven to have good search performance.

Table 7. Algorithm comparison with different machine numbers.

Method	Metrics	Number of Machines								Mean	Standard Deviation
		$m_{max}=1$	$m_{max}=3$	$m_{max}=5$	$m_{max}=7$	$m_{max}=9$	$m_{max}=11$	$m_{max}=13$	$m_{max}=15$		
APC	Total makespan t_{Σ}^{cmax}	0.900	0.322	0.185	0.129	0.106	0.082	0.074	0.063	0.233	0.2963
	Total conflict time t_j^{conf}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0000
	Total transfer time $t_{m,m'}^h$	0.000	0.166	0.199	0.267	0.319	0.386	0.450	0.546	0.292	0.1666
	No-loaded energy e_m^{noload}	0.262	0.289	0.311	0.334	0.360	0.382	0.408	0.432	0.347	0.0517
	Solution time (s)	399	404	409	413	416	418	427	432	414.750	10.0519
ACO	Total makespan t_{Σ}^{cmax}	0.900	0.366	0.213	0.145	0.106	0.098	0.087	0.074	0.249	0.2902
	Total conflict time t_j^{conf}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0000
	Total transfer time $t_{m,m'}^h$	0.000	0.173	0.201	0.286	0.318	0.409	0.492	0.594	0.309	0.1809
	No-loaded energy e_m^{noload}	0.262	0.305	0.339	0.362	0.399	0.438	0.474	0.517	0.387	0.0776
	Solution time (s)	407	410	414	421	419	426	432	445	421.750	11.8488
PSO	Total makespan t_{Σ}^{cmax}	0.900	0.358	0.227	0.153	0.125	0.081	0.082	0.078	0.251	0.2892
	Total conflict time t_j^{conf}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0000
	Total transfer time $t_{m,m'}^h$	0.000	0.187	0.217	0.298	0.344	0.426	0.505	0.627	0.326	0.1909
	No-loaded energy e_m^{noload}	0.262	0.297	0.317	0.331	0.364	0.390	0.421	0.462	0.356	0.0610
	Solution time (s)	401	406	411	412	414	415	428	437	415.5	11.1596
GWO	Total makespan t_{Σ}^{cmax}	0.900	0.387	0.236	0.168	0.129	0.105	0.092	0.083	0.263	0.2850
	Total conflict time t_j^{conf}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0000
	Total transfer time $t_{m,m'}^h$	0.000	0.179	0.206	0.290	0.346	0.424	0.446	0.587	0.310	0.1789
	No-loaded energy e_m^{noload}	0.262	0.294	0.331	0.342	0.363	0.396	0.437	0.475	0.363	0.0649
	Solution time (s)	403	405	408	413	419	424	429	433	416.750	9.1604
GA	Total makespan t_{Σ}^{cmax}	0.900	0.321	0.198	0.137	0.122	0.101	0.085	0.066	0.241	0.2933
	Total conflict time t_j^{conf}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0000
	Total transfer time $t_{m,m'}^h$	0.000	0.181	0.208	0.271	0.334	0.403	0.478	0.544	0.302	0.1666
	No-loaded energy e_m^{noload}	0.262	0.292	0.314	0.347	0.361	0.385	0.408	0.433	0.350	0.0521
	Solution time (s)	402	407	412	415	418	422	431	439	418.250	11.3417
DRL	Total makespan t_{Σ}^{cmax}	0.900	0.325	0.183	0.136	0.108	0.093	0.073	0.063	0.235	0.2955
	Total conflict time t_j^{conf}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0000
	Total transfer time $t_{m,m'}^h$	0.000	0.177	0.198	0.271	0.335	0.400	0.454	0.551	0.298	0.1682
	No-loaded energy e_m^{noload}	0.262	0.290	0.313	0.346	0.357	0.379	0.410	0.431	0.349	0.0514
	Solution time (s)	1905	1964	1992	2006	2013	2097	2121	2159	2032.125	77.2898

Table 8. Algorithm comparison with different human numbers.

