COMPLEX SYSTEM MODELING AND SIMULATION
ISSN 2096-9929 01/06 pp 257−270
Volume 1, Number 4, December 2021
DOI: 10.23919/CSMS.2021.0027

# A Review of Reinforcement Learning Based Intelligent Optimization for Manufacturing Scheduling

Ling Wang*, Zixiao Pan, and Jingjing Wang

**Abstract:** As the critical component of manufacturing systems, production scheduling aims to optimize objectives in terms of profit, efficiency, and energy consumption by reasonably determining the main factors including processing path, machine assignment, execute time and so on. Due to the large scale and strongly coupled constraints nature, as well as the real-time solving requirement in certain scenarios, it faces great challenges in solving the manufacturing scheduling problems. With the development of machine learning, Reinforcement Learning (RL) has made breakthroughs in a variety of decision-making problems. For manufacturing scheduling problems, in this paper we summarize the designs of state and action, tease out RL-based algorithm for scheduling, review the applications of RL for different types of scheduling problems, and then discuss the fusion modes of reinforcement learning and meta-heuristics. Finally, we analyze the existing problems in current research, and point out the future research direction and significant contents to promote the research and applications of RL-based scheduling optimization.

**Key words:** Reinforcement Learning (RL); manufacturing scheduling; scheduling optimization

## 1 Introduction

Production scheduling is a crucial connecting component in the manufacturing system. To improve the production efficiency and effectiveness, scheduling algorithms play an important role, which have always been a significant research topic in interdisciplinary fields, like industrial engineering, automation, management science, and so on. Production scheduling algorithms mainly include three categories, accurate algorithms, heuristics, and meta-heuristics. The exact algorithm can guarantee to obtain the optimal solution in theory, but it is difficult to solve the large-scale problems efficiently and effectively due to the NP-hard nature. Heuristics adopt some rules to construct scheduling solutions efficiently but without global optimization perspective. Moreover, the design of rules highly depends on the deep understanding of the problem specific characteristics. Meta-heuristics can obtain excellent scheduling solutions within acceptable computation time, but the design of search operators is seriously problem dependent. At the same time, for large-sacle problems the iterative search process is very time-consuming and difficult to be applied for real-time scenarios, such as Meituan on-line food delivery.

With the development of artificial intelligence, Reinforcement Learning (RL) has been successfully applied to the sequential decision-making problems, such as games[1] and robots control[2]. During recent years, RL has been successfully applied to solve several combinatorial optimization problems, such as Vehicle Routing Problem[3] (VRP) and Traveling Salesman Problem[4] (TSP). Supposing a production scheduling problem can be regarded as the environment of RL, an agent can learn a policy or rule via reasonable designs of actions and states, as well as interaction with the environment through a large number of offline training. Such a new idea provides a novel approach for solving scheduling problems, especially the uncertain and dynamic problems with

• Ling Wang, Zixiao Pan, and Jingjing Wang are with the Department of Automation, Tsinghua University, Beijing 100084, China. E-mail: wangling@tsinghua.edu.cn; pzx19@mails.tsinghua.edu.cn; wjj18@mails.tsinghua.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2021-10-21; accepted: 2021-11-22

high real-time requirements. By retrieving in Scopus with "reinforcement learning" and "shop scheduling" as the subject terms, it finds 214 articles. Figure 1 shows the statistics of these articles. It can be observed that articles about RL-based shop scheduling optimization have increased rapidly since 2015. Table 1 lists the publications in different directions with more than 3 relevant articles. Clearly, by fusing operation research and artificial intelligence, scheduling optimization based on RL has become an emerging topic in the relate fields.

Since RL has been a hotspot, this paper provides a review of the RL-based research progress for manufacturing scheduling. The remainder of the paper is organized as follows. Section 2 reviews the designs of action and state in RL for scheduling optimization. In Section 3, RL-based algorithms for scheduling are summarized. Section 4 reviews the applications of RL for different types of scheduling problems. Section 5 discusses the integration mode of RL and meta-heuristics. Finally, we analyze the existing problems in current research and point out the future research direction and significant contents to promote the development and applications of RL-based scheduling optimization.

