



OPEN ACCESS

EDITED BY

Kosmas Alexopoulos,
University of Patras, Greece

REVIEWED BY

Luiz Fernando Rodrigues Pinto,
Universidade Nove de Julho, Brazil
Apoorv Maheshwari,
Ernst and Young, United States

*CORRESPONDENCE

Wenqiang Zhang,
✉ zhangwq@haut.edu.cn

RECEIVED 12 November 2023

ACCEPTED 12 February 2024

PUBLISHED 28 February 2024

CITATION

Zhang W, Xiao G, Gen M, Geng H, Wang X,
Deng M and Zhang G (2024), Enhancing multi-
objective evolutionary algorithms with machine
learning for scheduling problems: recent
advances and survey.

Front. Ind. Eng. 2:1337174.

doi: 10.3389/fieng.2024.1337174

COPYRIGHT

© 2024 Zhang, Xiao, Gen, Geng, Wang, Deng
and Zhang. This is an open-access article
distributed under the terms of the [Creative
Commons Attribution License \(CC BY\)](#). The use,
distribution or reproduction in other forums is
permitted, provided the original author(s) and
the copyright owner(s) are credited and that the
original publication in this journal is cited, in
accordance with accepted academic practice.
No use, distribution or reproduction is
permitted which does not comply with these
terms.

Enhancing multi-objective evolutionary algorithms with machine learning for scheduling problems: recent advances and survey

Wenqiang Zhang^{1*}, Guanwei Xiao¹, Mitsuo Gen², Huili Geng¹,
Xiaomeng Wang¹, Miaolei Deng¹ and Guohui Zhang³

¹College of Information Science and Engineering, Henan University of Technology, Zhengzhou, Henan, China, ²Fuzzy Logic Systems Institute/Tokyo University of Science, Iizuka, Japan, ³School of Management Engineering, Zhengzhou University of Aeronautics, Zhengzhou, China

Multi-objective scheduling problems in workshops are commonly encountered challenges in the increasingly competitive market economy. These scheduling problems require a trade-off among multiple objectives such as time, energy consumption, and product quality. The importance of each optimization objective typically varies in different time periods or contexts, necessitating decision-makers to devise optimal scheduling plans accordingly. In actual production, decision-makers confront intricate multi-objective scheduling problems that demand balancing clients' requirements and corporate interests while concurrently striving to reduce production cycles and costs. In solving various problems, multi-objective evolutionary algorithms have attracted the attention of researchers and gradually become one of the mainstream methods to solve these problems. In recent years, research combining multi-objective evolutionary algorithms with machine learning technology has shown great potential, opening up new prospects for improving the performance of multi-objective evolutionary methods. This article comprehensively reviews the latest application progress of machine learning in multi-objective evolutionary algorithms for scheduling problems. We review various machine learning techniques employed for enhancing multi-objective evolutionary algorithms, particularly focusing on different types of reinforcement learning methods. Different categories of scheduling problems addressed using these methods were also discussed, including flow-shop scheduling issues, job-shop scheduling challenges, and more. Finally, we highlighted the challenges faced by the field and outlined future research directions.

KEYWORDS

flow-shop scheduling problem, machine learning, job-shop scheduling problem, reinforcement learning, optimization algorithm, multi-objective evolutionary algorithm

1 Introduction

Scheduling problems are a prevalent class of operational research problems widely applied in practical work. Their central objective is to allocate limited resources to multiple tasks within certain constraints, aiming to satisfy or optimize one or more performance indicators. The specific process of scheduling involves assigning specific tasks to designated resources and prioritizing tasks on the same resource, ultimately determining the start and end times for each task on each resource. Efficient optimization techniques and scheduling methods are key to achieve energy conservation (Gao et al., 2020), reduce consumption (Makhadmeh et al., 2019), lower emissions (Li and Wang, 2022), reduce costs (Tang et al., 2021), and improve the optimality of production systems (Kenné and Gharbi, 2000), and are the core component to improving production efficiency and economic benefits. Scheduling problems find significant applications in various domains, including production planning, supply chain management, transportation, aerospace, entertainment, healthcare, and telecommunications, among others. Consequently, research on scheduling holds paramount theoretical and practical value. The study of production scheduling problems originated in the 1950s and has attracted considerable attention from researchers worldwide due to its practical significance.

In the field of production scheduling, shop scheduling problems are the earliest and most extensively studied category. Shop scheduling refers to the process of optimizing the allocation of resources such as equipment, personnel (Luo Q. et al., 2022), and raw materials (Ramya et al., 2019) in a factory to meet production plan requirements and maximize production efficiency. Its objective is to ensure accurate execution of production plans while reducing production costs and improving efficiency. Proper shop scheduling plays a crucial role in manufacturing by ensuring smooth production processes, minimizing waste and wait times (Koulamas and Kyparisis, 2021), and improving production quality and efficiency (Zhao et al., 2021a). Common types of shop scheduling include flow-shop scheduling for continuous production processes and job-shop scheduling for discrete production processes, with variations such as parallel machine scheduling (Lei and Liu, 2020), hybrid shop scheduling (Botta-Genoulaz, 2000), and flexible shop scheduling (Li X. et al., 2022), each having their unique characteristics and scheduling algorithms that companies can choose based on their specific needs to achieve optimal production benefits. Figure 1 illustrates the classification of shop scheduling problems.

Several classic variants, such as the permutation flow-shop scheduling problem, hybrid flow-shop scheduling problem, job-shop scheduling problem, and flexible job-shop scheduling problem, among others, are encompassed by the shop scheduling problem. Building upon these classical problems, numerous other scheduling problems can be derived, such as the flexible job-shop scheduling problem considering composite processing limitations and the distributed flexible job-shop scheduling problem. Real-world shop scheduling can serve as a research problem, and researchers can tackle these problems by designing efficient scheduling algorithms.

Traditional job scheduling faces challenges in meeting market demands. Its main limitation lies in excessive focus on a single optimization goal, typically minimizing completion time (Ahmadian et al., 2021), while overlooking other crucial factors. Concerning production indicators, the emphasis solely on completion time (Umam et al., 2022) neglects considerations such as machine utilization (Abualigah and Diabat, 2021) and total processing time (Dai et al., 2019), making scheduling less adaptable to diverse market needs. Additionally, rigid scheduling strategies in traditional systems struggle to flexibly respond to market changes and demand fluctuations, resulting in delays or an inability to meet new market requirements. To better address market demands, current research is gradually shifting towards multi-objective optimization problems (MOPs), highlighting the importance of considering factors like machine utilization, delivery time (Liu et al., 2021), and inventory costs (San-José et al., 2019). Simultaneously, introducing more flexible and intelligent scheduling strategies and technologies becomes crucial for better adaptation to the ever-changing market environment. Therefore, research on MOPs in the field of scheduling holds significant engineering importance (Lei, 2009).

For most MOPs, the objectives are often in conflict. Consequently, optimizing all objectives simultaneously to their respective optimal values is unattainable. An array of compromise solutions among different objectives is referred to as the Pareto optimal set. The expression of MOP is (Cheung et al., 2016):

$$\begin{aligned} \min. F(t) &= (f_1(t), f_2(t), \dots, f_m(t))^T \\ \text{s. t. } t &\in \Omega \end{aligned} \quad (1)$$

where $t = (t_1, t_2, \dots, t_n) \in \Omega$ is a decision vector and Ω is the decision space. $F(t)$ consists of m objective functions (Li and Wang, 2022). The objectives in Eq. 1 often manifest mutual conflicts (Gu and Wang, 2020), where the optimization of one objective tends to deteriorate another objective. To address this issue, Edgeworth and Pareto (Stadler, 1979) introduced the concept of Pareto optimality as a means to balance these objectives and attain relatively superior solutions. Presented below are fundamental definitions related to Pareto optimality (Voorneveld, 2003; Deb and Gupta, 2005; Gen et al., 2008; Zitzler et al., 2008).

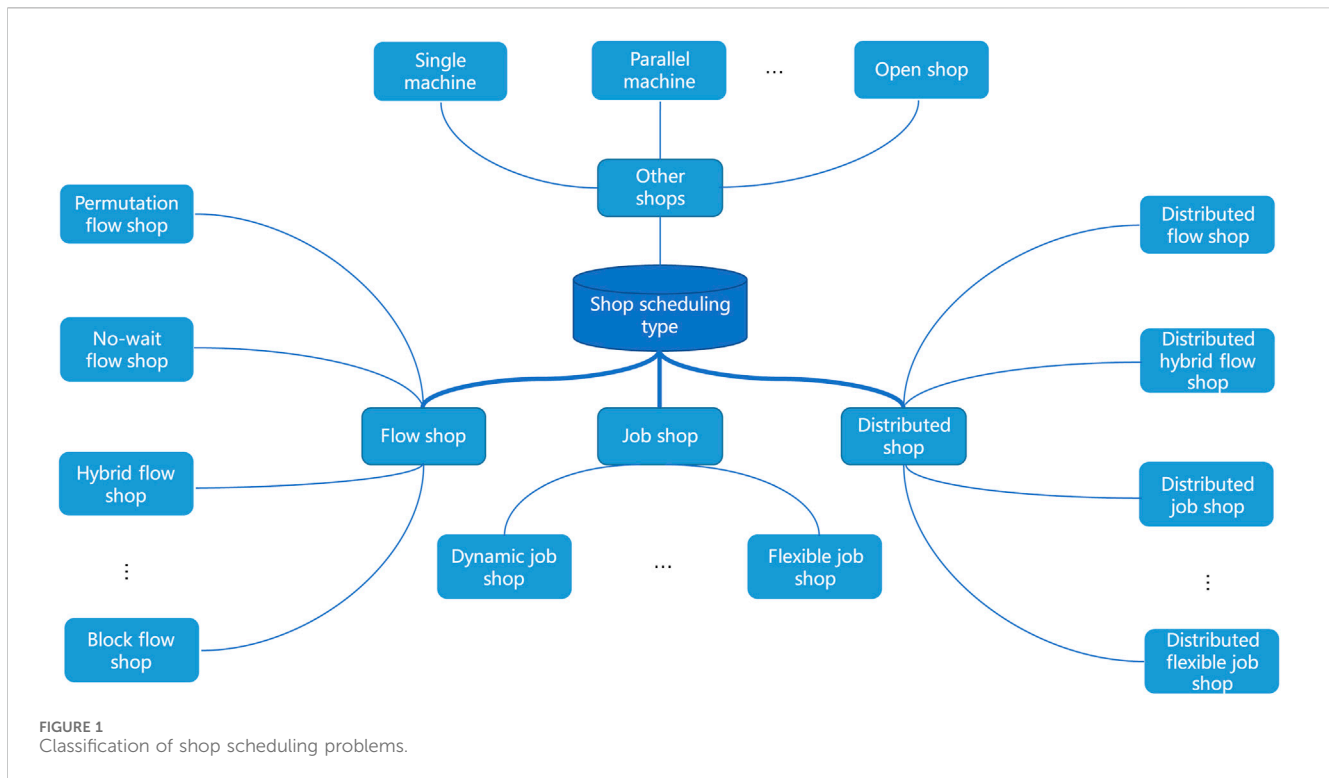
Definition 1. Pareto Dominance: If the vector $u = (u_1, u_2, \dots, u_m)^T$ is better than the vector $v = (v_1, v_2, \dots, v_m)^T$, iff $\forall i \in \{1, \dots, m\}$, $u_i \leq v_i$ and $u \neq v$. It is said that u dominates v , denoted as $u < v$.

Definition 2. Pareto optimal solutions: There does not exist any feasible solution $x \in D$ where all the objective values of $f(x)$ are not worse than the objective values of $f(x^*)$ and at least one objective value is strictly better than the corresponding objective value of $f(x^*)$. The mathematical expression is represented in Eq. 2.

$$x^* \in D, \nexists x \in D, f(x) \leq f(x^*) \quad (2)$$

Definition 3. Pareto Set: The amalgamation of all Pareto optimal solutions is commonly denoted as the Pareto set (PS). The mathematical expression is represented in Eq. 3.

$$PS = \{x_{ps} \in D \mid \nexists x \in D, x < x_{ps}\} \quad (3)$$



Definition 4. Pareto front: The assemblage of objective value vectors corresponding to each solution within the PS is referred to as the Pareto front (PF). The mathematical expression is represented in Eq. 4.

$$PF = \{F(\mathbf{x}) \in \Lambda \mid \mathbf{x} \in PS\} \quad (4)$$

Scheduling is a highly intricate and multi-dimensional discrete optimization challenge encompassing the optimal arrangement of jobs and operations, and the assignment of machines. It can be formulated a function, denoted as follows:

$$f(\mathbf{x}^*, \mathbf{y}^*) = \min_{\mathbf{x} \in \Omega} f(\mathbf{x}, \mathbf{y}) \quad (5)$$

In Eq. 5, where x represents an n -dimensional vector, $x = (x_1, x_2, \dots, x_n)$, representing the prioritized order of n jobs or operations. Similarly, y represents an n -dimensional vector, $y = (y_1, y_2, \dots, y_n)$, indicating the allocation status of machines. The sequence of jobs determines the order of machine allocation (Lin and Gen, 2018).

Numerous multi-objective optimization approaches have been proposed, such as: heuristic approaches (Fattahi et al., 2007), multi-objective evolutionary algorithms (MOEAs) (Gen and Cheng, 2000; Zitzler et al., 2001; Deb et al., 2002; Espejo et al., 2009; Deb, 2011; Zhang et al., 2014), machine learning-based approaches (Brik et al., 2019), etc.

Among them, MOEAs is a population-based evolutionary algorithm (EA) that generates new solutions by evaluating objective functions and using selection and recombination operators. The methods of MOEAs can generally be categorized into Pareto-dominance-based MOEAs (Srinivas and Deb, 1994), decomposition-based MOEAs (MOEA/D) (Zhang and Li, 2007), and indicator-based MOEAs (Zitzler and Künzli, 2004).

1. Pareto-dominance-based MOEAs: This method is based on Pareto dominance relationship, and by maintaining a set of non dominated solutions, the algorithm can iteratively evolve pareto optimal solutions that achieve compromise on multiple objectives. This algorithm adopts non dominated sorting and selection mechanisms, making the selected solutions more likely to make significant progress on the PF in each generation. The approach have been proven effective in solving MOPs with two or three objectives. However, when facing high-dimensional MOPs, many Pareto-dominance-based MOEAs may encounter selection pressure issues, making it challenging for them to evolve effectively towards the true PF. Due to the exponential growth of non-dominated solutions with an increasing number of objectives during the early iterations, the aforementioned issue arises. Consequently, the convergence-maintaining mechanism based solely on Pareto dominance (e.g., NSGA-III (Yi et al., 2020)) loses the pressure to drive the population towards the PF. Hence, relying solely on the criterion of Pareto dominance cannot effectively differentiate the convergence level of individuals. In light of this, the dual archive algorithm (Liu et al., 2019) boosts solution convergence, while the enhanced dual archive algorithm (Li et al., 2014) addresses the complexities associated with high-dimensional multi-objective problems. To achieve a harmonious equilibrium between population convergence and diversity, Wang and Tong (Wang and Tong, 2020) put forth a dimension-convergent MOEA.
2. Decomposition-based MOEAs: A novel approach called MOEA based on decomposition (MOEA/D). By leveraging

decomposition, MOEA/D offers a promising avenue for multi-objective evolutionary optimization. It facilitates the efficient optimization of multiple objectives by breaking them down into simpler subproblems and leveraging inter-subproblem interactions. This approach has the potential to enhance the scalability and computational efficiency of multiobjective evolutionary optimization algorithms (Trivedi et al., 2016). In MOEA/D, the evolutionary operators are highly sensitive to the characteristics of the problem, particularly in different search stages where the characteristics often exhibit diversity. However, in existing integrated approaches, the same evolutionary operators are applied to all subproblems/subspaces. For complex MOPs, the characteristics of subproblems/subspaces vary, which significantly weakens their distribution. Furthermore, the distribution of high-dimensional MOPs cannot be effectively guaranteed, resulting in suboptimal performance.

- Indicator-based MOEAs: A method of evaluating and comparing solution sets generated by algorithms by designing and utilizing performance indicators. These indicators typically include convergence, diversity, balance, etc., aimed at helping optimization algorithms better understand and improve their performance when dealing with multi-objective problems. By introducing these indicators, indicator-based MOEAs provide researchers with an effective way to quantify and compare the performance of multi-objective optimization algorithms. Among the indicator-based MOEAs (Beume et al., 2007), the *s*-metric selection evolutionary multi-objective algorithm stands out as a widely recognized and representative method. Furthermore, there are other notable approaches in this domain, such as MOEA whose environmental selection is based on an enhanced inverted generational distance indicator with noncontributing solution detection (IGD-NS) (Tian et al., 2016) and MOEA methods based on IGD-NS (Tian et al., 2017), etc.

Despite the tremendous success of existing MOEAs in addressing multi-objective problems, they are predominantly focused on unconstrained MOPs, and the performance of most MOEAs drastically deteriorates when confronted with large-scale MOPs. The emergence of machine learning (ML) has offered potential to enhance the performance of MOEAs. Early investigations into the integration of ML and EAs were initiated by Goldberg and Holland in 1988 (Booker et al., 1989).

This paper primarily emphasizes recent advancements in combining ML, particularly emerging reinforcement learning (RL), with MOEAs for solving multi-objective shop scheduling problems. In Section 2, we provide an in-depth and comprehensive discussion on the background knowledge of scheduling problems and RL. Subsequently, in Section 3, we investigate the detailed application of traditional MOEAs and RL in scheduling problems. Following that, in Section 4, we present the latest progress in the application of enhanced MOEAs in the field of scheduling. Finally, in Section 5, we summarize the paper and propose potential avenues for future research.

2 Shop scheduling problems

2.1 Flow-shop scheduling problem

The flow-shop scheduling problem (FSP) is an NP-hard optimization problem (Berlińska and Przybylski, 2021), and researchers typically employ nature-inspired EA or other meta-heuristic algorithms (Goli et al., 2023) to solve it. These algorithms often leverage inspiration from natural evolution processes, collective behavior, and physical laws to enhance search efficiency, using a variety of potential options to preserve population diversity and prevent being stuck in local optima (Katoch et al., 2021).

The FSP refers to the task of scheduling the sequence and timing of different operations in a manufacturing shop consisting of multiple machines. The objective is to achieve the optimal production efficiency and minimize the production cost.

The problem can be formulated using mathematical models that include the following elements.

- Individually assigned to a specific machine, each job is executed solely once on that designated machine.
- Each machine has a limited capacity, therefore only one job may be executed concurrently on each machine at any one time.
- Every job can only be finished on a single machine; it cannot be changed to another.
- Jobs are non-preemptive, once they commence running on a machine, they cannot be paused or transferred to other machines midway.
- Regardless of changes to the schedule, the processing time for every job on every machine is fixed.
- The order in which the machines are used is predetermined, and each job is completed on the machines in the designated sequence.

The following is a mathematical description of the FSP.

$$c(i, j) = t(i, j) \quad (6)$$

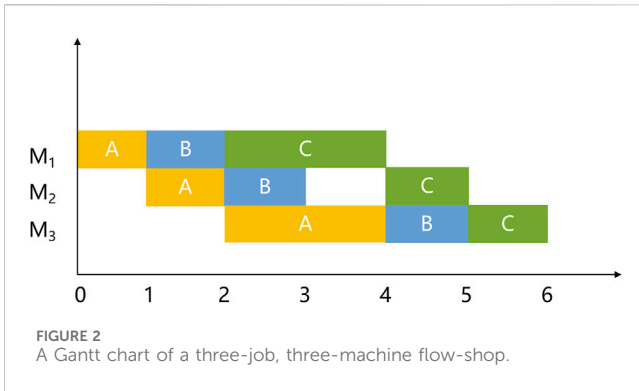
$$c(i, j) = c(i, j - 1) + t(i, j) \quad (7)$$

$$c(i, j) = c(i - 1, j) + t(i, j) \quad (8)$$

$$c(i, j) = \max(c(i - 1, j), c(i, j - 1)) + t(i, j) \quad (9)$$

$$MakeSpan = \min(c(n, m)) \quad (10)$$

Where the completion time of task *i* on machine *j* equals its own processing time, as shown by Eq. 6. Eq. 7 denotes the completion time of task *i* on machine *j* as the sum of the completion time of job *i* on the preceding machine and the processing time for the current operation. The completion time of task *i* on machine *j* is determined by adding the cumulative completion time of the previous job on machine *j* and the processing time necessary for the current operation, as stated in Eq. 8. As demonstrated by Eq. 9, the completion time of job *i* on machine *j* can be represented as the sum of the processing time for the current operation and the maximum value obtained by comparing the completion time of the *i* - 1st job on machine *j* with the completion time of job *i* on machine *j* - 1. Eq. 10 demonstrates that the makespan and overall



completion time of the complete assembly line are contingent upon the finalization time of the last task n on the last machine m .