Method	Metrics	Number of Humans								Mean	Standard Deviation
		$h_{max} = 1$	$h_{max} = 3$	$h_{max} = 5$	$h_{max} = 7$	$h_{max} = 9$	$h_{max} = 11$	$h_{max} = 13$	$h_{max} = 15$		
APC	Total makespan t_{Σ}^{max}	0.688	0.237	0.148	0.108	0.088	0.073	0.065	0.063	0.184	0.2231
	Total conflict time t_j^{conf}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0000
	Total transfer time $t_{m,m'}^h$	0.814	0.700	0.590	0.572	0.566	0.554	0.552	0.546	0.612	0.0915
	No-loaded energy e_m^{noload}	0.900	0.562	0.495	0.466	0.450	0.439	0.433	0.432	0.522	0.1673
	Solution time (s)	402	404	409	411	419	423	429	432	416.125	9.1442
ACO	Total makespan t_{Σ}^{max}	0.781	0.274	0.146	0.129	0.100	0.073	0.076	0.074	0.207	0.2538
	Total conflict time t_j^{conf}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0000
	Total transfer time $t_{m,m'}^h$	0.918	0.807	0.715	0.649	0.623	0.619	0.605	0.594	0.691	0.1062
	No-loaded energy e_m^{noload}	0.911	0.590	0.573	0.562	0.555	0.541	0.533	0.517	0.598	0.1392
	Solution time (s)	401	405	412	413	417	425	436	445	419.250	13.5819
PSO	Total makespan t_{Σ}^{max}	0.698	0.259	0.157	0.122	0.103	0.077	0.064	0.078	0.195	0.2238
	Total conflict time t_j^{conf}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0000
	Total transfer time $t_{m,m'}^h$	0.921	0.809	0.728	0.677	0.653	0.649	0.633	0.627	0.712	0.0971
	No-loaded energy e_m^{noload}	0.906	0.561	0.553	0.547	0.538	0.499	0.486	0.462	0.569	0.1522
	Solution time (s)	402	407	414	418	423	429	433	437	420.375	10.6635
GWO	Total makespan t_{Σ}^{max}	0.694	0.255	0.166	0.127	0.087	0.081	0.079	0.083	0.197	0.2200
	Total conflict time t_j^{conf}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0000
	Total transfer time $t_{m,m'}^h$	0.919	0.829	0.729	0.642	0.617	0.601	0.593	0.587	0.690	0.1081
	No-loaded energy e_m^{noload}	0.923	0.561	0.514	0.485	0.466	0.443	0.439	0.475	0.538	0.1710
	Solution time (s)	408	413	415	416	425	432	444	433	423.250	11.0816
GA	Total makespan t_{Σ}^{max}	0.693	0.248	0.151	0.114	0.092	0.078	0.064	0.066	0.188	0.2237
	Total conflict time t_j^{conf}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0000
	Total transfer time $t_{m,m'}^h$	0.907	0.712	0.616	0.571	0.567	0.551	0.549	0.544	0.627	0.1256
	No-loaded energy e_m^{noload}	0.905	0.563	0.499	0.478	0.467	0.444	0.432	0.433	0.528	0.1675
	Solution time (s)	399	402	410	414	419	426	431	439	417.5	12.2058
DRL	Total makespan t_{Σ}^{max}	0.688	0.243	0.146	0.108	0.089	0.074	0.066	0.063	0.185	0.2229
	Total conflict time t_j^{conf}	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.0000
	Total transfer time $t_{m,m'}^h$	0.901	0.702	0.594	0.580	0.575	0.567	0.554	0.551	0.628	0.1221
	No-loaded energy e_m^{noload}	0.908	0.569	0.504	0.467	0.451	0.438	0.432	0.431	0.525	0.1697
	Solution time (s)	1907	1938	2005	2019	2074	2126	2137	2159	2045.625	77.0651

Based on the results in Tables 7 and 8, the most efficient method should be the APC algorithm, followed by DRL [14,16,24] and then GA [13,25]. There are the following reasons.

First, the proposed APC algorithm increases global search capability through random seeding, and can generate a batch of completely different random solutions in each iteration, so it is easier to obtain optimal solutions than many state-of-the-art methods. The APC algorithm enhances local search ability through the cloning fruits, and crossing fruits for growing plant individuals. This can help the APC algorithm conduct extensive searches near optimal solutions, so it not only converges quickly but also is not easily trapped in local optima too early.