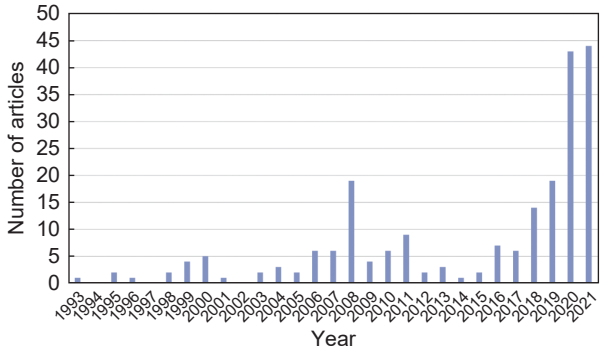


**Fig. 1 Statistics of the RL-based scheduling in Scopus.**

**Table 1 Source of the articles.**

| No. | Publication | Number of articles |
|---|---|---|
| 1 | *Lecture Notes in Computer Science* | 20 |
| 2 | *International J. of Production Research* | 9 |
| 3 | *Computers and Industrial Engineering* | 7 |
| 4 | *IEEE Access* | 7 |
| 5 | *IEEE Trans. on Automation Science and Engineering* | 5 |
| 6 | *Procedia CIRP* | 5 |
| 7 | *European Journal of Operational Research* | 4 |
| 8 | *Winter Simulation Conference* | 4 |
| 9 | *International J. of Advanced Manufacturing Technology* | 4 |
| 10 | *International J. of Simulation Modelling* | 3 |
| 11 | *IEEE Trans. on Industrial Informatics* | 3 |
| 12 | *Computer Integrated Manufacturing Systems* | 3 |
| 13 | *Advances in Intelligent Systems and Computing* | 3 |
| 14 | *Investigacion Operacional* | 3 |
| 15 | *Control and Decision* | 3 |
| 16 | *IEEE International Conference on Robotics and Automation* | 3 |

## 2 State and Action Designs for Scheduling

Different from supervised learning, RL allows an agent to learn optimal behavior without the labelled data through trial-and-error interactions with the environment so as to maximize a numerical reward signal. Figure 2 illustrates the interaction between the agent and the environment in the framework of RL. To be specific, at time $t$, the agent senses the state signal $s_t$ from the environment and performs action $a_t$. Thus, a reward signal $r_{t+1}$ can be obtained and the environment changes to a new state $s_{t+1}$. Then the agent updates the policy according to $r_{t+1}$ and selects the action $a_{t+1}$ under the state $s_{t+1}$ to obtain the reward signal $r_{t+2}$. Through interaction with the environment, the agent learns the decision-making policy in the process of trial-and-error. Finally, the agent can select appropriate actions according to the policy under state $s$ to maximize the cumulative reward.

Through interaction with the environment by extensive offline training, an agent can learn a policy or rule. As crucial component in RL, reasonable designs of action and state can describe the scheduling environment accurately and improve the efficiency of the learning process. In this section, we first review the designs of state and action for scheduling.

### 2.1 Designs of state for scheduling

The designs of state for scheduling problem can be divided into three categories.

(1) Take the production information or the statistics of production information as the state, including processing information, processing environment

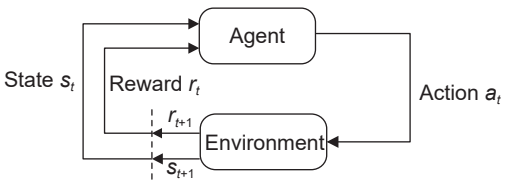


**Fig. 2 Agent-environment interaction in RL.**

information, order information, etc. This method can effectively reduce the loss of information. However, the production information is usually continuous data and the increase of problem scale will bring dimension disaster. Thus, neural network is usually used to approximate the value and policy function.