Typically, FSPs are categorized based on the sequence of operations and the attributes of the processing machines into various classifications, including permutation flow-shop scheduling Problem (PFSP) (de Fátima Morais et al., 2022), no-wait flow-shop scheduling Problem (NWFSP) (Zhao et al., 2019), hybrid flow-shop scheduling Problem (HFSP) (Fernandez-Viagas et al., 2019), no-idle FSP (Zhou et al., 2014; Zhang W. et al., 2021), and block FSP (Miyata and Nagano, 2019), etc. Figure 2 represents a Gantt chart for a flow-shop.

2.2 Hybrid flow-shop scheduling problem

The HFSP is an important production planning problem widely used in different manufacturing fields. Researchers use various algorithms to solve HFSP problems, considering different goals and process constraints.

The scheduling challenge in a hybrid-flow shop setting can be broadly defined as handling n jobs that follow similar processing routes across m processing stages. The presence of at least one parallel machine, and preferably two or more, is essential. Moreover, it is imperative that these parallel machines exhibit uniform processing capabilities.

With the provided processing times for all jobs, the primary goal of this issue is to ascertain the sequential arrangement of the n jobs both before and after each processing stage. Additionally, it involves allocating devices during each stage to minimize the total completion time of all jobs. The assumptions underlying the problem model are outlined below.

- Each job can be processed on only one machine at the same time.
- The same machine can only process one job at a time.
- The processing duration for each job on a specific device is predetermined.
- Once initiated, the processing procedure remains uninterrupted.
- Sequential dependencies are present solely within the operations of a given job, with no inter-operation constraints across distinct jobs.

- Unforeseen factors, such as machine malfunctions, are not taken into consideration.
- Any additional time incurred from transitioning between jobs on the same machine is not factored into the analysis.

2.2.1 Decision variables

$X_{q,j,i}$: The feasibility of processing job j on machine q during operation i . A value of $X_{q,j,i} = 1$ indicates the feasibility of processing, while $X_{q,j,i} = 0$ signifies the impossibility of processing;

Y_{i,j_1,j_2} : The precedence relationship between job j_1 and j_2 in operation i . A value of $Y_{i,j_1,j_2} = 1$ indicates that j_1 takes precedence, while $Y_{i,j_1,j_2} = 0$ indicates that j_2 takes precedence.

The following is a mathematical representation of the HFSP.

$$\min .T_{\max} \tag{11}$$

$$T_{\max} = \max\{F_{j_i}\} \tag{12}$$

$$F_{j_i} = S_{j_i} + T_{j_i} \tag{13}$$

$$\text{s. t. } \sum_{q=1}^{B_i} X_{q,j_i} = 1 \tag{14}$$

$$S_{j_{(i+1)}} - S_{j_i} \geq T_{j_i} \tag{15}$$

$$Y_{i,j_1,j_2} + Y_{i,j_2,j_1} \geq 1 \quad (j_1, j_2 = 1, \dots, n; j_1 \neq j_2) \tag{16}$$

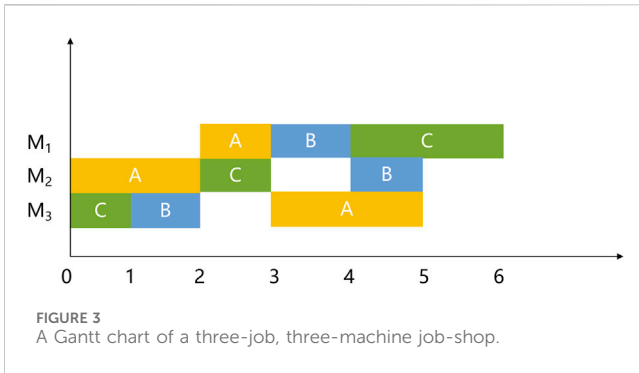
$$S_{j_i} - F_{j_2} + C \times (3 - Y_{i,j_2,j_1} - X_{j_1,q_i} - X_{j_2,q_i}) \geq 0 \tag{17}$$

Eq. 11 represents the minimization of the overall completion time. Eq. 12 signifies the overall completion time for all jobs, which is the maximum completion time within the set of completion times for each job j at operation i . Eq. 13 represents the processing time for any job at any operation. Eq. 14 indicates that any operation for any job is performed by only one machine within that operation. Eq. 15 states that any job must complete the processing of the current operation before moving on to the next one. Eqs 16, 17, when combined, express that any machine within any operation cannot process more than one job at the same time.

The actual processing environment of enterprises is complex and ever-changing. The traditional formulation of the HFSP is no longer adequate to address the diverse requirements of enterprises. In response to specific demands, considering variations in constraints, characteristics of machining tasks, and the quantity of jobs, diverse extensions of the HFSP have surfaced. Examples include the multi-stage HFSP (Hoogeveen et al., 1996), reentrant HFSP (Dugardin et al., 2010), no-wait HFSP (Engin and Güçlü, 2018), among others.

2.3 Job-shop scheduling problem

The job-shop scheduling problem (JSP) is a fundamental combinatorial optimization issue in the field of operational research and management science (Xiong et al., 2022). JSP is commonly characterized as the efficient scheduling of n jobs on m machines, optimizing performance indicators under the assumptions of known processing techniques, machine sequences, and processing times for each job. When addressing JSPs, the following fundamental constraints need to be taken into consideration.



- Every job is comprised of multiple operations that need to be accomplished.
- A job can only be executed on a single machine at any given time.
- The predetermined processing time for each operation on each machine has been determined.
- All jobs are accessible and ready for processing from the start
- Each machine can handle only one operation at a time.
- All operations must be completed on the same group of machines.

In this paper, JSP as an integer programming model will be represented.

$$\min. \max_{1 \leq i \leq n, 1 \leq k \leq m} \{c_{ik}\} \tag{18}$$

$$\text{s. t. } c_{ik} - s_{ik} + M(1 - a_{ihk}) \geq c_{ih}, \forall i, h \tag{19}$$

$$x_{ijk} > 0, c_{jk} - c_{ik} + M(1 - x_{ijk}) \geq s_{jk}, \forall j, k \tag{20}$$

$$c_{ik} \geq 0, \forall i, k \tag{21}$$

In Eq. 18, Where n denotes the number of jobs, m represents the number of machines. In Eqs 19–21, s_{ik} and c_{ik} indicate the processing time and completion time of job i on machine k . $i, j \in \{1, 2, \dots, n\}$; $h, k \in \{1, 2, \dots, m\}$. Here, M signifies a sufficiently large positive value. The binary variable a_{ihk} with values 0 and 1 denotes the precedence relationship between jobs i and h on machine k ; a_{ihk} being 0 implies job i is processed before job h on machine k , while a_{ihk} being 1 signifies the reverse. Similarly, the binary variable x_{ijk} , also with values 0 and 1, indicates the sequencing of jobs i and j on machine k ; x_{ijk} being 0 denotes job i is scheduled before job j on machine k , whereas x_{ijk} being 1 implies the opposite. The objective of the JSP is to determine the optimal schedule of operations for all jobs that minimizes the makespan (C_{max}), which is defined as the maximum completion time among all jobs.

JSP is a highly complex problem due to its combinatorial nature and the large number of possible solutions. Finding an optimal solution to JSP is known to be NP-hard, which means that it is computationally intractable for large instances. Therefore, many heuristic and meta-heuristic algorithms have been proposed to find near-optimal solutions to JSP (Tsujimura et al., 1995; Cheng et al., 1996; 1999; Tavakkoli-Moghaddam et al., 2005; Hao et al., 2017).

In addition, JSP has various forms and variants, including dynamic job-shop scheduling problem (Ramasesh, 1990; Kundakci and Kulak, 2016; Mohan et al., 2019), flexible job-shop

scheduling problem (FJSP) (Pezzella et al., 2008; Zhang G. et al., 2011; Xie et al., 2019), distributed job-shop scheduling problem (DJSP) (De Giovanni and Pezzella, 2010; Meng et al., 2020; Şahman, 2021), and JSP with consideration of energy and environmental factors (Jiang et al., 2019; Wang et al., 2020), etc. Figure 3 represents a Gantt chart for a job-shop.

2.4 Flexible job-shop scheduling problem

The FJSP is a variant of the classical JSP that has gained significant attention in the field of research on JSP. In traditional JSP, each process must be completed on a specific machine tool, while flexible job shop scheduling allows each process to be carried out on multiple different machine tools, and the time required for the execution of the operation exhibits variation contingent upon the performance of the machine tool. This extension brings more problem complexity while reducing machine constraints and expanding the search space for feasible solutions.

The issue may be delineated in the following manner: The production system is comprised of a total of m machines and n distinct sorts of jobs. Every job comprises one or more operations, with a predetermined sequence of these operations. Every operation can be processed on multiple different machines, each with its own performance and processing speed. The scheduling objective is to choose the most appropriate machine for each operation, establish the optimal sequence and start time for each job operation on each machine, and optimize the system’s performance indicators. Additionally, it is essential that the following limitations be duly met.

- Initial Setup: At the beginning of the task, all jobs and equipment are ready and ready to start working immediately.
- Machine Constraints: Each machine can only perform one operation at a time, meaning only one machining task is allowed to take place simultaneously.
- Operation Selection: Each operation can be processed on multiple machines, but only one machine can be selected for processing at a time.
- Continuous Processing: Once processing begins, it cannot be interrupted and must proceed continuously without pauses or interruptions.
- Operation Sequence: Each job must follow the specified sequence of operations, with the next operation starting only after the completion of the previous one.
- Job Priorities: All jobs have equal processing priorities, ensuring the fair allocation of processing times for all jobs and preventing prolonged waiting times for any particular job.

In 1990, Brucker and Schlie (Brucker and Schlie, 1990) first introduced the FJSP, and its solving methods can be broadly categorized into three main classes: exact algorithms (Woeginger, 2003), heuristic algorithms, and intelligent optimization algorithms (Li W. et al., 2021).

2.4.1 Exact algorithms

Exact algorithms are a category of algorithms specifically designed to find the optimal solution to a given problem. These algorithms are recognized for their systematic and deterministic

approaches, making them particularly suitable for addressing intricate combinatorial optimization problems. Some of the main exact algorithms include branch and bound (Narendra and Fukunaga, 1977), cutting-plane methods (Kelley, 1960), integer linear programming (Schrijver, 1998; Moon et al., 2004), mixed-integer linear programming (Floudas and Lin, 2005), and so on. Torabi et al. (Torabi et al., 2005) proposed a novel mixed integer nonlinear programming method for solving the common cycle multi product batch scheduling problem in flexible job shops. Emami et al. (Emami et al., 2016) developed an innovative Lagrangian relaxation algorithm for handling simultaneous order acceptance and scheduling problems in non identical parallel machine environments. Their method uses the cutting plane method to dynamically update Lagrange multipliers to improve the optimization efficiency of the problem.

The primary objective of exact algorithms is to determine the best solution to a given issue, as opposed to providing an approximation. In instances when issue sizes are modest, the processing time necessary to get the best solution by use of precise methods is often deemed satisfactory. Nevertheless, when the size of the issue rises, the computing complexity experiences a substantial increase. Therefore, the task of identifying the most efficient solution within a feasible time is a significant challenge and complexity, depending on the current computational resources, may become infeasible (Lin and Gen, 2018).

2.4.2 Heuristic algorithms

A heuristic algorithm is actually a set of rules with guiding properties, which are used to guide the algorithm in finding directions to solve problems in the search space. Under the guidance of these rules, the algorithm can find the optimal solution to the problem, but it may not necessarily be the optimal solution. At present, many heuristic algorithms have been widely applied to handle FJSP, with most of them using heuristic rules as the main method, which usually come from practical scheduling problems.

Scrich et al. (Scrich et al., 2004) proposed two heuristic algorithms based on taboo search when solving FJSP with the optimization objective of minimizing total delay: a hierarchical process and a multiple initiation process. The core idea of these algorithms is to generate initial solutions using scheduling rules, and then search in the critical path neighborhood represented by the disjunction graph to improve the solution. In addition, Shanker et al. (Shanker and Tzen, 1985) compared these heuristic algorithms with precise mixed integer programming and proposed a simulation model for system performance evaluation by combining four scheduling rules: first In first out, shortest processing, longest processing time, and most operations remaining first.

Heuristic algorithms have the ability to quickly respond and generate feasible scheduling solutions when solving specific problems. More importantly, the solving complexity of these algorithms remains low sensitivity as the problem size increases, making them still effective in dealing with large-scale problems. Therefore, heuristic algorithms play an important role in solving FJSP.

2.4.3 Intelligent optimization algorithms

Intelligent optimization algorithms are important research methods for solving FJSP, which can be divided into EA and swarm intelligence algorithms. The proposal of EA was inspired

by natural evolution, and classic EA includes genetic algorithms (Gao et al., 2007; Gen et al., 2009), evolutionary strategies (Beyer and Schwefel, 2002), genetic programming (Langdon and Poli, 2013), and differential evolution (Price, 2013). Among these methods, genetic algorithm (GA) has the most extensive and in-depth research and application.

Gao et al. (Gao et al., 2007) adopts a hybrid method of a new GA and an innovative local search process (bottleneck shifting) to optimize the three objectives in FJSP: minimizing C_{max} , minimizing maximum machine workload, and minimizing total workload. The author verified the performance of their proposed method by conducting numerical experiments on a large number of representative problems to verify its effectiveness in solving FJSP. In addition, Gao et al. (Gao et al., 2008) have developed a hybrid GA to solve FJSP with three objectives, which uses two vectors to represent the solution. They adopt advanced crossover and mutation operations to adapt to the characteristics of special chromosome structures and problems. In order to enhance the search ability, GA individuals first improve through variable neighborhood descent. As applications based on the FJSP model, Chou et al. (Chou et al., 2014) reported a case study of multiobjective hybrid genetic algorithm for thin film transistor liquid crystal display module assembly scheduling. Jamrus et al. (Jamrus et al., 2015) proposed a multistage production distribution under uncertainty demands by discrete Particle Swarm Optimization approaches with extended priority based-hybrid genetic algorithm. Chamnanlor et al. (Chamnanlor et al., 2017) reported embedding ant system in genetic algorithm for re-entrant hybrid flow shop scheduling problems with time window constraints.

The swarm intelligence algorithm is an excellent method for solving FJSP. This type of algorithm has simple principles, strong robustness, and can find almost the best solution in a relatively short time, and is easy to implement.

2.5 Distributed shop scheduling problem

The distributed shop scheduling problem (DSSP) is a study on optimizing the scheduling of job allocation between different factories and processing sequences within each factory in a distributed manufacturing context (Toptal and Sabuncuoglu, 2010). DSSP typically involves collaborative production among multiple factories or cooperative production among different companies, aiming to maximize production efficiency by optimizing scheduling indicators. DSSP is a problem involving multiple shops and multiple tasks, where each task requires processing in one or more shops. The objective of this problem is to complete all tasks as much as possible within a given time and minimize the total cost or maximize the total profit. Combining the traditional JSP, this paper presents the model of DJSP as follows.

- Workshops and their internal manufacturing resources are distributed at different geographical locations, interconnected through a network to facilitate the exchange and sharing of manufacturing task and process information.
- Each workshop and its internal manufacturing resources possess independent processing capabilities and intelligent decision-making capabilities. The processing capacities of

various machine devices may be uniform or diverse, with fixed categories, quantities, and capabilities.

- Strict processing sequence constraints exist between successive operations of the same jobs, necessitating sequential processing of each operation according to the technological order.
- Different operations of the same jobs task must be completed within the same workshop.
- During the shop scheduling process, jobs can be dynamically added, and global tasks may undergo reallocation and rescheduling based on demand.

The notations of mathematical model of DSSP is expressed as follows.

2.5.1 Indices

- j, h : Index of job, ($j, h = 0, 1, 2, \dots, n$), where job 0 is a virtual job;
- i, l : Index of machine, ($i, l = 1, 2, \dots, m$);
- k : Index of factory, ($k = 1, 2, \dots, f$).

2.5.2 Parameters

- n : The number of jobs;
- m : The number of machines;
- f : The number of factories;
- $p_{j,i}$: Processing time of job j on machine i ;
- $S_{j,i}$: Start time of job j on machine i ;
- A : A positive infinity.

2.5.3 Decision variables

- $X_{h,j,i,k}$: 1 when job j is processed on machine i in factory k after job h , otherwise 0;
- $a_{j,l,i}$: 1 when job j is processed on machine i after completion on machine l , otherwise 0.

2.5.4 Mathematical model

The mixed-integer linear programming model (MILP) of DSSP with minimization of C_{max} is formulated as follows.

$$\min. C_{max} \tag{22}$$

$$\text{s. t. } \sum_{h=0, h \neq j}^n \sum_{k=1}^f X_{h,j,1,k}, \forall j \tag{23}$$

$$\sum_{h=0, h \neq j}^n X_{h,j,i,k} = \sum_{h=0, h \neq j}^n X_{h,j,1,k}, \forall j, i > 1, k \tag{24}$$

$$\sum_{j=1, j \neq h}^n X_{h,j,i,k} \leq \sum_{j=0, j \neq h}^n X_{h,j,i,k}, \forall h > 1, i, k \tag{25}$$

$$\sum_{j=1}^n X_{0,j,i,k} = 1, \forall i, k \tag{26}$$

$$\sum_{j=1}^n (X_{h,j,i,k} + X_{j,h,i,k}) \leq 1, \forall j > n, h > j, i \tag{27}$$

$$S_{j,i} \geq S_{j,l} + p_{j,l}, \forall j, i, l \neq i \mid a_{j,l,i} = 1 \tag{28}$$

$$S_{j,i} \geq S_{h,i} + p_{h,i} - A(1 - X_{h,j,i,k}), \forall h > 0, j \neq h, i, k \tag{29}$$

$$C_{max} \geq S_{j,i} + p_{j,i}, \forall i \tag{30}$$

$$S_{j,i} \geq 0, \forall k, i \tag{31}$$

$$X_{h,j,i,k} = \{0, 1\}, \forall j, h \neq j, i, k \tag{32}$$

Eq. 22 indicates that the objective is to minimize the total time of the manufacturing process. Eq. 23 ensures that each job can only be assigned to one factory. Eqs 24, 25 ensure that all processing operations for a job are carried out in the same factory. Eq. 26 represents the initial operation for each job on each machine. Eq. 27 ensures that each operation can only have one adjacent operation. Eq. 28 indicates that the next operation for the same job cannot commence until the preceding operation is completed. Eq. 29 ensures that each machine must be in a non-occupied state before commencing processing. Eq. 30 defines the C_{max} of the problem. Eq. 31 ensures that the start time for processing is a non-negative value. Eq. 32 defines binary variables used to represent the status of each operation.

DSSP is an extension of the shop scheduling problem that considers collaboration and resource sharing among multiple factories and processing centers. Common models include distributed parallel machine scheduling (Lei and Liu, 2020), distributed flow-shop scheduling problem (Han X. et al., 2021), DJSP (Şahman, 2021), distributed assembly shop scheduling problem (Zhao et al., 2021b), etc. To improve the efficiency and quality of problem-solving, researchers have proposed various optimization algorithms and methods (Li et al., 2022b; Lei and Su, 2023; Song et al., 2023; Ying et al., 2023; Yue et al., 2023). Figure 4 depicts a schematic diagram of a distributed shop. Figure 5 shows the distribution of different types of shops in the shop scheduling problems collected in this paper.