Second, the APC algorithm has been optimized for computational performance with a random-pairwise single-elimination tournament system, so its linear time complexity and computational speed have advantages over many traditional state-of-the-art algorithms, i.e., GA [13,25], NSGA-II [8], and DRL [14,16,24]. The APC algorithm is more suitable for deployment on edge computing devices and embedded platforms. When the number of edge machines increases, the solution ability of the APC algorithm will also be enhanced.

Third, in manufacturing practice, conflict resolution helps improve occupational safety and ergonomics. For example, at the manufacturing site, the processing progress and remaining workload of different machines can be displayed, reminding human workers to avoid entering the machine processing site too early, thereby improving personal safety and processing efficiency.

Fourth, this article focuses on flexible job-shop scheduling and flow-shop scheduling rather than path planning. This model assumes a straight (Euclidean) path and a constant velocity, which is a simplified Manhattan or Euclidean distance metric, as shown in Equation (1). As shown in Figure 5, there may be obstacles and passage restrictions in the manufacturing workshop, which are not considered in our benchmark experiments.

6. Conclusions

This study investigated the conflict-free human-machine collaborative manufacturing problem, and our goal is to complete all manufacturing jobs while pursuing high manufacturing performance, conflict-free human-machine collaboration, and low no-load energy consumption. We introduced a job-flow-shop scheduling problem to integrate the FJSP of machine processing and the FSP of human operating, and proposed an APC-based scheduling architecture. A random-pairwise single-elimination tournament system is introduced for elite selection, with a time complexity of $O(S)$ linearly correlated with the population size (S), better than $O(S^3)$ of GA [13,25] and $O(S^2)$ of NSGA-II [8]. Our proposal contains four main steps, i.e., job analysis and manufacturing system evaluation, CHMCMP initialization, APC-based iterative search for the JFSP, and implementation of scheduling and manufacturing based on the optimal solution. Then, a medium-scale benchmark dataset was exploited according to a human-machine collaborative manufacturing scenario. Figures 6 and 7 discover that the FJSP and the FSP are entangled and influence each other in the CHMCMP, and solving FJSP and FSP separately cannot eliminate the conflict between the two. Compared with other state-of-the-art algorithms, the APC algorithm improves the makespan by up to 11.38%, the total transfer time of humans by up to 14.09%, and the no-loaded processing energy consumption by up to 12.62% with conflict avoidance.

However, due to insufficient experimental conditions, the proposed APC-based scheduling architecture was completed through simulation and not deployed in a real manufacturing system, which requires more testing and validation from peers. Additionally, our benchmark data set is a simplified Manhattan or Euclidean distance metric, which may not be consistent with the real-world manufacturing workshop with potential obstacles and passage restrictions. These need to be improved in future work.

Author Contributions: Conceptualization, Z.C.; Methodology, Z.C.; Software, X.D. and C.H.; Validation, X.D. and C.H.; Investigation, C.H., H.D. and Z.L.; Data curation, C.H., H.D. and Z.L.; Writing—original draft, X.D.; Writing—review and editing, Z.C. and X.D.; Project administration, Z.C.; Funding acquisition, Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (No. 71471102), Major Science and Technology Projects in Hubei Province of China (Grant No. 2020AEA012), and Yichang University Applied Basic Research Project in China (Grant No. A17-302-a13).

Data Availability Statement: The original contributions presented in this study are contained in the article. For further inquiries, please contact the corresponding author.

Acknowledgments: Thank you to the anonymous reviewers for their valuable feedback on this manuscript.