For the permutation flow shop scheduling, Wang and Pan[5] selected the processing time of jobs on each machine as the state, and proposed an improved pointer network to learn the policy. For the dynamic job-shop scheduling in smart manufacturing, Wang et al.[6] defined three matrices: the processing time matrix of the operations in jobs, the job processing status matrix, and the machine designated matrix as the state, and input the three matrices into neural network to learn the policy. Qu et al.[7] adopted the buffer size, workstation health information, and the workforce condition as the state of the system to guide the selection of the action for the next decision. For the dynamic multi-objective flexible job shop scheduling problem, Luo et al.[8] took 10 kinds of problem information, such as the number of machines, average utilization rate of machines, and the due date tightness as the state, and proposed an effective Deep Q-learning Network (DQN).

(2) Define the state according to the quantitative relationship between the production information or the statistics of the production information. In this way, the challenge of larger state space caused by the increase of problem scale can be avoided, but it will lead to the loss of problem information.

For the dynamic single machine scheduling problem, Wang and Usher[9] defined the states according to the quantity situation for the number of jobs in buffer and the estimation of the total lateness, which effectively reduced the state space. For the integrated scheduling and flexible maintenance in deteriorating multi-state single machine system, Wang et al.[10] divided the state space according to the quantitative relationship between the mean normal processing time and the estimation of the mean lateness of the remaining job. For the job shop scheduling, Zhao et al.[11] defined six states by comparing the estimated average slack time with the estimated average remaining time.

(3) Convert the scheduling problem to a graph, and define the state according to the situation of the nodes and edges in the graph. This method well considers the structural characteristics of the problem and efficiently represents the production environment. Meanwhile, Graph Neural Network (GNN), Convolutional Neural Network (CNN), Graph Convolutional Network (GCN), and other networks are usually adopted to extract the problem characteristics effectively.

Zhang et al.[12] adopted disjunctive graph to model the job shop scheduling problem and proposed a Proximate Policy Optimization (PPO) to optimize the GNN. For the adaptive job shop scheduling problem, Han and Yang[13] represented the production information as multi-channel images, and a CNN was used to approximate the state-action value function. For the dynamic scheduling of flexible manufacturing system, Hu et al.[14] used Petri-net to model the problem and a GCN was adopted to approximate the state-action value function in DQN.

In addition, there are some other ways to design the state. For example, for the permutation flow shop scheduling problem, Zhang and Ye[15] adopted the unprocessed jobs set on the first machine as the state. When there are $n$ unprocessed jobs, the number of states is $2^n$. For the unrelated parallel machine scheduling problem with sequence dependent setup time, Silva et al.[16] proposed a multi-agent optimization framework and designed four neighborhood structures as the state of the algorithm.

In conclusion, it can be seen that there are many ways of state designs. The preeminent state designs need to balance the loss of information and the size of state space. Meanwhile, the characteristics of the scheduling problem and the optimization goal also should be taken into account.

### 2.2 Designs of action for scheduling

The designs of action for scheduling problem can also be divided into three categories.

(1) Select heuristics as action. In this way, heuristics can be used cooperatively, and the number of actions is constant and independent of the size of the problem. However, the performance of the algorithm depends on the efficiency and quality of the selected heuristics.

Lin et al.[17] proposed a smart manufacturing factory framework based on edge computing. Seven heuristics, such as Most-Operations-Remaining (MOR), First-In-First-Out (FIFO), and Longest-Processing-Time (LPT), were selected as the actions in DQN. For the dynamic permutation flow shop scheduling problem, Yang et al.[18] took five rules, such as Shortest-Processing-Time (SPT) and LPT, as the actions of the agents. For the dynamic flexible job shop scheduling with new job insertions, Luo[19] designed six scheduling rules and

used these rules as the actions.

(2) Take the scheduling solution, such as job sequence, as the action. This method is mainly adopted to solve the static scheduling problem by using the end-to-end mode. In this way, the agent can quickly construct a scheduling solution.