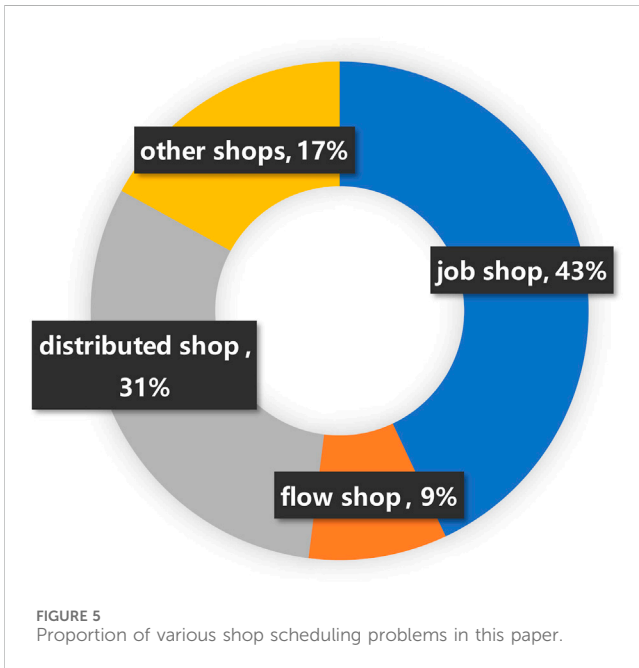
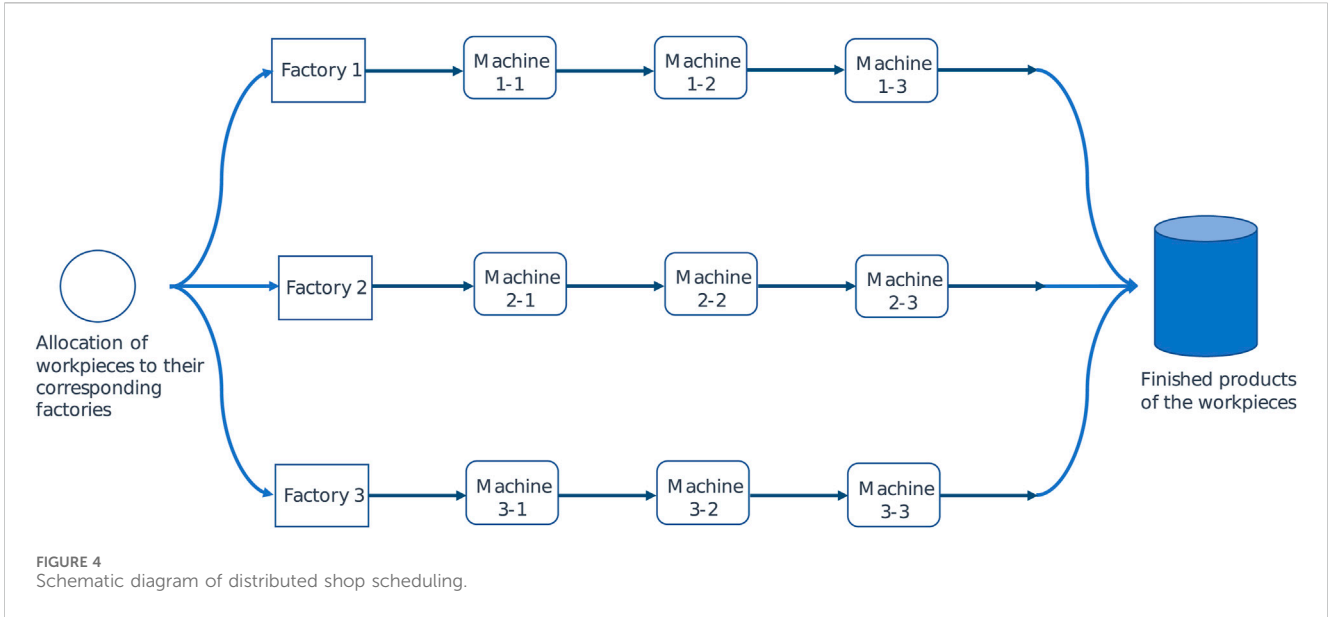
2.6 Distributed hybrid flow-shop scheduling problem

As the manufacturing industry shifts from a single factory to a multi factory model, distributed scheduling issues have become the focus. Among them, there is relatively little research on the distributed hybrid flow shop scheduling problem (DHFSP). Some studies have proposed algorithms to solve DHFSP, taking into account factors such as multiprocessor tasks (Ying and Lin, 2018; Cai et al., 2020), maximum completion time (Hao et al., 2019), and sequence dependent preparation time (Lei and Wang, 2020).

Studying the DHFSP problem is of great significance for improving production efficiency, reducing inventory costs, and better utilizing manufacturing resources. Compared to traditional HFSP problems, solving DHFSP problems can bring greater performance improvement to enterprises. In addition, DHFSP issues are widely present in multiple manufacturing scenarios, such as wafer manufacturing factories. However, due to the introduction of the factory allocation sub problem, the DHFSP problem is more complex and difficult to solve. Therefore, studying the DHFSP problem is of great significance.

2.6.1 Homogeneous distributed hybrid flow-shop scheduling problem

Compared to HFSP, the homogeneous DHFSP requires addressing three sub-problems simultaneously: the allocation of jobs among workshops, the processing sequence of jobs within each workshop, and the selection of machines. The single-layer encoding approach used for HFSP is no longer suitable for this new problem. This chapter provides a detailed introduction to the



homogeneous DHFSP and proposes a mathematical model based on mixed-integer linear programming to describe it.

- All factories have the same number of stages, the same number of machines, and identical machine performance.
- All jobs can freely choose to be processed in any factory.
- At any given moment, each job can only be assigned to one factory.
- Once a job begins processing in a certain factory, the assignment cannot be changed.

The notations of mathematical model of homogeneous DHFSP is expressed as follows.

2.6.1.1 Parameters

- F : The collection of factories, ($F = 1, \dots, f, \dots, i$);
- L : The collection of stages, ($L = 1, \dots, l, \dots, p$);
- J : The collection of jobs, ($J = 1, \dots, j, \dots, q$);
- T : The collection of machine positions, ($T = 1, \dots, t, \dots, k$);
- P_f : Machine collection for factory f ;
- $P_{l,f}$: Set of machines in the l th stage of the factory f ;
- $S_{j,l}$: The time required to complete job j during stage l ;
- $V_{j,l}$: The start time of job j on stage l ;
- $I_{j,l}$: The completion time of job j during stage l ;
- $PV_{f,p,t}$: Starting time of machine p in factory f at time t ;
- $PI_{f,p,t}$: The completion time of machine p in factory f at time t .

2.6.1.2 Decision variables

$A_{j,f}$: If job j is allocated to factory f , it is represented as 1; otherwise, it is represented as 0.

$B_{j,l,f,p,t}$: If job j is designated to the position t on machine p during stage l of factory f for processing, it is denoted as 1; otherwise, it is denoted as 0.

2.6.1.3 Mathematical model

The MILP of homogeneous DHFSP with minimization of C_{max} is formulated as follows.

$$\min. C_{max} \tag{33}$$

$$\text{s. t. } C_{max} \geq I_{j,p}, \forall j \in J \tag{34}$$

$$\sum_{f \in F} A_{j,f} = 1, \forall j \in J \tag{35}$$

$$A_{j,f} = \sum_{p \in P_{l,f}} \sum_{t \in T} B_{j,l,f,p,t}, \forall l \in L, j \in J, f \in F \tag{36}$$

$$\sum_{j \in J} B_{j,l,f,p,t} \leq 1, \forall f \in F, p \in P_{l,f}, t \in T \tag{37}$$

$$\sum_{j \in J} B_{j,l,f,p,t} \geq \sum_{j' \in J} B_{j',l,f,p,t+1}, \forall f \in F, l \in L, p \in P_{l,f}, t \in \{1, \dots, q-1\} \tag{38}$$

$$PI_{f,p,t} = PV_{f,p,t} + \sum (S_{j,l} B_{j,l,f,p,t}), \forall f \in F, l \in L, p \in P_{l,f} \quad (39)$$

$$PV_{f,p,t+1} \geq PI_{f,p,t}, \forall f \in F, p \in P_f, t \in \{1, \dots, q-1\} \quad (40)$$

$$PV_{f,p,1} \geq v_{j,j,l} - P(1 - B_{j,l,f,p,t}), \forall f \in F, l \in L, p \in P_f, t \in T \quad (41)$$

$$PV_{f,p,t} \leq V_{j,l} + P(1 - B_{j,l,f,p,t}), \forall f \in F, l \in L, j \in J, p \in P_{l,f}, t \in T \quad (42)$$

$$PV_{f,p,t} \geq V_{j,l} - P(1 - B_{j,l,f,p,t}), \forall f \in F, l \in L, j \in J, p \in P_{l,f}, t \in T \quad (43)$$

$$I_{j,l} = V_{j,l} + S_{j,l}, \forall j \in J, l \in \{1, \dots, p-1\} \quad (44)$$

$$I_{j,l} \leq V_{j,l+1}, \forall j \in J, l \in \{1, \dots, p-1\} \quad (45)$$

$$PV_{f,p,t} \geq 0, \forall f \in F, p \in P_f, t \in T \quad (46)$$

$$V_{j,l} \geq 0, \forall j \in J, l \in L \quad (47)$$

Eq. 33 represents the objective of homogeneous DHFSP optimization as minimizing C_{max} . Eq. 34 ensures that C_{max} is not less than the completion time of any factory. Eq. 35 restricts each job to be assigned to only one factory. Eq. 36 specifies that each job can be processed on only one machine in one stage within a factory at the same time. Eq. 37 indicates that each machine in each factory can only process one job at the same time. Eq. 38 indicates that each job selects a machine based on the machine's consecutive position and ensures the selection of the current position of the machine. Eqs 39, 40 represent the start and end times of each machine position in each factory. Eq. 41 ensures that the start time of a job is not earlier than the start time of the machine in the factory. Eqs 42, 43 describe the respective relationships between machine positions and job sequences. Eq. 44 defines the completion time of a job as the sum of the start time of the machine processing that job and the processing time. Eq. 45 indicates that the processing sequence is the same for all jobs. Eqs 46, 47 ensure that the start time of each machine is not earlier than 0.

2.6.2 Heterogeneous distributed hybrid flow-shop scheduling problem

Researchers have extended the homogenous DHFSP by considering the heterogeneity of the distributed factory structure, including variations in machine quantities and performance. The differences in machine performance are reflected in inconsistent processing times, and there are also sequence-dependent setup time constraints between different machines. This new problem is referred to as the heterogeneous DHFSP. The heterogeneous DHFSP introduces variations based on the DHFSP by incorporating multiple factory-specific processing environment differences. Building upon the description provided in Section 2.6.1, the distinctive settings for the heterogeneous DHFSP are as follows.

- There are i factory, each with the same number of stages, but the number of machines, processing time, and preparation time for each stage are different.
- The preparation time of a machine is related to the sequence of jobs before and after processing, that is, it is sequence dependent.
- The processing time and sequence related preparation time of the machine depend on the type of job.

The notations of mathematical model of heterogeneous DHFSP is expressed as follows.

2.6.2.1 Parameters

- F : The collection of factories, ($F = 1, \dots, f, \dots, i$);
- L : The collection of stages, ($L = 1, \dots, l, \dots, p$);
- J : The collection of jobs, ($J = 1, \dots, j, \dots, q$);
- T : The collection of machine positions, ($Q = 1, \dots, t, \dots, k$);
- P_f : Machine collection for factory f ;
- $P_{l,f}$: Set of machines in the l th stage of the factory f ;
- $S_{j,l,f,p}$: The duration it takes for job j to be completed on machine p during stage l within factory f ;
- $V_{j,l}$: The start time of job j on stage l ;
- $I_{j,l}$: The completion time of job j during stage l ;
- $PV_{f,p,t}$: Starting time of machine p in factory f at time t ;
- $PI_{f,p,t}$: The completion time of machine p in factory f at time t ;
- $V_{j',l}$: The initiation time for the setup from job j to job j' during stage l , where if job j is the initial job to be processed, then $j' = j$.

2.6.2.2 Decision variables

- $A_{j,f}$: $A_{j,f}$ is 1 if job j is allocated to factory f , and 0 otherwise.
- $B_{j,l,f,p,t}$: $B_{j,l,f,p,t}$ is 1 if job j is designated to position t on machine p during stage l of factory f for processing, and 0 otherwise.

2.6.2.3 Mathematical model

The MILP of heterogeneous DHFSP with minimization of C_{max} is formulated as follows.

$$\min. C_{max} \quad (48)$$

$$\text{s. t. } C_{max} \geq I_{j,p}, \forall j \in J \quad (49)$$

$$\sum_{f \in F} A_{j,f} = 1, \forall j \in J \quad (50)$$

$$A_{j,f} = \sum_{p \in P_{l,f}} \sum_{t \in T} B_{j,l,f,p,t}, \forall l \in L, j \in J, f \in F \quad (51)$$

$$\sum_{j \in J} B_{j,l,f,p,t} \leq 1, \forall f \in F, p \in P_{l,f}, t \in T \quad (52)$$

$$\sum_{j \in J} B_{j,l,f,p,t} \geq \sum_{j' \in J} B_{j',l,f,p,t+1}, \forall f \in F, l \in L, p \in P_{l,f}, t \in \{1, \dots, q-1\} \quad (53)$$

$$PI_{f,p,t} = pV_{f,p,t} + \sum (S_{j,l,f,p} B_{j,l,f,p,t}), \forall f \in F, l \in L, p \in P_{l,f}, t \in T \quad (54)$$

$$PV_{f,p,t+1} \geq PI_{f,p,t}, \forall f \in F, p \in P_{l,f}, t \in \{1, \dots, q-1\} \quad (55)$$

$$PV_{f,p,t+1} + P(1 - B_{j,l,f,p,t+1}) \geq PI_{f,p,t} + \sum_{j \in J} v_{j,j,l} B_{j,j,f,p,t}, \forall f \in F, l \in L, p \in P_f, t \in \{1, \dots, q-1\} \quad (56)$$

$$PV_{f,p,1} \geq v_{j,j,1} - P(1 - B_{j,l,f,p,t}), \forall f \in F, l \in L, p \in P_f, t \in T \quad (57)$$

$$PV_{f,p,t} \leq v_{j,l} + P(1 - B_{j,l,f,p,t}), \forall f \in F, l \in L, j \in J, p \in P_{l,f}, t \in T \quad (58)$$

$$PV_{f,p,t} \geq v_{j,l} - P(1 - B_{j,l,f,p,t}), \forall f \in F, l \in L, j \in J, p \in P_{l,f}, t \in T \quad (59)$$

$$F_{j,l} = V_{j,l} + \sum_{f \in F} \sum_{p \in P_{l,f}} \sum_{t \in T} S_{j,l,f,p} A_{j,l,f,p,t}, \forall j \in J, l \in \{1, \dots, p-1\} \quad (60)$$

$$I_{j,l} \leq V_{j,l+1}, \forall j \in J, l \in \{1, \dots, p-1\} \quad (61)$$

$$PV_{f,p,t} \geq 0, \forall f \in F, p \in P_f, t \in T \quad (62)$$

Eq. 48 represents the objective function aiming to minimize the maximum completion time. Eq. 49 denotes the objective value, ensuring it is greater than or equal to the completion time of any factory. Eq. 50 represents the constraint that each job can only be assigned to one factory. Eq. 51 ensures that each job can only be processed within one factory and only one machine can be selected at any given stage. Eq. 52 indicates that each machine is limited to handling a single job concurrently. Eq. 53 ensures that every job chooses a machine based on its sequential position and guarantees the selection of the machine currently in that position. Eqs 54 and 55 determine the start and end times of each machine's position in all factory. Eq. 56 represents the adjustment constraints for sequence-related setup times. Eq. 57 specifies that the initiation time of the initial job must not be earlier than the setup time for each machine within the factory. Eqs 58, 59 define the corresponding relationships between machine positions and job sequences. Eq. 60 defines the completion time of a job. Eq. 61 defines that jobs pass through all stages within a factory in sequence. Eq. 62 specifies that no machine at any level of a manufacturing shall have a start time that is less than 0.

Researchers have conducted studies on single-objective DHFSP with the objective of minimizing the maximum completion time. Various algorithms have been employed to address this problem, including the adaptive iterative greedy algorithm (Ying and Lin, 2018), dynamic frog-leaping algorithm (Cai et al., 2020), hybrid brainstorming algorithm (Hao et al., 2019), artificial bee colony algorithm (Li Y. et al., 2019; Li et al., 2020b; Li Y. et al., 2021), multi-neighborhood iterative greedy algorithm (Shao et al., 2020), dual-population competitive cultural genetic algorithm (Wang and Wang, 2020), improved brainstorming algorithm (Li J. et al., 2021), iterative greedy algorithm (Wang and Wang, 2019), and teaching optimization algorithm (Lei and Su, 2023). The common objective of these algorithms is to minimize the maximum completion time.

For the homogeneous DHFSP, Shao et al. (2021) proposed a MOEA based on multi-neighborhood local search to solve the multi-objective DHFSP. Cai et al. (Cai et al., 2018) proposed an improved non-dominated sorting genetic algorithm II (NSGA-II) for finding the Pareto-optimal solutions of the multi-objective distributed permutation flow-shop scheduling problem (DPFSP). The algorithm utilizes new solution representation, new population re-initialization method, effective crossover and mutation operators, as well as local search technique.

Switching to the heterogeneous DHFSP, Shao et al. (2022) employed network meta-heuristic algorithm (NMA) to address the problem of energy and labor perception in distributed heterogeneous DHFSP (ELDHHFSP). Initially, the author introduced a MILP model to depict ELDHHFSP and defined multiple optimization objectives. Subsequently, the author proposed the NMA, which comprises two crucial components: a probabilistic network model and a learning-based local search,

corresponding to global search and local search respectively. Zhang et al. (2023b) employed a multi-objective genetic algorithm along with particle swarm optimization and a Q-learning-based local search method to address the energy-efficient heterogeneous DHFSP. Their objectives were to optimize both the manufacturing C_{\max} and total energy consumption. To expedite rapid convergence of the solution across multiple directions on the Pareto frontier, they employed multiple sets of particle swarm optimization as a global search strategy. To enhance the search for problem-specific knowledge, they designed two local search strategies to further improve the quality and diversity of solutions. Additionally, they utilized Q-Learning (QL) to guide variable domain exploration for a better balance between algorithm exploration and exploitation.

Currently, both Particle Swarm Optimization (PSO) and QL algorithms are highly regarded and essential in this field. PSO with its exceptional search and optimization capabilities, has become a preferred tool for researchers tackling job scheduling problems. Simultaneously, QL algorithms have gained widespread attention for their outstanding performance in the field of RL, offering an effective approach for systems to learn and optimize strategies in unknown environments. Next, we will delve into these two methods in detail.

2.6.3 Particle swarm optimization and Q-learning

In 1995, James Kennedy and Russell Eberhart (Kennedy and Eberhart, 1995) were inspired by the foraging behavior of bird flocks and proposed the PSO algorithm. The inspiration of particle swarm optimization algorithm comes from the study of the foraging behavior of bird swarms, which help the entire population find the optimal destination through collective information sharing. Single objective PSO has significant advantages in convergence speed and is therefore widely used in solving MOP. Reyes-Sierra and Coello Coello (2006) have published a research result on using PSO to handle MOP, which is called multi objective particle swarm optimization (MOPSO). The conventional calculation steps for MOPSO are as follows.

- Initialize the particle swarm, including setting the number of particles, velocities, and positions.
- Evaluate the quality of particles using multiple fitness functions, typically employed in MOPSO.
- Update the individual's historical best positions, which, in multi-objective algorithms, form a collection known as pbestset.
- Update the global best position of the particle swarm, where in multi-objective algorithms, the global best positions constitute a set often referred to as gbestset.
- Output the optimal solution.

Update the velocity and position of each particle based on the following formula.

$$v(t) = wv(t-1) + c_1r_1(pbest - x(t)) + c_2r_2(gbest - x(t)) \quad (63)$$

$$x(t+1) = x(t) + v(t) \quad (64)$$

In Eqs. 63, 64 Where $v(t)$ and $x(t)$ respectively represent the velocity and position of a particle. The w is the inertia weight, c_1 and c_2 represent the cognitive learning factor and social learning factor, which can adjust the particle's proximity to its previous best distance. r_1 and r_2 are random numbers between 0 and 1 used to enhance the randomness of particle movement.