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

References

1. Rahman, A.; Ali, M.H.; Mahmood, M.A.; Liou, F. Industry 5.0, human-machine interface, and smart manufacturing in additive manufacturing—a recent trend. *Int. J. Adv. Manuf. Technol.* **2025**, *141*, 109–140. [\[CrossRef\]](#)
2. Rani, S.; Jining, D.; Shoukat, K.; Shoukat, M.U.; Nawaz, S.A. A human-machine interaction mechanism: Additive manufacturing for industry 5.0—design and management. *Sustainability* **2024**, *16*, 4158. [\[CrossRef\]](#)
3. Islam, M.T.; Sepanloo, K.; Woo, S.; Woo, S.H.; Son, Y.J. A review of the industry 4.0 to 5.0 transition: Exploring the intersection, challenges, and opportunities of technology and human-machine collaboration. *Machines* **2025**, *13*, 267. [\[CrossRef\]](#)
4. Bao, N.S.; Yang, Y.P.; Fan, Y.C.; Simeone, A. Optimising apparel production in Industry 5.0 using a human-centric flexible manufacturing approach. *Int. J. Adv. Manuf. Technol.* **2025**, *139*, 1881–1895. [\[CrossRef\]](#)
5. Cai, Z.; Yu, Q.; Lu, Z.; Liu, Z.; Gong, G. An artificial plant community algorithm for collision-free multi-robot aggregation. *Appl. Sci.* **2025**, *15*, 4240. [\[CrossRef\]](#)
6. Rahman, S.M.M. Trust-based modular cyber-physical-human robotic system for collaborative manufacturing: Modulating communications. *Machines* **2025**, *13*, 731. [\[CrossRef\]](#)
7. Vyavahare, S.; Naranje, V.; Badadhe, A.; Mahale, R.; Chapke, Y. Human-machine collaborative additive manufacturing: A systematic review and bibliometric analysis. *Robot. Intell. Autom.* **2025**, *45*, 646–675. [\[CrossRef\]](#)
8. Wang, M.; Tong, Y.F.; Zhuang, C.B. Knowledge-aware cell formation in matrix-structured manufacturing systems via large and small model synergistic methods. *Expert Syst. Appl.* **2026**, *303*, 130630. [\[CrossRef\]](#)
9. He, N.H.; Sahnoun, M.; Zhang, D.; Bettayeb, B. A hybrid approach using ant colony optimisation for integrated scheduling of production and transportation tasks within flexible manufacturing systems. *Comput. Oper. Res.* **2025**, *180*, 107059. [\[CrossRef\]](#)
10. Song, J.Y.; Shen, Y.P.; Wang, L.P.; Liu, C.C.; Tang, D.B.; Nie, Q.W. An efficient job insertion algorithm for hybrid human-machine collaborative flexible job shop scheduling with random job arrivals. *Electronics* **2025**, *14*, 3397. [\[CrossRef\]](#)
11. Tarek, N.; Chelloug, S.A.; Alhelaly, S.; El-Hefnawy, N.A.; Abdel-Kader, H.; Abdelatey, A. Swarm algorithms for sustainable dynamic flexible job shop rescheduling under machine breakdown in smart manufacturing plants. *PeerJ Comput. Sci.* **2025**, *11*, e3379. [\[CrossRef\]](#)
12. Wu, J.W.; Liu, Y.; Zhang, Y.N. Multi-objective evolutionary co-learning framework for energy-efficient hybrid flow-shop scheduling problem with human-machine collaboration. *Swarm Evol. Comput.* **2025**, *95*, 101932. [\[CrossRef\]](#)
13. Zuo, G.Z.; Jia, Z.Y.; Wu, Z.Y.; Shi, J.W.; Wang, G. A Q-learning guided dual population genetic algorithm for distributed permutation flow shop scheduling problem with machine having fuzzy processing efficiency. *Expert Syst. Appl.* **2025**, *285*, 127882. [\[CrossRef\]](#)
14. Li, Y.X.; Liu, Q.H.; Zhang, C.J.; Li, X.Y.; Gao, L. Graph-based dual-agent deep reinforcement learning for dynamic human-machine hybrid reconfiguration manufacturing scheduling. *IEEE Trans. Syst. Man Cybern. Syst.* **2025**, *55*, 8729–8741. [\[CrossRef\]](#)
15. Fernández-Miguel, A.; Ortiz-Marcos, S.; Jiménez-Calzado, M.; del Hoyo, A.P.F.; García-Muiña, F.E.; Settembre-Blundo, D. Agentic AI in smart manufacturing: Enabling human-centric predictive maintenance ecosystems. *Appl. Sci.* **2025**, *15*, 11414. [\[CrossRef\]](#)
16. An, J.; Zhu, H.Y.; Ping, R.Y.; Wu, S.Y.; Jin, X.L.; He, X.; Gui, X.L. CRFS: A decision conflict resolution model based on human-machine coordination in equipment manufacturing. *IEEE Trans. Comput. Soc. Syst.* **2025**, *12*, 2072–2083. [\[CrossRef\]](#)
17. Shi, Z.Q.; Chen, H.; Qu, T.; Yu, S.Y. Human-machine cooperative steering control considering mitigating human-machine conflict based on driver trust. *IEEE Trans. Hum. Mach. Syst.* **2022**, *52*, 1036–1048. [\[CrossRef\]](#)
18. Cai, Z.Y.; Jiang, S.; Dong, J.H.Z.; Tang, S.J. An artificial plant community algorithm for the accurate range-free positioning of wireless sensor networks. *Sensors* **2023**, *23*, 2804. [\[CrossRef\]](#) [\[PubMed\]](#)