For the permutation flow shop scheduling, Wang and Pan[5] designed a new policy network to model the problem. The policy network can directly output the scheduling sequence using the processing information. For the workflow management system, Kintsakis et al.[20] designed a neural network to achieve sequence to sequence generation and directly output the scheduling solutions.

(3) Define the scheduling operators based on the problem characteristics as action. The agent learns to select an appropriate operator, such as deciding the machine assignment, adjusting the job sequence, etc., to generate a new solution. This approach should have a good understanding of the problem to avoid generating infeasible solutions.

For the online single machine scheduling, Li et al.[21] took the length of the jobs in the waiting queue as the state, and defined the selection of an unprocessed job as the action. Q-learning, single step State-Action-Reward-State-Action (SARSA), multi-step Watkins's Q, and multi-step SARSA were adopted to solve the problem, respectively. For the job shop scheduling problems, Williem and Setiawan[22] selected the critical path schedule as initial state. Two operators, pool-reassignment and task-move, were designed as the action. To solve the flow shop scheduling with two-robot job transfer, Arviv et al.[23] proposed an RL algorithm with two Q-learning functions. The transfers of jobs were defined as the action and the cooperative scheduling between robots and production line was realized. For the robust scheduling of semiconductor manufacturing facilities, Park et al.[24] constructed the state by concatenating four local features of a machine and defined the selection of an unprocessed job as the action.

Besides the above three categories, there are other methods for the action design. For example, Silva et al.[16] proposed a multi-agent framework combined with metaheuristics for unrelated parallel machines scheduling. The action was defined as the selection of neighbourhood structures. For the scheduling problem in multi storage edge computing, Wang et al.[25] adopted PPO to solve the problem and the selection of where to execute the task was defined as the action.

It can be seen that there are various methods of action designs. We need to consider the property of the problem in order to generate the appropriate form and number of actions.

## 3 RL-Based Algorithm for Scheduling

According to the usage of the environment model, RL can be divided into two categories, i.e., model-free RL and model-based RL. Model-based RL relies on the environment model which contains state transition and reward prediction. Although agents of the model-based RL can directly obtain the new state and reward, it is difficult to obtain state transition information for production scheduling problems. Unlike model-based RL, model-free RL relies on the real-time interaction between the agent and the environment without state transition information. Currently, most of the existing RL-based production scheduling optimization algrorithms are model-free RL algorithms which can be further divided into value-based RL and policy-based RL.

### 3.1 Value-based RL for scheduling

Value-based RL constructs the optimal strategy by selecting the action with the maximum state-action value. Obviously, the construction and calculation of the value function are the core of the value-based RL. This kind of RL has high sample utilization, but it is easy to over fit with poor generalization. The representative algorithms of value-based RL on production scheduling optimization include SARSA, Q-learning, and DQN.

#### 3.1.1 SARSA for scheduling

SARSA is an on-policy Temporal Difference (TD) algorithm. In the iterative learning process, agent adopts the $\varepsilon$-greedy method to select $a_t$ under the state $s_t$ and obtains the reward $r_{t+1}$. Then, the environment changes to a new state $s_{t+1}$[26]. For the new state, the agent continues to select $a_{t+1}$ by using the $\varepsilon$-greedy method and updates the value function $q(s_t,a_t)$ as follows:

$$q(s_t,a_t)=q(s_t,a_t)+\alpha\times(r_{t+1}+\gamma\times q(s_{t+1},a_{t+1})-q(s_t,a_t)) \tag{1}$$

where $\alpha$ is the learning rate and $\gamma$ is the discount factor.

For scheduling optimization, Palombarini et al.[27] proposed a novel approach to generate rescheduling knowledge based on SARSA and an industrial instance was tested to verify the effectiveness of the algorithm.