The MOPSO method, although proficient in addressing MOPs, faces certain challenges. One prominent drawback is its susceptibility to the choice of control parameters (Zhang W. et al., 2023), including the inertia weight w , cognitive learning factor c_1 , and social learning factor c_2 . Inadequately calibrated parameters may impede convergence, leading to suboptimal solutions. Furthermore, the algorithm's effectiveness can be influenced by the selection of fitness functions and their respective weights. To address these challenges, researchers have explored alternative optimization approaches. One intriguing avenue is the application of QL, a value-based RL algorithm.

In this algorithm, $Q(s, a)$ represents the expected return value when taking action a in a specific state s . At each time step, the environment provides a corresponding reward based on the agent's action. The core idea of the algorithm is to construct a Q-table, which is used to store Q-values for different state-action pairs, and then select actions that maximize the expected return based on these Q-values.

When building QL, Firstly, it is necessary to define real-time rewards, which guides the agent to select actions to obtain the Q-table. The update of Q-values is as follows (Watkins and Dayan, 1992):

$$Q(s, a) = Q(s, a) + \alpha(R(s, a) + \gamma \max_{a'} Q(s', a') - Q(s, a)) \quad (65)$$

In Eq. 65, where the parameter α is the learning rate, with a value of 1, and the parameter γ is the discount factor, which determines the extent of value iteration updates, ranging from 0 to 1. In this study, Q-values are stored in the Q-table and updated iteratively. The agent selects actions from the action set based on an ϵ -greedy behavioral policy. The symbol ϵ represents the probability of choosing a greedy action.

In order to enhance the flexibility of the PSO algorithm during the iteration process, cope with convergence difficulties and overcome local optimization problems, an effective method is to adopt an adaptive parameter adjustment strategy. This can be achieved by designing appropriate states, actions, and reward mechanisms, and utilizing QL. The QL based PSO (QL-PSO) (Xu and Pi, 2020) algorithm can better adapt to the characteristics of the problem, automatically adjust parameters to adapt to different problem instances, thereby improving its performance and robustness. This adaptive parameter adjustment strategy helps to overcome the limitations of conventional PSO algorithms and make them perform better in complex problems. The algorithm process of QL-PSO is roughly as follows (Zhang et al., 2023b).

- Initialize particle swarm and evaluate particle quality.
- Employ a mixed sampling strategy to divide the particle swarm into three sub-particle swarms and employ the PSO algorithm to update each sub-particle swarm.
- Divide the entire particle swarm into three sub particle swarms again, and initialize the Q-table, actions, states, and rewards.

Then use the QL algorithm for local search on each sub particle swarm.

- Merge the three sub-particle swarms with the overall particle swarm from the end of the second step, employing a selection strategy based on Pareto dominating and dominated relationship-based fitness function (PDDR-FF) (Zhang et al., 2014), selecting the top 50% of particles ranked by fitness function values to form the next-generation of the new particle swarm.

3 Machine learning and multi-objective evolutionary algorithms

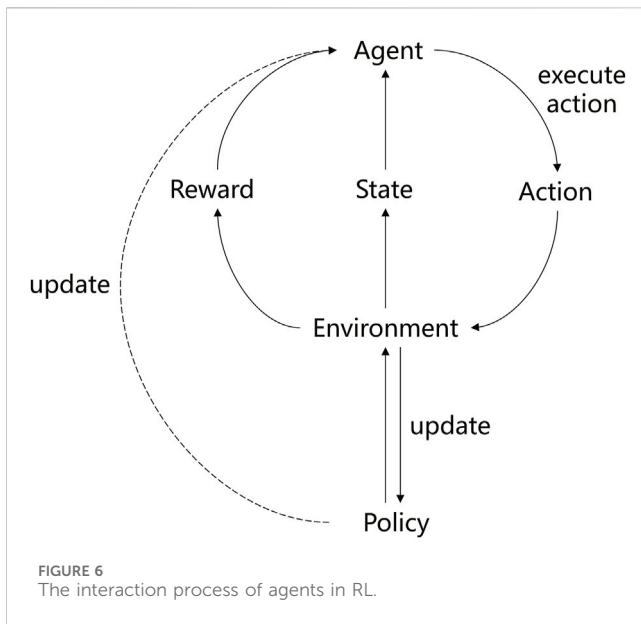
3.1 Machine learning

Traditional methods encounter a trade-off between efficiency and quality. Meta-heuristic algorithms, while powerful, can be computationally expensive, making them less practical in dynamic environments with frequent events, where optimal solution quality may suffer. In contrast, rule-based methods provide solutions quickly but at the cost of sacrificing solution quality. These methods rely on limited information types and simple mathematical operations for decision-making.

During the search process, EAs accumulate valuable information related to search and population dynamics, as well as problem characteristics. ML techniques can harness this information to extract meaningful insights, enhancing the overall search performance of algorithms. ML techniques include statistical methods such as interpolation and regression (Cleveland et al., 2017), orthogonal experimental design (Lopes et al., 2020), opposition-based learning (Mahdavi et al., 2018), principal component analysis (Abdi and Williams, 2010), artificial neural networks (Ismayilov and Topcuoglu, 2020), support vector machines (Pisner and Schnyer, 2020), cluster analysis (Trebuña and Halčinová, 2013), case-based reasoning (Kolodner, 1992), mean and variance (Makridakis et al., 2018), competitive learning (Rumelhart and Zipser, 1985), Bayesian networks (Heckerman, 2008), and RL (Wiering and Van Otterlo, 2012). The amalgamation of ML approaches with evolutionary computing has been empirically shown to provide benefits in terms of both convergence speed and solution quality. Early researches (Lin and Gen, 2018) have proposed using ML techniques to enhance EAs. Supervised learning (Jourdan et al., 2006; Zhang J. et al., 2011) has faster convergence and better solution quality compared to evolutionary learning (Kotsiantis et al., 2007).

EAs does not rely on backpropagation and has become a powerful optimization tool due to its high parallelism and wide applicability. Their ability to conduct global searches in parameter space and robustness make them particularly effective, with relatively low demands on environmental reward settings. However, EAs exhibit lower sample efficiency, a relatively singular exploration approach, and a lack of learning and generalization capabilities.

Comparatively, with its characteristic of data-driven decision-making, ML can achieve performance in specific scenarios that approaches or even surpasses human capabilities. Yet, ML faces significant limitations. Primarily, its performance highly depends on



the quality of training data, with low-quality data potentially leading to incomplete or inaccurate models. Secondly, the task-specific model types in ML hinder its ability to seamlessly perform diverse tasks. Lastly, the lack of generality in ML models prevents them from adapting to various unspecific tasks. Therefore, when considering the characteristics of EAs and ML, it is essential to choose an appropriate method based on the specific requirements of the problem or flexibly combine both in practical applications to leverage their respective strengths and compensate for their limitations.

Having discussed the characteristics of EAs and ML, our focus will now shift towards their integration, particularly the fusion of RL and MOEAs. This integration is not only aimed at overcoming their respective limitations but also at creating a more comprehensive and powerful optimization framework. In the following chapters, we will delve into the methods, advantages, and potential application areas of this integration.

3.2 Reinforcement learning

RL is a learning approach that seeks to optimize the total reward obtained via the iterative interaction between an autonomous agent and its surrounding environment (Arulkumaran et al., 2017). It can be delineated using the Markov Decision Process framework, which includes action space A , state space S , reward function r , state transition probability p and discount factor γ . When the agent is in state $s \in S$, it can choose action $a \in A$, and the environment will transition to a new state s' with probability $p(s'|s, a)$ and provide a reward $r(s, a, s')$. The agent's goal is to learn a policy $\pi(a|s)$ that maximizes the cumulative reward $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$, where γ is the discount factor that balances the importance of immediate and future rewards.

In the context of scheduling inside a shop, the state space S may be used to denote the existing state of the shop, encompassing factors such as the condition of machines and the advancement of jobs. The action space A can represent scheduling decisions, such as

which jobs to assign to which machines. The state transition probability p can represent the change in the shop state based on the scheduling decision. The reward function r can represent the reward obtained from the scheduling decision, such as the number of completed jobs and time saved. The discount factor γ can represent the ratio between immediate and future rewards, typically between 0 and 1. The policy π of the agent may be used to make decisions depending on the current state of the shop, with the objective of maximizing the cumulative reward. The interaction process between an intelligent agent and its environment is shown in Figure 6.

Through RL algorithms, shops can achieve intelligent scheduling to improve efficiency and quality. Compared to supervised learning and evolutionary learning, RL has the following advantages.

1. RL can operate without labeled data, which makes it more suitable for real-world environments.
2. It is more adaptive and can learn and make decisions in dynamic and complex environments.
3. RL receives real-time feedback by interacting with the environment, which helps with rapid learning and decision-making. Decision-making.
4. RL can explore and discover better solutions autonomously, without relying solely on human prior knowledge.
5. In some cases, RL may outperform supervised learning and evolutionary learning.

3.3 MOEAs and their application to scheduling problems

The extensive use of MOEAs has attracted significant attention in the context of multi-objective shop scheduling problems. This chapter will predominantly concentrate on recent advancements within this domain, providing an in-depth exposition of the pertinent progress achieved in recent years. Wang et al. (Wang et al., 2021) utilized a multi-objective mathematical model and a modified MOEA/D algorithm to address the energy-efficient scheduling problem in a distributed heterogeneous welding flow-shop scheduling problem, with the aim of minimizing both total energy consumption (TEC) and completion time simultaneously. In the revised MOEA/D algorithm, various genetic operators and problem-specific local search strategies were designed for multi-level optimization. However, the implementation and computational complexity of the aforementioned methods are high, which may require a long running time to achieve good results. In response to this, Zhang et al. (Zhang et al., 2022) proposed an automatic MOEA to solve the HFSP with C_{\max} and total number of sublots as objectives. This study solely focuses on the two objectives of C_{\max} and the total number of sub-batches, limiting its ability to tackle intricate production scheduling challenges. To address this issue, Han et al. (Han W. et al., 2021) proposed a heuristic decoding MOEA to solve the HFSP with worker constraint. The proposed method can effectively address complex production scheduling problems. Moreover, experimental results show that the proposed algorithm performs well in achieving the C_{\max} objective. Deng and Wang (Deng and Wang, 2017) proposed a competitive

TABLE 1 The application of MOEAs in shop scheduling problems.

	Methodology	Shortcomings	Article
Hybrid flow shop	Automatic MOEA	Poor generalizability	Zhang et al. (2022)
	Heuristic decoding MOEA		Han et al. (2021a)
Fuzzy Job-Shop	Hybrid Adaptive Differential EA	High time and space complexity	Wang et al. (2022a)
	ENSHA	Poor Scalability	Li et al. (2019b)
Flexible job shop	hybrid artificial bee colony algorithm	High implementation complexity	Gong et al. (2020)
	ENSGA	Poor generalizability	Tan et al. (2021)
Distributed hybrid flow shop	EDA+IG	Stochasticity	Zheng et al. (2020)
Distributed permutation flow shop	CMA	Poor generalizability	Deng and Wang (2017)
Distributed heterogeneous welding flow shop	A multi-objective mathematical model and a modified MOEA/D algorithm	The algorithm requires improvement	Wang et al. (2021)

memetic algorithm (CMA) to solve the multi-objective DPFSP with the criteria of C_{\max} and total tardiness (TTD). The generality of the above-mentioned methods is relatively poor, as they cannot fully consider various uncertainties in the production process.

To tackle this challenge of generality, Wang et al. (Wang G.-G. et al., 2022) proposed a hybrid adaptive differential evolution algorithm to solve the multi-objective fuzzy JSP. Li et al. (Li Z. et al., 2019) proposed an elitist non-dominated sorting hybrid algorithm (ENSHA) to solve the multi-objective FJSP with sequence-dependent setup times/costs, which minimizes two objectives: maximum completion time and total setup cost. Gong et al. (Gong et al., 2020) proposed a hybrid artificial bee colony algorithm to solve the FJSP with worker flexibility. However, all the above algorithms are incapable of handling large-scale problems.

Within the domain of shop scheduling problems, the expansion of the problem's scale leads to an escalated requirement for computational resources and time (Li and Pan, 2015), while intensifying issues of resource competition and bottlenecks due to a larger number of tasks contending for limited resources (Zhang and Wu, 2010), thereby causing imbalances in resource utilization and production delays (Goli et al., 2019). To overcome the problem, Tan et al. (2021) proposed a fatigue-aware dual-resource-constrained flexible job-shop problem method, aimed at simultaneously alleviating fatigue and improving production efficiency through joint scheduling of machines and workers. A multi-objective optimization model was developed with the aim of reducing both the maximum worker fatigue and completion time. Furthermore, a suggested solution to address the issue included the development of an improved iteration of NSGA-II, known as Enhanced NSGA-II (ENSGA). The ENSGA algorithm has four distinct scheduling rules that have been specifically developed to provide solutions of superior quality. Additionally, two distinct area structures have been established, using an innovative methodology for designing critical routes. This technique serves to significantly enhance the efficacy of local search operations. Although this method performs well on large-scale problems, further experiments and data may be required to verify its universality on different domains and datasets.

Regarding the strong coupling dilemma in shop scheduling (Galiana et al., 2005), Zheng et al. (2020) proposed a collaborative EA with problem-specific strategies by combining estimation of distribution algorithm (EDA) and iterative greedy search (IG) to tackle the multi-objective fuzzy distributed hybrid flow-shop problem with fuzzy processing time and fuzzy delivery time. The following Table 1 summarizes the applications of MOEAs in scheduling problems.

3.4 Applications of ML in solving scheduling problems

When confronted with intricate shop scheduling dilemmas, MOEAs necessitate multiple iterations within the solution space, resulting in a relatively sluggish convergence speed and potential performance limitations (Zhang et al., 2019). In contrast, ML exhibits enhanced adaptability in dynamic shop scheduling scenarios and can leverage neural networks to augment its learning capabilities.

In the domain of DJSP, traditional scheduling methods often only consider existing jobs and ignore the possibility of new ones appearing at any time. To address this issue, Luo et al. (Luo, 2020) proposed a new job insertion policy leveraging the deep Q-learning network (DQN) algorithm, which can dynamically insert new jobs and optimize scheduling by real-time perception and decision-making of the current shop state. In addition to this, Luo et al. (2021a) also proposed an online rescheduling framework named two-layer DQN (THDQN) for the dynamic multi-objective flexible job-shop Problem with the insertion of new jobs. Four distinct objectives were introduced, in accordance with four distinct variations of reward functions, with each objective optimizing a tardiness or machine utilization metric. However, the aforementioned algorithms only consider a limited number of production environment factors, and all of them are value-based methods that cannot directly optimize policies. Therefore, Luo et al. (Luo et al., 2021b) used a real-time scheduling method for the dynamic part no-wait multi-objective FJSP in modern discrete flexible manufacturing systems, based on hierarchical multi-agent

deep reinforcement learning (DRL) called hierarchical multi-agent proximal policy optimization (HMAPPO), which handles situations such as new job insertion and machine failures. The method consists of three intelligent agents based on proximal policy optimization (PPO), namely, the target agent, job agent, and machine agent.

With the intention of elevating generality in practical implementations, Samsonov et al. (2021) respectively used DQN and soft actor-critic (SAC) algorithms to solve the production planning and control problem in DJSP. Lang et al. (2020) used a discrete-event simulation model to train two DQN agents, where one agent is responsible for selecting operation sequences and the other is responsible for allocating jobs to machines. They applied this model to solve a FJSP with integrated process planning. The research shows that DQN appears to generalize the training data to other problem instances.

The aforementioned algorithm exhibits certain limitations with regards to training speed and efficiency, which may potentially encumber its training efficacy on large-scale datasets. Furthermore, the algorithm's intricacy and computational requirements might lead to suboptimal performance within resource-constrained environments, thereby imposing constraints on its feasibility for practical shop scheduling. To better mitigate these issues, Liu et al. (2022a) utilized a double deep Q-network (DDQN) algorithm to train a scheduling agent in a FJSP with constant job arrival times. The algorithm effectively captures the correlation between production information and scheduling objective in order to make timely scheduling choices. Hameed and Schwung (2020) proposed a method that combines distributed intelligent agent learning and internal agent interaction mechanisms, and uses graph neural networks as feature extraction models to address the scalability and environmental variability issues in JSP. The paper points out that compared to centralized optimization algorithms such as genetic algorithms, graph neural networks have greater advantages in representing complex and variable scheduling environments. Additionally, the authors validated the superiority of the proposed method in two experimental scenarios: a robot manufacturing unit and an injection molding machine. However, there are some potential challenges to the stability of this method. In actual shop production environments, complex work processes, resource constraints, and unforeseeable interference factors may result in this method not performing stably enough in response to specific workshop situations. This deficiency may be particularly evident in the face of sudden changes, urgent tasks, or resource bottlenecks, thereby affecting the accuracy and stability of scheduling results.

To improve these issues, Zhang et al. (2020) utilized the PPO algorithm to automatically learn the priority dispatching rule (PDR) and combined it with graph neural networks to effectively improve the generalization of the algorithm. Luo P. C. et al. (2022) used the PPO algorithm to solve the DJSP under resource constraints. Wang H. et al. (2022) have designed a new dynamic multi-objective scheduling algorithm (DMOSA) based on DRL, which utilizes two DQN and a real-time processing framework to handle changing events and generate comprehensive scheduling strategies. The method was subjected to simulation using six distinct kinds of dynamic events that are often seen in real-world production settings. The optimization procedure focused on three

specific objectives: average machine utilization, average job processing delay rate and the longest job processing time. These objectives were pursued while adhering to a predefined set of limitations. With that being said, it is important to note that these algorithms may need substantial experimentation and refinement, including aspects such as reward mechanisms, neural network architecture, and scheduling policies, in order to effectively address a wider array of industrial production requirements. As a result, practical applications may entail a substantial amount of time and effort.

With the objective of improving scheduling efficiency and diminishing operating costs, Leng et al. (2022) developed a multi-objective DQN algorithm to determine the Pareto frontier. The purpose of reward shaping is to enhance the convergence of the neural network. The algorithm addressed the multi-objective reordering scheduling problem in automotive manufacturing systems with color batch requirements in the painting shop and sequencing requirements in the assembly shop. This approach requires offline training before implementation and may only be suitable for manufacturing factories with specific configurations. Therefore, it is necessary to further improve the practicality and adaptability of the algorithm.

The following is a summary of the application of RL in scheduling problems in Table 2.

In summary, it can be inferred that ML, especially RL, has the following advantages and disadvantages in scheduling applications:

Advantages:

1. Efficiency: ML utilizes the insights of historical data to enhance the scheduling process, enabling the system to predict and optimize based on identified patterns and trends.
2. Adaptability: The ML model demonstrates the ability to dynamically adjust to constantly changing conditions.
3. Scalability: ML can be used to solve various complexity and scale scheduling problems from simple task scheduling to large-scale production scheduling.
4. Generalizability: Once an ML model is trained for a set of scheduling problems, it can be applied to similar scheduling problems in different fields.

Disadvantages:

1. Data requirements: Many scheduling problems have limited data, but ML models require a substantial amount of data for training and optimization.
2. Interpretability: Certain ML models, such as neural networks, might provide challenges in terms of comprehending their decision-making mechanisms.
3. Overfitting: Models trained in ML may excessively tailor themselves to the training data, resulting in poor performance when applied to new data or under conditions of change.

The strengths and weaknesses of RL in workshop scheduling problems are illustrated in Figure 7.

Despite the advantages that ML holds in addressing scheduling issues, it is crucial to contemplate its limitations and potential drawbacks before practical implementation, ensuring the full realization of its immense potential.

TABLE 2 The application of RL in shop scheduling problems.