19. Konstantinidis, F.K.; Myrillas, N.; Mouroutsos, S.G.; Koulouriotis, D.; Gasteratos, A. Assessment of industry 4.0 for modern manufacturing ecosystem: A systematic survey of surveys. *Machines* **2022**, *10*, 746. [\[CrossRef\]](#)
20. Habib, L.; Pacaux-Lemoine, M.P.; Berdal, Q.; Trentesaux, D. From human-human to human-machine cooperation in manufacturing 4.0. *Processes* **2021**, *9*, 1910.
21. Pizon, J.; Gola, A. Human-machine relationship-perspective and future roadmap for industry 5.0 solutions. *Machines* **2023**, *11*, 203. [\[CrossRef\]](#)
22. Abrisqueta, F.A.L.; Gonzalez, A.G.; Martinez, R.Z. Redefining human-machine collaboration: Industry 5.0 to improve safety and efficiency. *IEEE Lat. Am. Trans.* **2025**, *23*, 729–735. [\[CrossRef\]](#)
23. Simeone, A.; Fan, Y.C.; Antonelli, D.; Catalano, A.R.; Priarone, P.C.; Settineri, L. Inclusive manufacturing: A contribution to assembly processes with human-machine reciprocal learning. *CIRP Ann. Manuf. Technol.* **2024**, *73*, 5–8. [\[CrossRef\]](#)
24. Joo, T.; Jun, H.; Shin, D. Task allocation in human-machine manufacturing systems using deep reinforcement learning. *Sustainability* **2022**, *14*, 2245. [\[CrossRef\]](#)
25. Zhou, Y.M.; Jiang, J.; Shi, Q.N.; Fu, M.J.; Zhang, Y.; Chen, Y.H.; Zhou, L.F. GA-HPO PPO: A hybrid algorithm for dynamic flexible job shop scheduling. *Sensors* **2025**, *25*, 6736. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Manafi, E.; Domenech, B.; Tavakkoli-Moghaddam, R.; Ranaboldo, M. A self-learning whale optimization algorithm based on reinforcement learning for a dual-resource flexible job shop scheduling problem. *Appl. Soft Comput.* **2025**, *180*, 113436. [\[CrossRef\]](#)
27. Cai, Z.; Ma, Z.; Zuo, Z.; Xiang, Y.; Wang, M. An image edge detection algorithm based on an artificial plant community. *Appl. Sci.* **2023**, *13*, 4159. [\[CrossRef\]](#)
28. Assid, M.; Gharbi, A.; Pellerin, R. Machine learning-based dynamic production planning and control in unreliable manufacturing systems with supply disruptions. *Int. J. Prod. Res.* **2025**, *63*, 9229–9247. [\[CrossRef\]](#)
29. Wang, X.J.; Zheng, G.F.; Wu, Y.; Xiong, X.R.; Zhou, G.H.; Tolba, A.; Ning, Z.L. Automatic image annotation for human-machine interaction in industrial IoT flexible manufacturing. *IEEE Internet Things J.* **2025**, *12*, 32310–32322. [\[CrossRef\]](#)
30. Li, Y.X.; Zhou, J.L.; Yao, Y.J.; Liu, Q.H.; Li, X.Y.; Gao, L. Graph reinforcement learning based dynamic scheduling for human-machine symbiosis manufacturing considering multi-type disturbance events. *J. Ind. Inf. Integr.* **2025**, *47*, 100917. [\[CrossRef\]](#)
31. Pan, Y.X.; Xu, J. Human-machine plan conflict and conflict resolution in a visual search task. *Int. J. Hum. Comput. Stud.* **2025**, *193*, 103377. [\[CrossRef\]](#)

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