Type of shop scheduling	Methodology	Shortcomings	Ref
Flexible job-shops	DRL	Poor flexibility	Luo (2020)
	DDQN	Poor generalizability	Liu et al. (2022a)
	Two DQN agents	Insufficient convergence quality and algorithm stability	Lang et al. (2020)
Dynamic job-shop	DQN and SAC	Poor training efficiency	Samsonov et al. (2021)
	PPO	Poor Scalability	Luo et al. (2022a)
	HMAPPO	Poor generalization and scalability	Luo et al. (2021b)
Job-shop	Distributed intelligent agent	Poor stability	Hameed and Schwung (2020)
	PPO+PDR	Poor interpretability	Zhang et al. (2020)
Dynamic flexible job-shop	DMOSA based on DRL	Poor generalizability and practicality	Wang et al. (2022b)
	THDQN	Poor flexibility	Luo et al. (2021a)
Assembly shop	MODQA	Poor practicality and generalizability	Leng et al. (2022)

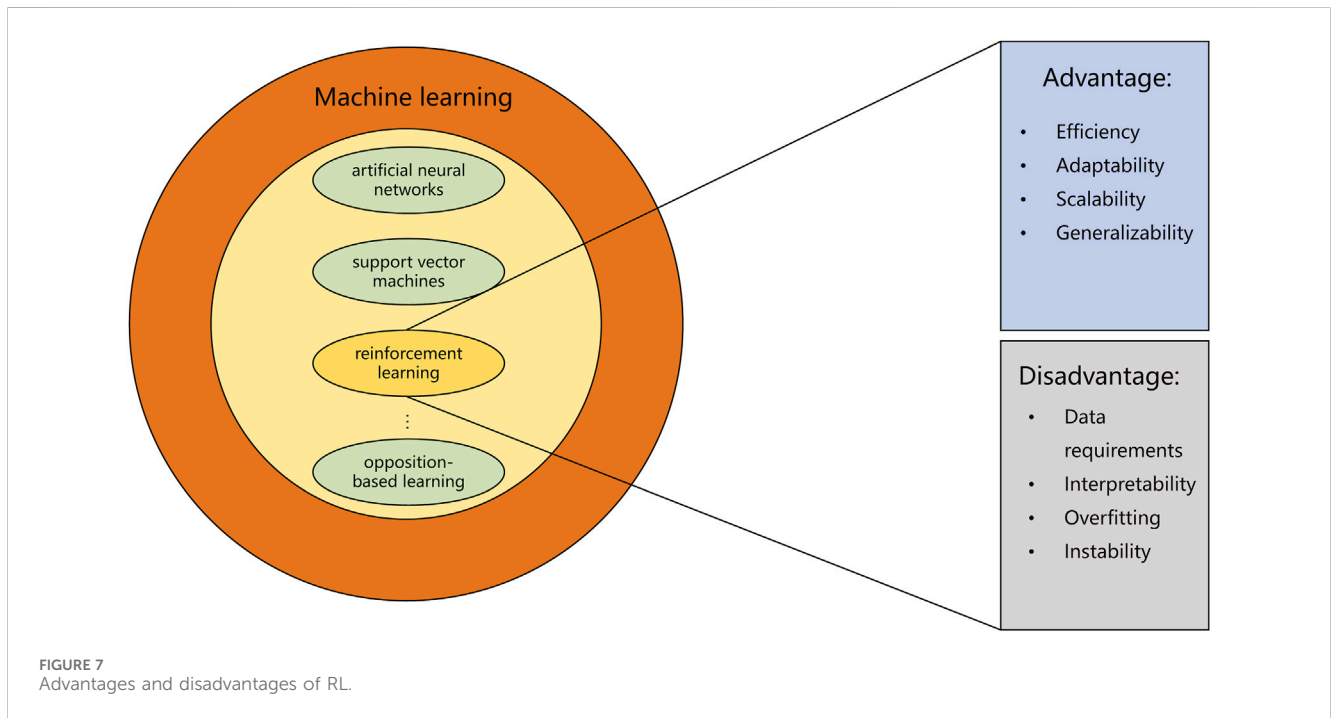


FIGURE 7 Advantages and disadvantages of RL.

4 Enhancing MOEAs for solving shop scheduling problems

Enhancing MOEAs (EMOEAs) utilizes ML algorithms to model the objective function and then integrates the resulting model into MOEAs. Through this approach, MOEAs can better search the solution space and find optimal solutions. ML can assist MOEAs in exploring unknown domains and improving search efficiency, thus accelerating the solution process.

Meta-heuristic optimization techniques are commonly utilized to address intricate optimization problems across diverse domains, with MOEAs being a well-researched method due to its efficacy in

tackling multi-objective problems. Nevertheless, conventional MOEAs approaches encounter drawbacks such as the need for manual tuning of algorithmic parameters to achieve optimal performance and suboptimal results in handling high-dimensional problems with large scales. To overcome these challenges, ML-assisted techniques have been proposed in recent years, some of which have demonstrated remarkable results in resolving scheduling problems. In this chapter, we will classify and present different methods based on the proportion and weight of RL and MOEAs in each approach, which include ML-assisted MOEAs, MOEAs-assisted ML, and collaborative MOEAs and ML.

4.1 ML assists MOEAs

In conventional ML algorithms, [Pericleous et al. \(2017\)](#) investigated the hybridization of MOEA/D with six general-purpose heuristic methods to locally optimize solutions during the evolutionary process. Initially, six individual hybrid MOEA/D were considered, whereby the same local search heuristic method was applied at each step of evolution. Subsequently, in light of the characteristics of the issue and the objectives, the integration of MOEA/D with Meta-lamarckian learning (MLL) was used to dynamically choose the most effective local search heuristic approach from the general heuristic pool during each evolutionary phase and for every problem neighborhood.

[Wang and Tang \(2017\)](#) proposed a machine-learning-based multi-objective memetic algorithm (ML-MOMA) to address the discrete PFSP. Within ML-MOMA, each solution is allocated an individual archive for storing its discovered non-dominated solutions, and a novel population update approach is devised based on these individual archives. Additionally, a novel adaptive multi-objective local search method is proposed, which utilizes the analysis of historical data acquired during the search process to dynamically decide the selection of non-dominated solutions for local search and the execution of local search itself.

In response to the labor shortages and social distancing challenges faced by manufacturing plants affected by COVID-19, [Li et al. \(2021b\)](#) studied an energy-efficient JSP with limited workers. The study established a multi-objective model with five objectives: maximum completion time, TTD, total idle time (TIT), total worker cost (TWC), and TEC. In order to address this particular many-objective optimization issue, a unique approach was taken by including a fitness assessment mechanism that relies on fuzzy correlation entropy. Additionally, two distinct techniques for constructing reference points were introduced to establish a connection between the many-objective optimization problem and fuzzy sets. An environmental selection mechanism was proposed to achieve a balance between solution convergence and variety, using the fuzzy correlation entropy and clustering approach.

[Lou et al. \(2022\)](#) investigates the multi-objective FJSP with human factors to reduce costs and improve efficiency. However, traditional FJSP only consider machine flexibility, while ignoring human factors. Thus, this paper establishes a multi-objective mixed-integer nonlinear programming model and proposes a learning and decomposition-based multi-objective memetic algorithm (MOMA-LD) to simultaneously optimize three objectives: C_{\max} , machine workload (MW), and total machine workload (TMW). By incorporating a learning-based adaptive local search algorithm into a decomposition-based MOEA. The MOMA-LD framework utilizes ML approaches to determine the suitability of local search solutions. It also dynamically allocates computing resources during the evolutionary process, taking into account the convergence level of the population.

[Karimi-Mamaghan et al. \(2023\)](#) have spearheaded the conception of a cutting-edge and highly efficient iterative greedy algorithm, cultivated through their research. The algorithm's eminent contribution resides in its revolutionary perturbation mechanism, ingeniously employing QL to discern and select for appropriate perturbation operations among the search

process. However, this approach is only applicable to single-objective problems.

In comparison to conventional ML, RL is better suited for exploration and decision-making problems, demonstrating greater adaptability and generalization capabilities. [Zhao et al. \(2023b\)](#) proposed a RL-based brain storm optimization (RLBSO) approach to tackle the multi-objective and energy-efficient distributed assembly NFSP. The optimization objectives include minimizing the C_{\max} , minimizing the total TEC, and achieving a balanced resource allocation. Four operations were designed, including key factory insertion, key factory swap, key factory insertion into other factories, and key factory swap with other factories, to optimize the objective of minimizing the C_{\max} . The QL mechanism was employed to guide the operation selection and avoid blind search during the iteration process. The product was assigned to factories in the objective space based on the processing time using a clustering-based learning mechanism to balance the resource allocation. To reduce the TEC, the operation speed on the non-critical path was slowed down, considering the characteristics of the NFSP. Moreover, the author ([Zhao et al., 2021b](#)) utilized the cooperative water wave optimization (CWWO) algorithm to address the distributed assembly scheduling problem in an no-idle flow-shop, enhancing the search capabilities of the algorithm by introducing RL, path relinking, VNS methodology, and multi-neighborhood perturbation strategies, ultimately improving the efficiency and stability of the problem-solving process. Nonetheless, this method only optimizes single-objective problems and is only applicable to specific distributed flow-shop scheduling problems, which limits its applicability.

[Cheng et al. \(2022\)](#) advanced a hybrid job-shop and flow-shop production scheduling problem using velocity scaling and no idle time strategies. A MILP was formulated to optimize both production efficiency and energy consumption by determining the speed levels of operations and the production sequence for job-shop and flow-shop products. Following that, a novel hyper-heuristic method called multi-objective Q-learning-based hyper-heuristic with Bi-criteria selection (QHH-BS) was devised, which utilizes a multi-objective QL approach to generate a collection of PF solutions of superior quality. This algorithm introduces a novel three-layer encoding scheme for representing the production sequence of job-shop and flow-shop products. Additionally, a sequence implementation is employed, which incorporates both PF and indicator-based selection criteria to promote diversity and convergence. Furthermore, a QL algorithm with a reward mechanism based on multi-objective indicators is utilized to select an optimizer from a pool of three high-performing optimizers in each iteration, thereby facilitating improved exploration and exploitation.

[Wang J.-j. et al. \(2023\)](#) utilizing the CMA to tackle the energy-aware distributed welding shop scheduling problem arising from the trends of globalization and sustainable industrial development, with the aim of simultaneously optimizing the minimal manufacturing cycle time and TEC. A proposal is made to improve the quality and diversity of the initial population by utilizing a hybrid initialization approach based on the modified Nawaz-Enscore-Ham (NEH) algorithm ([Fernandez-Viagas et al., 2017](#)). A feedback-based collaborative search is developed by effectively leveraging historical data, incorporating a collaboration selection method

that aims to strike a balance between exploration and exploitation. Furthermore, the study proposes a number of problem-specific operators and introduces a local reinforcement method based on QL to augment the system's development capabilities.

Li et al. (2023) proposed a MILP model and a learning-based reference vector membrane algorithm (LRVMA) to address the energy-efficient FJSP with type 2 processing times with multiple objectives, aiming to enhance profit and reduce energy consumption for better practical production simulation. The study included the development of four problem-specific beginning rules and the introduction of four problem-specific local search strategies. The Tchebycheff decomposition technique was successfully used to attain an efficient method for solution selection. Additionally, a parameter selection strategy based on RL was introduced to enhance the variety of non-dominated solutions. Furthermore, an energy-saving strategy was designed to reduce energy consumption.

4.2 MOEAs assists ML

This approach utilizes MOEAs algorithms for model optimization. Specifically, MOEAs improves the performance and prediction accuracy of the model by optimizing its hyperparameters or model selection. This approach can help ML algorithms overcome local optima and over fitting problems, and improve the model's generalization ability.

Khadka and Tumer (2018) proposed a hybrid algorithm called evolutionary reinforcement learning (ERL) that combines EA and DRL to address issues related to sparse reward temporal credit assignment, effective exploration, and unstable convergence due to hyperparameter sensitivity in DRL algorithms. ERL leverages the advantages of EA, such as sequential credit assignment based on fitness indicators, diverse policy exploration, and stability improvement using population models, while also utilizing the gradient information of DRL algorithms to improve sample efficiency and learning speed.

Furthermore, The author introduces a method called Collaborative evolutionary reinforcement learning (CERL) (Khadka et al., 2019), designed to address the exploration problem and hyperparameter sensitivity in DRL. CERL employs multiple learning algorithms to simultaneously explore different regions of the solution space, and dynamically allocates computational resources to support the best learners. Neural evolution combines these algorithms to generate a new learner that outperforms all the constituent learners in the experiment, while also possessing higher sample efficiency. Pourchot and Sigaud (Pourchot and Sigaud, 2018) introduced a novel method that integrates Deep Neuroevolution with DRL to address the low sample efficiency of Deep Neuroevolution and the sensitivity to hyperparameters of Deep RL. They introduced a new algorithm, which combines the simple cross-entropy method with a new Deep RL algorithm called Twin Delayed Deep Deterministic Policy Gradient. Bodnar et al. (2020) presents a novel algorithm named PDERL to enhance the scalability of GA in RL. This algorithm integrates evolution and learning hierarchically by employing a learning-based mutation operator to compensate for the simplicity of GA gene encoding. Suri et al. (2020) provide a review of the strengths and limitations of RL and MOEAs in

recent years, and discuss how to combine these two methods to address scalability and hyperparameter sensitivity issues in RL. Moreover, they propose an Evolution-based SAC algorithm based on the ERL framework, which integrates SAC algorithm and evolutionary strategies. Cideron et al. (2020) propose a novel RL algorithm, which combines the advantages of offline policy RL algorithms with the quality diversity (QD) approach. The authors train a population of offline deep DRL agents to maximize both diversity within the population and the agents' rewards. QD-RL selects agents from the diversity-reward PF, enabling stable and efficient population updates. Marchesini et al. (2021) propose a novel hybrid framework that combines EAs and DRL to leverage the strengths of both approaches. To address the issue of high computational cost in evaluating ERL algorithms, the authors propose the SUPE-RL algorithm, which performs parallel evaluation of the population at fixed intervals during the RL process.

The above content indicates that EA-assisted RL is mainly applied to DRL, where the policy of DRL is used as individuals in the population for RL optimization. Although it belongs to evolutionary computation guided RL, the empirical guidance is relatively weak due to the randomness of the samples, which limits the improvement of environmental exploration capability.

In addition, the utilization of MOEAs assisted ML in shop scheduling is still in its nascent stage. For MOPs, reasonable and effective hybrid methods need to be designed to solve them. Therefore, this field still deserves further exploration and research.

4.3 The collaboration between MOEAs and ML

This approach combines the advantages of the previous two methods, namely, utilizing both MOEAs and ML for solving problems. In this approach, MOEAs and ML algorithms work together, providing information and feedback to each other, to achieve better solution results. Specifically, MOEAs can provide more data and optimization methods for ML, while ML can provide more accurate objective function modeling and prediction for MOEAs. This approach can help solve more complex problems, improve solution efficiency and accuracy.

Chen et al. (2020) have proposed a self-learning genetic algorithm (SLGA) to solve the FJSP, which utilizes GA as the fundamental optimization technique and adapts its key parameters intelligently based on RL. The algorithm employs SARSA and QL algorithms as the initial and subsequent learning methods, respectively, and designs state determination and reward methods to realize RL in the GA environment. However, this method only considers C_{max} as the optimization objective.

Li et al. (2022a) propose a multi-objective flexible job-shop scheduling problem with fuzzy processing times, aimed at optimizing the objectives of C_{max} and total machine load. To tackle MOFFJSP, a RL-based MOEA/D algorithm, called RMOEA/D, is presented. In this algorithm, the authors use an initial policy with three rules to obtain high-quality initial populations, propose a parameter adaptive strategy based on QL to guide population selection of the best parameters to increase diversity, design a RL-based variable neighborhood search to lead

solution selection to the correct local search method, and use an elite archive to increase the utilization of discarded historical solutions.

Zhao et al. (2023b) investigate an energy-efficient distributed NFSP with sequence-dependent setup times, aiming to minimize C_{\max} while reducing TEC. They construct a mixed integer linear programming model and propose a cooperative meta-heuristic algorithm based on QL (CMAQ) to solve the problem. In CMAQ, a heuristic method named RNra is proposed to generate initial solutions, and a dual QL-based two-population cooperation framework is designed to further optimize solutions. Based on the energy-efficient DNWFSP-SDST's characteristics, a knowledge-based energy-saving strategy is proposed to improve C_{\max} and TEC. Furthermore, the author proposed a QL-based meta-heuristic algorithm (HHQL) (Zhao et al., 2023a), for solving the energy-efficient distributed blocking FSP (EEDBFSP). This algorithm first uses QL to select appropriate low-level heuristics (LLHs) from a pre-designed LLH set and employs QL for LLH selection based on historical feedback from LLHs. In addition, a concurrent initialization method is proposed to construct the initial population, taking into account both TTD and TEC. The ϵ -greedy strategy is incorporated into the LLH selection process, allowing the utilization of acquired knowledge while ensuring a specific degree of exploration.

Wang and Wang (2021) proposes a CMA with a RL policy agent, along with a MILP, for the energy-aware distributed hybrid flow-shop problem to minimize both the C_{\max} and energy consumption simultaneously. Initially, an encoding scheme and a rational decoding method are devised to account for the trade-off between the two conflicting objectives. Subsequently, two problem specific heuristic methods are introduced for hybrid initialization, aiming to generate diverse solutions. Furthermore, a solution selection method based on decomposition strategy effectively balances convergence and diversity, and the RL policy agent selects appropriate improvement operators to optimize the solutions. Fourth, the algorithm's developmental capability is further enhanced through the integration of reinforced search employing multiple problem-specific operators. Additionally, two energy-saving strategies are devised to improve non-dominated solutions.

Du et al. (2023) introduced a novel approach for solving the FJSP with a time-of-use electricity price constraint. The proposed method, referred to as the estimation of distribution algorithm and deep Q-Network (EDA-DQN), is a hybrid multi-objective optimization algorithm. This problem encompasses various factors such as machine processing speed, setup time, idle time, and transportation time between machines, while simultaneously optimizing both the maximum completion time and total electricity price (TEP). The researchers devised two knowledge-based initialization strategies to enhance performance. The deep Q-network employed 34 state features to illustrate the scheduling situation. Additionally, nine knowledge-based actions were utilized to enhance the scheduling solution, complemented by a reward mechanism aligned with dual objectives. The study effectively showcased the success and performance of the suggested hybrid approach in addressing the integrated FJSP through comprehensive numerical assessments.

Table 3 is a brief overview of the applications of EMOEAs in scheduling problems. Figure 8 shows the taxonomy on the use of EMOEAs.

4.4 The application of EMOEAs in other fields

The application prospects of EMOEAs are extensive. For instance, in the manufacturing industry, it can be utilized to optimize various product parameters to achieve the optimal balance between performance and cost-effectiveness. In the transportation sector, EMOEAs can be employed to plan traffic routes. In healthcare issues, such as predicting patients' disease risks and treatment effects, as well as designing better treatment regimens. In the financial domain, it can be used to optimize investment portfolios to achieve better returns and risk control.

Zhang Z. et al. (2023) have proposed a cost-oriented hybrid model multi-person assembly line balancing approach to address the uncertain demand environment. They have also designed a MOEA based on RL to solve the problem. The algorithm comprises a priority-based solution representation, as well as a new task-worker-sequence decoding approach that considers robustness and idle time reduction. The authors have put forward five types of crossover and three types of mutation operators, with QL strategy determining the crossover and mutation operators for each iteration to effectively obtain the pareto solution set. Finally, a time-based probabilistic adaptive strategy has been devised to efficiently coordinate the crossover and mutation operators. Liu et al. (2022b) proposed a problem decomposition framework to address MOPs, which involves decomposing MOPs into multiple objective knapsack problems and traveling salesman problems (TSP). They employed MOEAs and DRL methods to solve MOKP and TSP.

Huang et al. (2020) put forward an adaptive terrain roughness multi-objective differential evolution algorithm based on information entropy and RL strategies, aimed at addressing the issues of redundant search and mapping imbalance in multi-objective problems. The algorithm estimates the local terrain's unimodal or multi-modal topological structure through information entropy and combines with RL strategies to determine the optimal probability distribution of the algorithm's search strategy set, thus effectively improving the convergence of the search algorithm during the optimization process. Zhang Y. et al. (2021) proposed a multi-objective deep reinforcement learning and EA for complex problems, applied to solve the multi-objective vehicle routing problem with time windows. The algorithm employs a decomposition strategy to generate a set of sub-problems for attention models, and introduces comprehensive contextual information to enhance the attention models. Song et al. (2022) investigated the problem of trajectory control and task offloading in a drone-assisted mobile edge computing system, using an evolutionary multi-objective reinforcement learning (EMORL) algorithm. They improved the original EMORL's multi-task and multi-objective proximal policy optimization by retaining all new learning tasks in the

TABLE 3 The application of EMOEAs in shop scheduling problems.

	Types of shop scheduling	Methodology	Optimization objectives	Ref
ML assists MOEA	Permutation flow-shop	MLL-based hybrid MOEA/D	C_{max} +TTD	Pericleous et al. (2017)
		IG+QL	C_{max}	Karimi-Mamaghan et al. (2023)
		ML-MOMA	C_{max} +total flowtime+TTD	Wang and Tang (2017)
	Flexible job-shop	LRVMA	C_{max} +EC	Li et al. (2023)
		SLGA	C_{max}	Chen et al. (2020)
		MOMA-LD	C_{max} +MW+TMW	Lou et al. (2022)
	Job-shop	FEM	C_{max} +TT+TIT+TWC+TEC	Li et al. (2021b)
	Hybrid workshop	QHH-BS	C_{max} +TEC	Cheng et al. (2022)
	Distributed Shop	CMA +adapted NEH+QL	C_{max} +TEC	Wang et al. (2023a)
		RLBSO	C_{max} +TEC	Zhao et al. (2023b)
CWWO+RL		C_{max}	Zhao et al. (2021b)	
The collaboration between MOEA and ML	Flexible job-shop	Knowledge-based EDA-DQN	C_{max} +TEP	Du et al. (2023)
		RL-based MOEA/D	C_{max} +TMW	Li et al. (2022a)
	Distributed shop	CMAQ	C_{max} +TEC	Zhao et al. (2023b)
		CMA + RL	C_{max} +TEC	Wang and Wang (2021)
		HHQL	TEC+TTD	Zhao et al. (2023a)

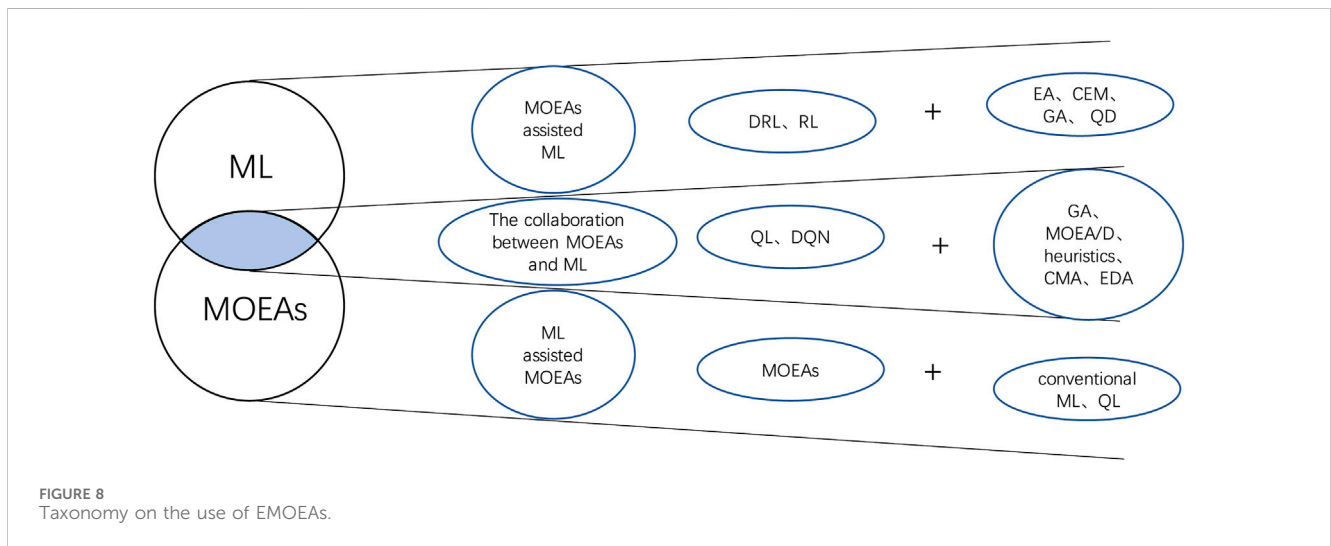


FIGURE 8 Taxonomy on the use of EMOEAs.

descendant population. Wang M. et al. (2023) employed a MOEA combines decomposition and harris hawks learning for medical ML, and applied it to medical cancer gene expression datasets as well as clinical data for lupus nephritis and pulmonary arterial hypertension.

In its early stages, the evolution of EMOEAs has already demonstrated significant potential across diverse fields. Through thorough exploration of algorithms and their applications, one can anticipate a forthcoming wave of innovation and development.

EMOEAs are poised to become potent instruments for optimizing a range of practical problems, thereby driving progress and fostering innovation in relevant domains.

5 Conclusion

Despite being combined with ML, particularly the popular RL in recent years, the capability of MOEAs for optimizing multiple

objectives has not been fully demonstrated (Zhan et al., 2023), indicating significant room for improvement in MOEAs driven by RL. This paper focuses on presenting recent achievements in the field of shop scheduling and presents potential challenges that shop scheduling might face in the future:

First and foremost, research involving the utilization of EMOEAs to address shop scheduling problems remains in its nascent stage. Thus far, majority efforts have been largely confined to theoretical exploration, with limited instances in widespread implementation within actual production systems. Additionally, both MOEAs and ML methods have yet to receive exhaustive investigation in theory. The foundational theoretical research pertaining to EMOEAs presents a formidable challenge (Chai et al., 2013). Different categories of knowledge acquired during the evolutionary process may prove intricate to mathematically define, posing an imminent issue demanding resolution for practical applications (Liu et al., 2023).

Secondly, EMOEAs face challenges related to interpretability and robustness. Shop scheduling involves optimizing multiple objectives, such as minimizing production time and maximizing resource utilization. However, real-world shop scheduling data often exhibits complexity, variability, noise, and uncertainty. These factors can lead to ML models being influenced by biased approximations. Therefore, it is a significant undertaking to make significant progress in gaining a more profound comprehension of the ML models that can efficaciously contribute to evolutionary work in scheduling problems. Additionally, being able to provide a clear explanation for the strong performance of a chosen ML model in specific real-world scheduling applications is also a substantial endeavor.

Thirdly, with the emergence of EMOEAs, the issues surrounding fairness and verifiability in comparing their performance have become even more pronounced (Ishibuchi et al., 2022). When appraising the performance of MOEAs, ensuring absolute equity has grown exceedingly intricate due to the myriad of factors necessitating careful consideration. These factors encompass varying population sizes, a diverse array of testing problems, multiple performance evaluation indicators, and fluctuations in the conditions for algorithm termination, potentially leading to wholly disparate performance comparison outcomes.

Furthermore, the current majority of ML models often exhibit challenges in achieving comprehensive explication of their performance. Given this context, in light of these circumstances, determining how to rigorously verify and analyze the performance of emerging EMOEAs holds paramount significance in a scientifically rigorous manner (Osaba et al., 2021).

Finally, in distributed shop scheduling, there are multiple objectives that need to be optimized, such as minimizing production costs, maximizing production efficiency, and minimizing production cycles. These objectives may have mutual constraints and conflicts, and achieving a balance and trade-off among them is a challenging problem. The application of RL to the problem of multi-objective optimization in distributed shop scheduling holds significant potential for many applications and research endeavors. It has the capability to offer enhanced decision support for industrial production by enabling more intelligent and efficient processes.

Moreover, current research often only focuses on the optimization of machine resources, neglecting the impact of worker resource allocation on production efficiency. In fact, shop

scheduling problems not only involve the allocation of machine resources but also require consideration of worker allocation and scheduling because machines need to be operated by workers. Therefore, future research needs to further consider the issue of dual resource constraints for machines and workers.

To advance the development of EMOEAs for addressing application problems in scheduling, the following potential directions and future research opportunities are presented below.

- The optimization of EMOEAs is a crucial research direction. This involves integrating high-performing ML models that excel in both performance and computational economy with MOEAs. Rather than just emulating prevalent ML models, the focus is on the integration of these models. This integration aims to enhance efficiency and effectiveness. Mitigating computational costs is a key consideration. This includes not only the computational expenditure of evaluating ML models but also the fine-tuning process of the learning models themselves. Consequently, finding the optimal equilibrium between learning and optimization becomes a subject worthy of profound investigation in future research.
- Applying MOEAs to assist ML in the context of shop scheduling is also a burgeoning research direction. Within ML, the quality of data and the selection of features are pivotal to model performance. MOEAs prove instrumental in discerning optimal feature subsets, thereby curtailing data dimensions and augmenting both training efficiency and model efficacy (Zhou et al., 2021). In the ML domain, hyperparameters wield substantial influence over model performance. Employing MOEAs facilitates the automated quest for fitting hyperparameter combinations that strike a balance among diverse performance indicators (Liu and Jin, 2019). Concerning model selection, MOEAs serve to guide the curation of the most suitable ML model for specific tasks. By optimizing multiple indicators, models that excel across various performance facets can be ascertained (Yang et al., 2018). In real-world datasets, the ubiquity of noise necessitates models fortified by MOEAs, enabling commendable performance amidst noisy data realms. For voluminous datasets, MOEAs aid in selecting the most representative sub-samples, thus mitigating training time while upholding model performance. Moreover, they engender equilibrium across multiple performance benchmarks and contribute to enhancing model interpretability to a significant degree.
- Technological advancements have the potential to greatly enhance the effectiveness of Evolutionary EMOEAs when used to the resolution of scheduling problems (Tan and Ding, 2015). Technological advancements have the potential to greatly enhance the effectiveness of EMOEAs when used to the resolution of scheduling problems (Li et al., 2020a). To achieve better outcomes in complex and dynamic real-world scheduling environments, the development of tailored hardware and software solutions can be considered. Lastly, automated design also stands as a crucial direction driving the advancement of EMOEAs (Yi et al., 2023). Through automated design, it is possible to further elevate the performance of EMOEAs, reduce manual intervention, and expedite the optimization process.

The objective of this work is to examine and analyze efficient methodologies for resolving real-world scheduling challenges, with a specific focus on the utilization of EMOEAs. In comparison to traditional MOEAs, EMOEAs have exhibited significant potential and competitiveness. The paper commences with a straightforward shop scheduling problem and subsequently extends its scope to encompass various diverse scheduling issues. Then, by contrasting conventional MOEAs with recent EMOEAs, the application of EMOEAs is expounded upon from three perspectives: ML assists MOEAs, MOEAs assists ML, and the collaboration between MOEAs and ML. Finally, an exploration of potential challenges that EMOEAs might encounter in future applications is undertaken, followed by the presentation of several prospective research directions. These directions aim to propel the further advancement of EMOEAs in the realm of shop scheduling problems.

Author contributions

WZ: Conceptualization, Funding acquisition, Methodology, Project administration, Supervision, Writing–review and editing. GX: Conceptualization, Data curation, Investigation, Methodology, Writing–original draft, Writing–review and editing. MG: Conceptualization, Formal Analysis, Funding acquisition, Investigation, Writing–review and editing. HG: Methodology, Writing–review and editing. XW: Methodology, Writing–review and editing. MD: Funding acquisition, Validation, Writing–review and editing. GZ: Conceptualization, Funding acquisition, Validation, Writing–review and editing.

References

- Abdi, H., and Williams, L. J. (2010). Principal component analysis. *WIREs Comput. Stat.* 2, 433–459. doi:10.1002/wics.101
- Abualigah, L., and Diabat, A. (2021). A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments. *Clust. Comput.* 24, 205–223. doi:10.1007/s10586-020-03075-5
- Ahmadian, M. M., Khatami, M., Salehipour, A., and Cheng, T. (2021). Four decades of research on the open-shop scheduling problem to minimize the makespan. *Eur. J. Operational Res.* 295, 399–426. doi:10.1016/j.ejor.2021.03.026
- Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). Deep reinforcement learning: a brief survey. *IEEE Signal Process. Mag.* 34, 26–38. doi:10.1109/msp.2017.2743240
- Berlińska, J., and Przybylski, B. (2021). Scheduling for gathering multitype data with local computations. *Eur. J. Operational Res.* 294, 453–459. doi:10.1016/j.ejor.2021.01.043
- Beume, N., Naujoks, B., and Emmerich, M. (2007). Sms-emoa: multiobjective selection based on dominated hypervolume. *Eur. J. Operational Res.* 181, 1653–1669. doi:10.1016/j.ejor.2006.08.008
- Beyer, H.-G., and Schwefel, H.-P. (2002). Evolution strategies—a comprehensive introduction. *Nat. Comput.* 1, 3–52. doi:10.1023/a:1015059928466
- Bodnar, C., Day, B., and Lió, P. (2020). Proximal distilled evolutionary reinforcement learning. *Proc. AAAI Conf. Artif. Intell.* 34, 3283–3290. doi:10.1609/aaai.v34i04.5728
- Booker, L. B., Goldberg, D. E., and Holland, J. H. (1989). Classifier systems and genetic algorithms. *Artif. Intell.* 40, 235–282. doi:10.1016/0004-3702(89)90050-7
- Botta-Genoulaz, V. (2000). Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness. *Int. J. Prod. Econ.* 64, 101–111. doi:10.1016/s0925-5273(99)00048-1
- Brik, B., Bettayeb, B., Sahnoun, M., and Duval, F. (2019). Towards predicting system disruption in industry 4.0: machine learning-based approach. *Procedia Comput. Sci.* 151, 667–674. doi:10.1016/j.procs.2019.04.089
- Brucker, P., and Schlie, R. (1990). Job-shop scheduling with multipurpose machines. *Computing* 45, 369–375. doi:10.1007/bf02238804
- Cai, J., Zhou, R., and Lei, D. (2020). Dynamic shuffled frog-leaping algorithm for distributed hybrid flow shop scheduling with multiprocessor tasks. *Eng. Appl. Artif. Intell.* 90, 103540. doi:10.1016/j.engappai.2020.103540
- Cai, S., Yang, K., and Liu, K. (2018). Multi-objective optimization of the distributed permutation flow shop scheduling problem with transportation and eligibility constraints. *J. Operations Res. Soc. China* 6, 391–416. doi:10.1007/s40305-017-0165-3
- Chai, T., Jin, Y., and Sendhoff, B. (2013). Evolutionary complex engineering optimization: opportunities and challenges [guest editorial]. *IEEE Comput. Intell. Mag.* 8, 12–15. doi:10.1109/mci.2013.2264563
- Chamnanlor, C., Sethanan, K., Gen, M., and Chien, C.-F. (2017). Embedding ant system in genetic algorithm for re-entrant hybrid flow shop scheduling problems with time window constraints. *J. Intelligent Manuf.* 28, 1915–1931. doi:10.1007/s10845-015-1078-9
- Chen, R., Yang, B., Li, S., and Wang, S. (2020). A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem. *Comput. Industrial Eng.* 149, 106778. doi:10.1016/j.cie.2020.106778
- Cheng, L., Tang, Q., Zhang, L., and Zhang, Z. (2022). Multi-objective q-learning-based hyper-heuristic with bi-criteria selection for energy-aware mixed shop scheduling. *Swarm Evol. Comput.* 69, 100985. doi:10.1016/j.swevo.2021.100985
- Cheng, R., Gen, M., and Tsujimura, Y. (1996). A tutorial survey of job-shop scheduling problems using genetic algorithms—I. representation. *Comput. Industrial Eng.* 30, 983–997. doi:10.1016/0360-8352(96)00047-2
- Cheng, R., Gen, M., and Tsujimura, Y. (1999). A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies. *Comput. Industrial Eng.* 36, 343–364. doi:10.1016/s0360-8352(99)00136-9
- Cheung, Y.-M., Gu, F., and Liu, H.-L. (2016). Objective extraction for many-objective optimization problems: algorithm and test problems. *IEEE Trans. Evol. Comput.* 20, 755–772. doi:10.1109/tevc.2016.2519758

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. National Natural Science Foundation of China (62276091, U1904167), Science & Technology Research Project of Henan Province (232102211049, 222102210140), Key Research and Development Special Program of Henan Province (231111221200), Zhengzhou Science and Technology Collaborative Innovation Project (21ZZXTCX19), Innovative Research Team (in Science and Technology) in University of Henan Province (21IRTSTHN018), and Scientific Research (C) of Japan Society of Promotion of Science (JSPS) (19K12148).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- Chou, C.-W., Chien, C.-F., and Gen, M. (2014). A multiobjective hybrid genetic algorithm for tft-lcd module assembly scheduling. *IEEE Trans. Automation Sci. Eng.* 11, 692–705. doi:10.1109/tase.2014.2316193
- Cideron, G., Pierrot, T., Perrin, N., Beguir, K., and Sigaud, O. (2020). *Qd-rl: efficient mixing of quality and diversity in reinforcement learning*, 28–73. *arXiv preprint arXiv:2006.08505*.
- Cleveland, W. S., Grosse, E., and Shyu, W. M. (2017). “Local regression models,” in *Statistical models in S* (London, New York: Routledge), 309–376.
- Reyes-Sierra, M., and Coello Coello, C. A. (2006). Multi-objective particle swarm optimizers: a survey of the state-of-the-art. *Int. J. Comput. Intell. Res.* 2, 287–308. doi:10.5019/ijcir.2006.68
- Dai, M., Tang, D., Giret, A., and Salido, M. A. (2019). Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints. *Robotics Computer-Integrated Manuf.* 59, 143–157. doi:10.1016/j.rcim.2019.04.006
- Deb, K. (2011). “Multi-objective optimisation using evolutionary algorithms: an introduction,” in *Multi-objective evolutionary optimisation for product design and manufacturing* (Springer), 3–34.
- Deb, K., and Gupta, H. (2005). “Searching for robust pareto-optimal solutions in multi-objective optimization,” in International Conference on Evolutionary Multi-Criterion Optimization, Guanajuato, Mexico, 9–11 March, 2005 (Springer), 150–164.
- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: nsga-ii. *IEEE Trans. Evol. Comput.* 6, 182–197. doi:10.1109/4235.996017
- De Giovanni, L., and Pezzella, F. (2010). An improved genetic algorithm for the distributed and flexible job-shop scheduling problem. *Eur. J. Operational Res.* 200, 395–408. doi:10.1016/j.ejor.2009.01.008
- Deng, J., and Wang, L. (2017). A competitive memetic algorithm for multi-objective distributed permutation flow shop scheduling problem. *Swarm Evol. Comput.* 32, 121–131. doi:10.1016/j.swevo.2016.06.002
- Du, Y., Li, J.-q., Chen, X.-l., Duan, P.-y., and Pan, Q.-k. (2023). Knowledge-based reinforcement learning and estimation of distribution algorithm for flexible job shop scheduling problem. *IEEE Trans. Emerg. Top. Comput. Intell.* 7, 1036–1050. doi:10.1109/tetci.2022.3145706
- Dugardin, F., Yalaoui, F., and Amodeo, L. (2010). New multi-objective method to solve reentrant hybrid flow shop scheduling problem. *Eur. J. Operational Res.* 203, 22–31. doi:10.1016/j.ejor.2009.06.031
- Emami, S., Sabbagh, M., and Moslehi, G. (2016). A Lagrangian relaxation algorithm for order acceptance and scheduling problem: a globalised robust optimisation approach. *Int. J. Comput. Integr. Manuf.* 29, 535–560. doi:10.1080/0951192x.2015.1068452
- Engin, O., and Güçlü, A. (2018). A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems. *Appl. Soft Comput.* 72, 166–176. doi:10.1016/j.asoc.2018.08.002
- Espejo, P. G., Ventura, S., and Herrera, F. (2010). A survey on the application of genetic programming to classification. *IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev.* 40, 121–144. doi:10.1109/tsmcc.2009.2033566
- Fattahi, P., Saidi Mehrabad, M., and Jolai, F. (2007). Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. *J. Intelligent Manuf.* 18, 331–342. doi:10.1007/s10845-007-0026-8
- Fernandez-Viagas, V., Perez-Gonzalez, P., and Framinan, J. M. (2019). Efficiency of the solution representations for the hybrid flow shop scheduling problem with makespan objective. *Comput. Operations Res.* 109, 77–88. doi:10.1016/j.cor.2019.05.002
- Fernandez-Viagas, V., Ruiz, R., and Framinan, J. M. (2017). A new vision of approximate methods for the permutation flowshop to minimise makespan: state-of-the-art and computational evaluation. *Eur. J. Operational Res.* 257, 707–721. doi:10.1016/j.ejor.2016.09.055
- Floudas, C. A., and Lin, X. (2005). Mixed integer linear programming in process scheduling: modeling, algorithms, and applications. *Ann. Operations Res.* 139, 131–162. doi:10.1007/s10479-005-3446-x
- Galiana, F. D., Bouffard, F., Arroyo, J. M., and Restrepo, J. F. (2005). Scheduling and pricing of coupled energy and primary, secondary, and tertiary reserves. *Proc. IEEE* 93, 1970–1983. doi:10.1109/jproc.2005.857492
- Gao, J., Gen, M., Sun, L., and Zhao, X. (2007). A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems. *Comput. Industrial Eng.* 53, 149–162. doi:10.1016/j.cie.2007.04.010
- Gao, J., Sun, L., and Gen, M. (2008). A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. *Comput. Operations Res.* 35, 2892–2907. doi:10.1016/j.cor.2007.01.001
- Gao, K., Huang, Y., Sadollah, A., and Wang, L. (2020). A review of energy-efficient scheduling in intelligent production systems. *Complex and Intelligent Syst.* 6, 237–249. doi:10.1007/s40747-019-00122-6
- Gen, M., and Cheng, R. (2000). *Genetic algorithms and engineering optimization*, vol. 7. John Wiley and Sons.
- Gen, M., Cheng, R., and Lin, L. (2008). *Network models and optimization: multiobjective genetic algorithm approach*. Springer Science and Business Media.
- Gen, M., Gao, J., and Lin, L. (2009). Multistage-based genetic algorithm for flexible job-shop scheduling problem. *Intelligent Evol. Syst. IES* 187, 183–196.
- Goli, A., Ala, A., and Hajiaghayi-Keshтели, M. (2023). Efficient multi-objective meta-heuristic algorithms for energy-aware non-permutation flow-shop scheduling problem. *Expert Syst. Appl.* 213, 119077. doi:10.1016/j.eswa.2022.119077
- Goli, A., Babaei Tirkolae, E., and Soltani, M. (2019). A robust just-in-time flow shop scheduling problem with outsourcing option on subcontractors. *Prod. Manuf. Res.* 7, 294–315. doi:10.1080/21693277.2019.1620651
- Gong, G., Chiong, R., Deng, Q., and Gong, X. (2020). A hybrid artificial bee colony algorithm for flexible job shop scheduling with worker flexibility. *Int. J. Prod. Res.* 58, 4406–4420. doi:10.1080/00207543.2019.1653504
- Gu, Z.-M., and Wang, G.-G. (2020). Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization. *Future Gener. Comput. Syst.* 107, 49–69. doi:10.1016/j.future.2020.01.048
- Hameed, M. S. A., and Schwung, A. (2020). *Reinforcement learning on job shop scheduling problems using graph networks*. *arXiv preprint arXiv:2009.03836*.
- Han, W., Deng, Q., Gong, G., Zhang, L., and Luo, Q. (2021a). Multi-objective evolutionary algorithms with heuristic decoding for hybrid flow shop scheduling problem with worker constraint. *Expert Syst. Appl.* 168, 114282. doi:10.1016/j.eswa.2020.114282
- Han, X., Han, Y., Chen, Q., Li, J., Sang, H., Liu, Y., et al. (2021b). Distributed flow shop scheduling with sequence-dependent setup times using an improved iterated greedy algorithm. *Complex Syst. Model. Simul.* 1, 198–217. doi:10.23919/csms.2021.0018
- Hao, J.-H., Li, J.-Q., Du, Y., Song, M.-X., Duan, P., and Zhang, Y.-Y. (2019). Solving distributed hybrid flowshop scheduling problems by a hybrid brain storm optimization algorithm. *IEEE Access* 7, 66879–66894. doi:10.1109/access.2019.2917273
- Hao, X., Gen, M., Lin, L., and Suer, G. A. (2017). Effective multiobjective EDA for bi-criteria stochastic job-shop scheduling problem. *J. Intelligent Manuf.* 28, 833–845. doi:10.1007/s10845-014-1026-0
- Heckerman, D. (2008). A tutorial on learning with bayesian networks. *Innovations Bayesian Netw. Theory Appl.*, 33–82.
- Hoogeveen, J. A., Lenstra, J. K., and Veltman, B. (1996). Preemptive scheduling in a two-stage multiprocessor flow shop is np-hard. *Eur. J. Operational Res.* 89, 172–175. doi:10.1016/0377-2217(94)00235-5
- Huang, Y., Li, W., Tian, F., and Meng, X. (2020). A fitness landscape ruggedness multiobjective differential evolution algorithm with a reinforcement learning strategy. *Appl. Soft Comput.* 96, 106693. doi:10.1016/j.asoc.2020.106693
- Ishibuchi, H., Pang, L. M., and Shang, K. (2022). “Difficulties in fair performance comparison of multiobjective evolutionary algorithms,” in Proceedings of the Genetic and Evolutionary Computation Conference Companion, Boston, Massachusetts, July 9–13, 2022, 937–957.
- Ismayilov, G., and Topcuoglu, H. R. (2020). Neural network based multi-objective evolutionary algorithm for dynamic workflow scheduling in cloud computing. *Future Gener. Comput. Syst.* 102, 307–322. doi:10.1016/j.future.2019.08.012
- Jamrus, T., Chien, C.-F., Gen, M., and Sethanan, K. (2015). Multistage production distribution under uncertain demands with integrated discrete particle swarm optimization and extended priority-based hybrid genetic algorithm. *Fuzzy Optim. Decis. Mak.* 14, 265–287. doi:10.1007/s10700-014-9200-6
- Jiang, E. d., Wang, L., and Peng, Z. (2020). Solving energy-efficient distributed job shop scheduling via multi-objective evolutionary algorithm with decomposition. *Swarm Evol. Comput.* 58, 100745. doi:10.1016/j.swevo.2020.100745
- Jiang, T., Zhang, C., and Sun, Q.-M. (2019). Green job shop scheduling problem with discrete whale optimization algorithm. *IEEE Access* 7, 43153–43166. doi:10.1109/access.2019.2908200
- Jordan, L., Dhaenens, C., and Talbi, E.-G. (2006). “Using datamining techniques to help metaheuristics: a short survey,” in Hybrid Metaheuristics: Third International Workshop, HM 2006, Gran Canaria, Spain, October 13–14, 2006 (Springer), 57–69.
- Karimi-Mamaghan, M., Mohammadi, M., Pasdeloup, B., and Meyer, P. (2023). Learning to select operators in meta-heuristics: an integration of q-learning into the iterated greedy algorithm for the permutation flowshop scheduling problem. *Eur. J. Operational Res.* 304, 1296–1330. doi:10.1016/j.ejor.2022.03.054
- Katoch, S., Chauhan, S. S., and Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools Appl.* 80, 8091–8126. doi:10.1007/s11042-020-10139-6
- Kelley Jr, J. E. (1960). The cutting-plane method for solving convex programs. *J. Soc. Industrial Appl. Math.* 8, 703–712. doi:10.1137/0108053
- Kenné, J. P., and Gharbi, A. (2000). Production planning problem in manufacturing systems with general failure and repair time distributions. *Prod. Plan. Control* 11, 581–588. doi:10.1080/095372800414142
- Kennedy, J., and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN’95-international Conference on neural networks*. (IEEE) 4, 1942–1948. doi:10.1109/ICNN.1995.488968

- Khadka, S., Majumdar, S., Nassar, T., Dwiell, Z., Tumer, E., Miret, S., et al. (2019). "Collaborative evolutionary reinforcement learning," in International Conference on Machine Learning (PMLR), Long Beach, CA, USA, June 10-15, 2019, 3341–3350.
- Khadka, S., and Tumer, K. (2018). Evolution-guided policy gradient in reinforcement learning. *Adv. Neural Inf. Process. Syst.* 31.
- Kolodner, J. L. (1992). An introduction to case-based reasoning. *Artif. Intell. Rev.* 6, 3–34. doi:10.1007/bf00155578
- Kotsiantis, S. B., Zaharakis, I., and Pintelas, P. (2007). Supervised machine learning: a review of classification techniques. *Emerg. Artif. Intell. Appl. Comput. Eng.* 160, 3–24.
- Koulamas, C., and Kyriaris, G. J. (2021). The no-wait flow shop with rejection. *Int. J. Prod. Res.* 59, 1852–1859. doi:10.1080/00207543.2020.1727042
- Kundakci, N., and Kulak, O. (2016). Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem. *Comput. Industrial Eng.* 96, 31–51. doi:10.1016/j.cie.2016.03.011
- Lang, S., Behrendt, F., Lanzerath, N., Reggelin, T., and Müller, M. (2020). "Integration of deep reinforcement learning and discrete-event simulation for real-time scheduling of a flexible job shop production," in 2020 winter simulation conference (WSC) (IEEE), 3057–3068.
- Langdon, W. B., and Poli, R. (2013). *Foundations of genetic programming*. Springer Science and Business Media.
- Lei, D. (2009). Multi-objective production scheduling: a survey. *Int. J. Adv. Manuf. Technol.* 43, 926–938. doi:10.1007/s00170-008-1770-4
- Lei, D., and Liu, M. (2020). An artificial bee colony with division for distributed unrelated parallel machine scheduling with preventive maintenance. *Comput. Industrial Eng.* 141, 106320. doi:10.1016/j.cie.2020.106320
- Lei, D., and Su, B. (2023). A multi-class teaching-learning-based optimization for multi-objective distributed hybrid flow shop scheduling. *Knowledge-Based Syst.* 263, 110252. doi:10.1016/j.knsys.2023.110252
- Lei, D., and Wang, T. (2020). Solving distributed two-stage hybrid flowshop scheduling using a shuffled frog-leaping algorithm with memplex grouping. *Eng. Optim.* 52, 1461–1474. doi:10.1080/0305215x.2019.1674295
- Leng, J., Wang, X., Wu, S., Jin, C., Tang, M., Liu, R., et al. (2022). A multi-objective reinforcement learning approach for resequencing scheduling problems in automotive manufacturing systems. *Int. J. Prod. Res.* 61, 5156–5175. doi:10.1080/00207543.2022.2098871
- Li, B., Li, J., Tang, K., and Yao, X. (2014). "An improved two archive algorithm for many-objective optimization," in 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 06-11 July 2014 (IEEE), 2869–2876.
- Li, J., Li, J., Zhang, L., Sang, H., Han, Y., and Chen, Q. (2021a). Solving type-2 fuzzy distributed hybrid flowshop scheduling using an improved brain storm optimization algorithm. *Int. J. Fuzzy Syst.* 23, 1194–1212. doi:10.1007/s40815-021-01050-9
- Li, J.-Q., and Pan, Q.-k. (2015). Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm. *Inf. Sci.* 316, 487–502. doi:10.1016/j.ins.2014.10.009
- Li, J.-Y., Zhan, Z.-H., Liu, R.-D., Wang, C., Kwong, S., and Zhang, J. (2021). Generation-level parallelism for evolutionary computation: a pipeline-based parallel particle swarm optimization. *IEEE Trans. Cybern.* 51, 4848–4859. doi:10.1109/tycb.2020.3028070
- Li, M., and Wang, G.-G. (2022). A review of green shop scheduling problem. *Inf. Sci.* 589, 478–496. doi:10.1016/j.ins.2021.12.122
- Li, R., Gong, W., and Lu, C. (2022a). A reinforcement learning based RMOEA/D for bi-objective fuzzy flexible job shop scheduling. *Expert Syst. Appl.* 203, 117380. doi:10.1016/j.eswa.2022.117380
- Li, R., Gong, W., Lu, C., and Wang, L. (2023). A learning-based memetic algorithm for energy-efficient flexible job shop scheduling with type-2 fuzzy processing time. *IEEE Trans. Evol. Comput.* 27, 610–620. doi:10.1109/tevc.2022.3175832
- Li, R., Gong, W., Wang, L., Lu, C., and Jiang, S. (2022b). Two-stage knowledge-driven evolutionary algorithm for distributed green flexible job shop scheduling with type-2 fuzzy processing time. *Swarm Evol. Comput.* 74, 101139. doi:10.1016/j.swevo.2022.101139
- Li, W., He, L., and Cao, Y. (2022). Many-objective evolutionary algorithm with reference point-based fuzzy correlation entropy for energy-efficient job shop scheduling with limited workers. *IEEE Trans. Cybern.* 52, 10721–10734. doi:10.1109/tycb.2021.3069184
- Li, W., Wang, G.-G., and Gandomi, A. H. (2021c). A survey of learning-based intelligent optimization algorithms. *Archives Comput. Methods Eng.* 28, 3781–3799. doi:10.1007/s11831-021-09562-1
- Li, X., Guo, X., Tang, H., Wu, R., Wang, L., Pang, S., et al. (2022c). Survey of integrated flexible job shop scheduling problems. *Comput. Industrial Eng.* 174, 108786. doi:10.1016/j.cie.2022.108786
- Li, Y., Li, F., Pan, Q.-K., Gao, L., and Tasgetiren, M. F. (2019a). An artificial bee colony algorithm for the distributed hybrid flowshop scheduling problem. *Procedia Manuf.* 39, 1158–1166. doi:10.1016/j.promfg.2020.01.354
- Li, Y., Li, X., Gao, L., and Meng, L. (2020b). An improved artificial bee colony algorithm for distributed heterogeneous hybrid flowshop scheduling problem with sequence-dependent setup times. *Comput. Industrial Eng.* 147, 106638. doi:10.1016/j.cie.2020.106638
- Li, Y., Li, X., Gao, L., Zhang, B., Pan, Q.-K., Tasgetiren, M. F., et al. (2021d). A discrete artificial bee colony algorithm for distributed hybrid flowshop scheduling problem with sequence-dependent setup times. *Int. J. Prod. Res.* 59, 3880–3899. doi:10.1080/00207543.2020.1753897
- Li, Z., Qian, B., Hu, R., Chang, L., and Yang, J. (2019b). An elitist nondominated sorting hybrid algorithm for multi-objective flexible job-shop scheduling problem with sequence-dependent setups. *Knowledge-Based Syst.* 173, 83–112. doi:10.1016/j.knsys.2019.02.027
- Lin, L., and Gen, M. (2018). Hybrid evolutionary optimisation with learning for production scheduling: state-of-the-art survey on algorithms and applications. *Int. J. Prod. Res.* 56, 193–223. doi:10.1080/00207543.2018.1437288
- Liu, J., and Jin, Y. (2019). "Evolving hyperparameters for training deep neural networks against adversarial attacks," in 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 06-09 December 2019 (IEEE), 1778–1785.
- Liu, R., Piplani, R., and Toro, C. (2022a). Deep reinforcement learning for dynamic scheduling of a flexible job shop. *Int. J. Prod. Res.* 60, 4049–4069. doi:10.1080/00207543.2022.2058432
- Liu, S., Lin, Q., Li, J., and Tan, K. C. (2023). A survey on learnable evolutionary algorithms for scalable multiobjective optimization. *IEEE Trans. Evol. Comput.* 27, 1941–1961. doi:10.1109/tevc.2023.3250350
- Liu, W., Dridi, M., Fei, H., and El Hassani, A. H. (2021). Hybrid metaheuristics for solving a home health care routing and scheduling problem with time windows, synchronized visits and lunch breaks. *Expert Syst. Appl.* 183, 115307. doi:10.1016/j.eswa.2021.115307
- Liu, W., Wang, R., Zhang, T., Li, K., Li, W., Ishibuchi, H., et al. (2023). Hybridization of evolutionary algorithm and deep reinforcement learning for multi-objective orienteering optimization. *IEEE Trans. Evol. Comput.* 27, 1260–1274. doi:10.1109/tevc.2022.3199045
- Liu, X.-F., Zhou, Y.-R., Yu, X., and Lin, Y. (2019). Dual-archive-based particle swarm optimization for dynamic optimization. *Appl. Soft Comput.* 85, 105876. doi:10.1016/j.asoc.2019.105876
- Lopes, F., Agnelo, J., Teixeira, C. A., Laranjeiro, N., and Bernardino, J. (2020). Automating orthogonal defect classification using machine learning algorithms. *Future Gener. Comput. Syst.* 102, 932–947. doi:10.1016/j.future.2019.09.009
- Lou, H., Wang, X., Dong, Z., and Yang, Y. (2022). Memetic algorithm based on learning and decomposition for multiobjective flexible job shop scheduling considering human factors. *Swarm Evol. Comput.* 75, 101204. doi:10.1016/j.swevo.2022.101204
- Luo, P. C., Xiong, H. Q., Zhang, B. W., Peng, J. Y., and Xiong, Z. F. (2022a). Multi-resource constrained dynamic workshop scheduling based on proximal policy optimisation. *Int. J. Prod. Res.* 60, 5937–5955. doi:10.1080/00207543.2021.1975057
- Luo, Q., Deng, Q., Gong, G., Guo, X., and Liu, X. (2022b). A distributed flexible job shop scheduling problem considering worker arrangement using an improved memetic algorithm. *Expert Syst. Appl.* 207, 117984. doi:10.1016/j.eswa.2022.117984
- Luo, S. (2020). Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. *Appl. Soft Comput.* 91, 106208. doi:10.1016/j.asoc.2020.106208
- Luo, S., Zhang, L., and Fan, Y. (2021a). Dynamic multi-objective scheduling for flexible job shop by deep reinforcement learning. *Comput. Industrial Eng.* 159, 107489. doi:10.1016/j.cie.2021.107489
- Luo, S., Zhang, L., and Fan, Y. (2022). Real-time scheduling for dynamic partial-no-wait multiobjective flexible job shop by deep reinforcement learning. *IEEE Trans. Automation Sci. Eng.* 19, 3020–3038. doi:10.1109/tase.2021.3104716
- Mahdavi, S., Rahnamayan, S., and Deb, K. (2018). Opposition based learning: a literature review. *Swarm Evol. Comput.* 39, 1–23. doi:10.1016/j.swevo.2017.09.010
- Makhadmeh, S. N., Khader, A. T., Al-Betar, M. A., Naim, S., Abasi, A. K., and Alyasseri, Z. A. A. (2019). Optimization methods for power scheduling problems in smart home: survey. *Renew. Sustain. Energy Rev.* 115, 109362. doi:10.1016/j.rser.2019.109362
- Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: concerns and ways forward. *PLoS One* 13, e0194889. doi:10.1371/journal.pone.0194889
- Marchesini, E., Corsi, D., and Farinelli, A. (2021). "Genetic soft updates for policy evolution in deep reinforcement learning," in International Conference on Learning Representations, Austria, May 3-7, 2021.
- Meng, L., Zhang, C., Ren, Y., Zhang, B., and Lv, C. (2020). Mixed-integer linear programming and constraint programming formulations for solving distributed flexible job shop scheduling problem. *Comput. Industrial Eng.* 142, 106347. doi:10.1016/j.cie.2020.106347

- Miyata, H. H., and Nagano, M. S. (2019). The blocking flow shop scheduling problem: a comprehensive and conceptual review. *Expert Syst. Appl.* 137, 130–156. doi:10.1016/j.eswa.2019.06.069
- Mohan, J., Lanka, K., and Rao, A. N. (2019). A review of dynamic job shop scheduling techniques. *Procedia Manuf.* 30, 34–39. doi:10.1016/j.promfg.2019.02.006
- Moon *, C., Kim, J. S., and Gen, M. (2004). Advanced planning and scheduling based on precedence and resource constraints for e-plant chains. *Int. J. Prod. Res.* 42, 2941–2955. doi:10.1080/00207540410001691956
- Moras, M. d. F., Ribeiro, M. H. D. M., da Silva, R. G., Mariani, V. C., and Coelho, L. d. S. (2022). Discrete differential evolution metaheuristics for permutation flow shop scheduling problems. *Comput. Industrial Eng.* 166, 107956. doi:10.1016/j.cie.2022.107956
- Narendra, and Fukunaga, (1977). A branch and bound algorithm for feature subset selection. *IEEE Trans. Comput.* C-26, 917–922. doi:10.1109/tc.1977.1674939
- Osaba, E., Villar-Rodríguez, E., Del Ser, J., Nebro, A. J., Molina, D., LaTorre, A., et al. (2021). A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems. *Swarm Evol. Comput.* 64, 100888. doi:10.1016/j.swevo.2021.100888
- Pericleous, S., Konstantinidis, A., Achilleos, A., and Papadopoulos, G. A. (2017). “Generic hybridization of moea/d with learning for permutation flow shop scheduling problem,” in 2017 8th International Conference on Information, Intelligence, Systems and Applications (IISA), Larnaca, Cyprus, 27–30 August 2017 (IEEE), 1–6.
- Pezzella, F., Morganti, G., and Ciaschetti, G. (2008). A genetic algorithm for the flexible job-shop scheduling problem. *Comput. Operations Res.* 35, 3202–3212. doi:10.1016/j.cor.2007.02.014
- Pisner, D. A., and Schnyer, D. M. (2020). “Support vector machine,” in *Machine learning* (Elsevier), 101–121.
- Pourchot, A., and Sigaud, O. (2018). *Cem-rl: combining evolutionary and gradient-based methods for policy search*. arXiv preprint arXiv:1810.01222.
- Price, K. V. (2013). “Differential evolution,” in *Handbook of optimization: from classical to modern approach* (Springer), 187–214.
- Ramasesh, R. (1990). Dynamic job shop scheduling: a survey of simulation research. *Omega* 18, 43–57. doi:10.1016/0305-0483(90)90017-4
- Ramya, G., Chandrasekaran, M., and Shankar, E. (2019). Case study analysis of job shop scheduling and its integration with material requirement planning. *Mater. Today Proc.* 16, 1034–1042. doi:10.1016/j.matpr.2019.05.192
- Rumelhart, D. E., and Zipser, D. (1985). Feature discovery by competitive learning. *Cognitive Sci.* 9, 75–112. doi:10.1207/s15516709cog0901_5
- Şahman, M. A. (2021). A discrete spotted hyena optimizer for solving distributed job shop scheduling problems. *Appl. Soft Comput.* 106, 107349. doi:10.1016/j.asoc.2021.107349
- Samsonov, V., Kemmerling, M., Paegert, M., Lütticke, D., Sauermann, F., Gützlaff, A., et al. (2021). “Manufacturing control in job shop environments with reinforcement learning,” in ICAART, 589–597.
- San-José, L. A., Sicilia, J., González-De-la Rosa, M., and Febles-Acosta, J. (2019). Analysis of an inventory system with discrete scheduling period, time-dependent demand and backlogged shortages. *Comput. Operations Res.* 109, 200–208. doi:10.1016/j.cor.2019.05.003
- Schrijver, A. (1998). *Theory of linear and integer programming*. John Wiley and Sons.
- Srich, C. R., Armentano, V. A., and Laguna, M. (2004). Tardiness minimization in a flexible job shop: a tabu search approach. *J. Intelligent Manuf.* 15, 103–115. doi:10.1023/bjmis.0000010078.30713.e9
- Shanker, K., and Tzen, Y.-J. (1985). A loading and dispatching problem in a random flexible manufacturing system. *Int. J. Prod. Res.* 23, 579–595. doi:10.1080/00207548508904730
- Shao, W., Shao, Z., and Pi, D. (2020). Modeling and multi-neighborhood iterated greedy algorithm for distributed hybrid flow shop scheduling problem. *Knowledge-Based Syst.* 194, 105527. doi:10.1016/j.knsys.2020.105527
- Shao, W., Shao, Z., and Pi, D. (2021). Multi-objective evolutionary algorithm based on multiple neighborhoods local search for multi-objective distributed hybrid flow shop scheduling problem. *Expert Syst. Appl.* 183, 115453. doi:10.1016/j.eswa.2021.115453
- Shao, W., Shao, Z., and Pi, D. (2022). A network memetic algorithm for energy and labor-aware distributed heterogeneous hybrid flow shop scheduling problem. *Swarm Evol. Comput.* 75, 101190. doi:10.1016/j.swevo.2022.101190
- Song, F., King, H., Wang, X., Luo, S., Dai, P., Xiao, Z., et al. (2022). Evolutionary multi-objective reinforcement learning based trajectory control and task offloading in uav-assisted mobile edge computing. *IEEE Trans. Mob. Comput.* 22, 7387–7405. doi:10.1109/TMC.2022.3208457
- Song, H.-B., Yang, Y.-H., Lin, J., and Ye, J.-X. (2023). An effective hyper heuristic-based memetic algorithm for the distributed assembly permutation flow-shop scheduling problem. *Appl. Soft Comput.* 135, 110022. doi:10.1016/j.asoc.2023.110022
- Srinivas, N., and Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.* 2, 221–248. doi:10.1162/evco.1994.2.3.221
- Stadler, W. (1979). A survey of multicriteria optimization or the vector maximum problem, part i: 1776–1960. *J. Optim. Theory Appl.* 29, 1–52. doi:10.1007/bf00932634
- Suri, K., Shi, X. Q., Plataniotis, K. N., and Lawryshyn, Y. A. (2020). *Maximum mutation reinforcement learning for scalable control*. arXiv preprint arXiv:2007.13690.
- Tan, W., Yuan, X., Wang, J., and Zhang, X. (2021). A fatigue-conscious dual resource constrained flexible job shop scheduling problem by enhanced nsga-ii: an application from casting workshop. *Comput. Industrial Eng.* 160, 107557. doi:10.1016/j.cie.2021.107557
- Tan, Y., and Ding, K. (2016). A survey on gpu-based implementation of swarm intelligence algorithms. *IEEE Trans. Cybern.* 46, 2028–2041. doi:10.1109/tcyb.2015.2460261
- Tang, X., Cao, W., Tang, H., Deng, T., Mei, J., Liu, Y., et al. (2022). Cost-efficient workflow scheduling algorithm for applications with deadline constraint on heterogeneous clouds. *IEEE Trans. Parallel Distributed Syst.* 33, 2079–2092. doi:10.1109/tpds.2021.3134247
- Tavakkoli-Moghaddam, R., Jolai, F., Vaziri, F., Ahmed, P., and Azaron, A. (2005). A hybrid method for solving stochastic job shop scheduling problems. *Appl. Math. Comput.* 170, 185–206. doi:10.1016/j.amc.2004.11.036
- Tian, Y., Cheng, R., Zhang, X., Cheng, F., and Jin, Y. (2018). An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Trans. Evol. Comput.* 22, 609–622. doi:10.1109/tevc.2017.2749619
- Tian, Y., Zhang, X., Cheng, R., and Jin, Y. (2016). “A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric,” in 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, British Columbia, Canada, 24–29 July 2016 (IEEE), 5222–5229.
- Toptal, A., and Sabuncuoglu, I. (2010). Distributed scheduling: a review of concepts and applications. *Int. J. Prod. Res.* 48, 5235–5262. doi:10.1080/00207540903121065
- Torabi, S., Karimi, B., and Fatemi Ghomi, S. (2005). The common cycle economic lot scheduling in flexible job shops: the finite horizon case. *Int. J. Prod. Econ.* 97, 52–65. doi:10.1016/j.ijpe.2004.05.005
- Trebuña, P., and Halčinová, J. (2013). Mathematical tools of cluster analysis. *Appl. Math.* 04, 814–816. doi:10.4236/am.2013.45111
- Trivedi, A., Srinivasan, D., Sanyal, K., and Ghosh, A. (2016). A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Trans. Evol. Comput.* 21, 440–462. doi:10.1109/TEVC.2016.2608507
- Tsujimura, Y., Gen, M., and Kubota, E. (1995). Solving job-shop scheduling problem with fuzzy processing time using genetic algorithm. *J. Jpn. Soc. Fuzzy Theory Syst.* 7, 1073–1083. doi:10.3156/fuzzy.7.5_1073
- Umam, M. S., Mustafid, M., and Suryono, S. (2022). A hybrid genetic algorithm and tabu search for minimizing makespan in flow shop scheduling problem. *J. King Saud University-Computer Inf. Sci.* 34, 7459–7467. doi:10.1016/j.jksuci.2021.08.025
- Voorneveld, M. (2003). Characterization of pareto dominance. *Operations Res. Lett.* 31, 7–11. doi:10.1016/s0167-6377(02)00189-x
- Wang, G., Li, X., Gao, L., and Li, P. (2021). Energy-efficient distributed heterogeneous welding flow shop scheduling problem using a modified moea/d. *Swarm Evol. Comput.* 62, 100858. doi:10.1016/j.swevo.2021.100858
- Wang, G.-G., Gao, D., and Pedrycz, W. (2022a). Solving multiobjective fuzzy job-shop scheduling problem by a hybrid adaptive differential evolution algorithm. *IEEE Trans. Industrial Inf.* 18, 8519–8528. doi:10.1109/tii.2022.3165636
- Wang, H., Cheng, J., Liu, C., Zhang, Y., Hu, S., and Chen, L. (2022b). Multi-objective reinforcement learning framework for dynamic flexible job shop scheduling problem with uncertain events. *Appl. Soft Comput.* 131, 109717. doi:10.1016/j.asoc.2022.109717
- Wang, J.-J., and Wang, L. (2019). “An iterated greedy algorithm for distributed hybrid flowshop scheduling problem with total tardiness minimization,” in 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), Vancouver, BC, Canada, 22–26 August 2019 (IEEE), 350–355.
- Wang, J.-J., and Wang, L. (2021). A bi-population cooperative memetic algorithm for distributed hybrid flow-shop scheduling. *IEEE Trans. Emerg. Top. Comput. Intell.* 5, 947–961. doi:10.1109/tetci.2020.3022372
- Wang, J.-J., and Wang, L. (2022). A cooperative memetic algorithm with learning-based agent for energy-aware distributed hybrid flow-shop scheduling. *IEEE Trans. Evol. Comput.* 26, 461–475. doi:10.1109/tevc.2021.3106168
- Wang, J.-J., Wang, L., and Xiu, X. (2023a). A cooperative memetic algorithm for energy-aware distributed welding shop scheduling problem. *Eng. Appl. Artif. Intell.* 120, 105877. doi:10.1016/j.engappai.2023.105877
- Wang, M., Li, X., Chen, L., and Chen, H. (2023b). Medical machine learning based on multiobjective evolutionary algorithm using learning decomposition. *Expert Syst. Appl.* 216, 119450. doi:10.1016/j.eswa.2022.119450
- Wang, P., and Tong, X. (2020). A dimension convergence-based evolutionary algorithm for many-objective optimization problems. *IEEE Access* 8, 224631–224642. doi:10.1109/access.2020.3043253
- Wang, X., and Tang, L. (2017). A machine-learning based memetic algorithm for the multi-objective permutation flowshop scheduling problem. *Comput. Operations Res.* 79, 60–77. doi:10.1016/j.cor.2016.10.003

- Watkins, C. J., and Dayan, P. (1992). Q-learning. *Mach. Learn.* 8, 279–292. doi:10.1023/a:1022676722315
- Wiering, M. A., and Van Otterlo, M. (2012). Reinforcement learning. *Adapt. Learn. Optim.* 12, 729.
- Woeginger, G. J. (2003). “Exact algorithms for np-hard problems: a survey,” in *Combinatorial Optimization—Eureka, You Shrink! Papers Dedicated to Jack Edmonds 5th International Workshop, Aussois, France, March 5–9, 2001* (Springer), 185–207.
- Xie, J., Gao, L., Peng, K., Li, X., and Li, H. (2019). Review on flexible job shop scheduling. *IET Collab. Intell. Manuf.* 1, 67–77. doi:10.1049/iet-cim.2018.0009
- Xiong, H., Shi, S., Ren, D., and Hu, J. (2022). A survey of job shop scheduling problem: the types and models. *Comput. Operations Res.* 142, 105731. doi:10.1016/j.cor.2022.105731
- Xu, Y., and Pi, D. (2020). A reinforcement learning-based communication topology in particle swarm optimization. *Neural Comput. Appl.* 32, 10007–10032. doi:10.1007/s00521-019-04527-9
- Yang, C., Ding, J., Jin, Y., Wang, C., and Chai, T. (2019). Multitasking multiobjective evolutionary operational indices optimization of beneficiation processes. *IEEE Trans. Automation Sci. Eng.* 16, 1046–1057. doi:10.1109/tase.2018.2865593
- Yi, J.-H., Xing, L.-N., Wang, G.-G., Dong, J., Vasilakos, A. V., Alavi, A. H., et al. (2020). Behavior of crossover operators in NSGA-III for large-scale optimization problems. *Inf. Sci.* 509, 470–487. doi:10.1016/j.ins.2018.10.005
- Yi, W., Qu, R., Jiao, L., and Niu, B. (2023). Automated design of metaheuristics using reinforcement learning within a novel general search framework. *IEEE Trans. Evol. Comput.* 27, 1072–1084. doi:10.1109/tevc.2022.3197298
- Ying, K.-C., and Lin, S.-W. (2018). Minimizing makespan for the distributed hybrid flowshop scheduling problem with multiprocessor tasks. *Expert Syst. Appl.* 92, 132–141. doi:10.1016/j.eswa.2017.09.032
- Ying, K.-C., Pourhejazy, P., Cheng, C.-Y., and Syu, R.-S. (2023). Supply chain-oriented permutation flowshop scheduling considering flexible assembly and setup times. *Int. J. Prod. Res.* 61, 258–281. doi:10.1080/00207543.2020.1842938
- Yue, L., Wang, H., Mumtaz, J., Rauf, M., and Li, Z. (2023). Energy-efficient scheduling of a two-stage flexible printed circuit board flow shop using a hybrid pareto spider monkey optimisation algorithm. *J. Industrial Inf. Integration* 31, 100412. doi:10.1016/j.jii.2022.100412
- Zhan, Z.-H., Li, J.-Y., Kwong, S., and Zhang, J. (2023). Learning-aided evolution for optimization. *IEEE Trans. Evol. Comput.* 27, 1794–1808. doi:10.1109/tevc.2022.3232776
- Zhang, B., Pan, Q.-k., Meng, L.-l., Lu, C., Mou, J.-h., and Li, J.-q. (2022). An automatic multi-objective evolutionary algorithm for the hybrid flowshop scheduling problem with consistent sublots. *Knowledge-Based Syst.* 238, 107819. doi:10.1016/j.knsys.2021.107819
- Zhang, C., Song, W., Cao, Z., Zhang, J., Tan, P. S., and Chi, X. (2020). Learning to dispatch for job shop scheduling via deep reinforcement learning. *Adv. Neural Inf. Process. Syst.* 33, 1621–1632.
- Zhang, G., Gao, L., and Shi, Y. (2011a). An effective genetic algorithm for the flexible job-shop scheduling problem. *Expert Syst. Appl.* 38, 3563–3573. doi:10.1016/j.eswa.2010.08.145
- Zhang, J., Zhan, Z.-h., Lin, Y., Chen, N., Gong, Y. J., Zhong, J.-h., et al. (2011b). Evolutionary computation meets machine learning: a survey. *IEEE Comput. Intell. Mag.* 6, 68–75. doi:10.1109/mci.2011.942584
- Zhang, Q., and Li, H. (2007). MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* 11, 712–731. doi:10.1109/tevc.2007.892759
- Zhang, R., and Wu, C. (2010). A hybrid approach to large-scale job shop scheduling. *Appl. Intell.* 32, 47–59. doi:10.1007/s10489-008-0134-y
- Zhang, W., Gen, M., and Jo, J. (2014). Hybrid sampling strategy-based multiobjective evolutionary algorithm for process planning and scheduling problem. *J. Intelligent Manuf.* 25, 881–897. doi:10.1007/s10845-013-0814-2
- Zhang, W., Geng, H., Li, C., Gen, M., Zhang, G., and Deng, M. (2023a). Q-learning-based multi-objective particle swarm optimization with local search within factories for energy-efficient distributed flow-shop scheduling problem. *J. Intelligent Manuf.*, 1–24. doi:10.1007/s10845-023-02227-9
- Zhang, W., Hou, W., Li, C., Yang, W., and Gen, M. (2021a). Multidirection update-based multiobjective particle swarm optimization for mixed no-idle flow-shop scheduling problem. *Complex Syst. Model. Simul.* 1, 176–197. doi:10.23919/csms.2021.0017
- Zhang, W., Li, C., Gen, M., Yang, W., and Zhang, G. (2024). A multiobjective memetic algorithm with particle swarm optimization and q-learning-based local search for energy-efficient distributed heterogeneous hybrid flow-shop scheduling problem. *Expert Syst. Appl.* 237, 121570. doi:10.1016/j.eswa.2023.121570
- Zhang, W., Wang, Y., Yang, Y., and Gen, M. (2019). Hybrid multiobjective evolutionary algorithm based on differential evolution for flow shop scheduling problems. *Comput. Industrial Eng.* 130, 661–670. doi:10.1016/j.cie.2019.03.019
- Zhang, Y., Wang, J., Zhang, Z., and Zhou, Y. (2021b). “Modr/d-el: multiobjective deep reinforcement learning with evolutionary learning for multiobjective optimization,” in *2021 International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, China, July 18–22, 2021 (IEEE), 1–8.
- Zhang, Z., Tang, Q., Chica, M., and Li, Z. (2023c). Reinforcement learning-based multiobjective evolutionary algorithm for mixed-model multimanned assembly line balancing under uncertain demand. *IEEE Trans. Cybern.*, 1–14. doi:10.1109/tcyb.2022.3229666
- Zhao, F., Di, S., and Wang, L. (2023a). A hyperheuristic with q-learning for the multiobjective energy-efficient distributed blocking flow shop scheduling problem. *IEEE Trans. Cybern.* 53, 3337–3350. doi:10.1109/tcyb.2022.3192112
- Zhao, F., Jiang, T., and Wang, L. (2023b). A reinforcement learning driven cooperative meta-heuristic algorithm for energy-efficient distributed no-wait flow-shop scheduling with sequence-dependent setup time. *IEEE Trans. Industrial Inf.* 19, 8427–8440. doi:10.1109/tii.2022.3218645
- Zhao, F., Ma, R., and Wang, L. (2022). A self-learning discrete jaya algorithm for multiobjective energy-efficient distributed no-idle flow-shop scheduling problem in heterogeneous factory system. *IEEE Trans. Cybern.* 52, 12675–12686. doi:10.1109/tcyb.2021.3086181
- Zhao, F., Qin, S., Zhang, Y., Ma, W., Zhang, C., and Song, H. (2019). A hybrid biogeography-based optimization with variable neighborhood search mechanism for no-wait flow shop scheduling problem. *Expert Syst. Appl.* 126, 321–339. doi:10.1016/j.eswa.2019.02.023
- Zhao, F., Zhang, L., Cao, J., and Tang, J. (2021b). A cooperative water wave optimization algorithm with reinforcement learning for the distributed assembly no-idle flowshop scheduling problem. *Comput. Industrial Eng.* 153, 107082. doi:10.1016/j.cie.2020.107082
- Zheng, J., Wang, L., and Wang, J.-j. (2020). A cooperative coevolution algorithm for multi-objective fuzzy distributed hybrid flow shop. *Knowledge-Based Syst.* 194, 105536. doi:10.1016/j.knsys.2020.105536
- Zhou, Y., Chen, H., and Zhou, G. (2014). Invasive weed optimization algorithm for optimization no-idle flow shop scheduling problem. *Neurocomputing* 137, 285–292. doi:10.1016/j.neucom.2013.05.063
- Zhou, Y., Kang, J., Kwong, S., Wang, X., and Zhang, Q. (2021). An evolutionary multi-objective optimization framework of discretization-based feature selection for classification. *Swarm Evol. Comput.* 60, 100770. doi:10.1016/j.swevo.2020.100770
- Zitzler, E., Knowles, J., and Thiele, L. (2008). Quality assessment of pareto set approximations. *Multiobjective Optim. Interact. Evol. Approaches*, 373–404.
- Zitzler, E., and Künzli, S. (2004). “Indicator-based selection in multiobjective search,” in *International Conference on Parallel Problem Solving from Nature*, Birmingham, United Kingdom, 18–22 September, 2004 (Springer), 832–842.
- Zitzler, E., Laumanns, M., and Thiele, L. (2001). SPEA2: improving the strength pareto evolutionary algorithm. *TIK-Report* 103. doi:10.3929/ethz-a-004284029