Chen et al.[28] designed a self-learning Genetic Algorithm (GA) and introduced SARSA and Q-learning to improve the search capability in different stages. Orhean et al.[29] adopted SARSA to solve the scheduling problem in heterogeneous distributed systems. Experiments show that the performance of SARSA was better than Q-learning on this problem. As for dynamic multi-site scheduling problem, Aissani et al.[30] designed a multi-agent approach by using SARSA. The effectiveness of the proposed method was verified by comparing with GA and mixed integer linear program.

#### 3.1.2 Q-learning for scheduling

Different from SARSA, Q-learning is an off-policy TD algorithm. It updates the value function $q(s_t, a_t)$ as follows:

$$q(s_t, a_t) = q(s_t, a_t) + \alpha \times \left( r_{t+1} + \gamma \times \max_a q(s_{t+1}, a) - q(s_t, a_t) \right) \tag{2}$$

Unlike SARSA, Q-learning selects the maximum value of $q(s_{t+1}, a)$ under the new state $s_{t+1}$ to update $q(s_t, a_t)$[26]. In recent years, Q-learning has made some progress for scheduling optimization. To solve flexible job-shop scheduling, Bouazza et al.[31] applied Q-learning to select rules such as Shortest-Queue (SQ) and Less-Queued-Element (LQE) to realize machine selection. Meanwhile, some rules, such as FIFO and Shortest-Job-First (SJF), were selected to sequence jobs. Moreover, two Q-matrices were adopted to record the state-action value. For the dynamic job shop scheduling, Wang[32] proposed a weighted Q-learning algorithm based on clustering and dynamic search. Four state features were defined to reduce the dimension of the state space. An improved and iteration update strategy was proposed to select the optimal state-action pair. For adaptive order dispatching in the semiconductor industry, Stricker et al.[33] designed an RL-based adaptive control system by using Q-learning. To address the uncertainty of production environment, Wang and Yan[34] proposed an adaptive scheduling mechanism based on Q-learning. To avoid the impact of large state space and minimize the error between the clustering and real states, the state membership was included when updating the weighted Q-value. Experiments showed that this strategy can improve the performance of the algorithm effectively. For the adaptive assembly scheduling of aero-engine, Wang et al.[35] proposed a double-layer Q-learning method. The top level of Q-learning was used to learn machine allocation, and the bottom level of Q-learning was used to learn the scheduling policy for the jobs on each machine. The experimental results showed that the proposed algorithm is able to achieve good and adaptive performances.

#### 3.1.3 Deep Q-learning network for scheduling

Both SARSA and Q-learning adopt a table to record the state-action value, but the table is no longer applicable when the scale of the state space or the action space is too large. Therefore, the deep Q-learning network is proposed by integrating Q-learning and the deep neural network to approximate the value function. DQN uses experience replay and target network to overcome the instability of the algorithm. Liu et al.[36] provided a review about the DQN and its improved methods.

For scheduling optimization, Waschneck et al.[37] presented the Deep RL (DRL) method for production scheduling by using DQN and adopted a case of semiconductor production to validate the proposed algorithm. For the flexible shop floor, Hu et al.[38] proposed an adaptive DRL based AGVs real-time scheduling approach by using DQN. The suitable scheduling rules and AGVs can be selected under different states. The real-world case was used to validate the effectiveness of the algorithm. For the closed-loop rescheduling, Palombarini and Martínez[39] adopted the Gantt chart and prior knowledge as inputs, and proposed a rescheduling algorithm based on DQN. The effectiveness of the algorithm was verified with an industrial example.

Among the three value-based RL algorithms, Q-learning is greedy and easy to be trapped into local optimization. SARSA is relatively conservative, but the exploration rate in the $\varepsilon$-greedy method needs to be controlled to ensure the convergence. DQN is suitable for solving the large-scale problems, but the sampling of DQN is inefficient and strongly dependent on the parameter setting.

### 3.2 Policy-based RL for scheduling

Different from the value-based RL, the policy-based RL does not consider the value function and directly searches for the best policy. Moreover, the policy-based RL usually adopts a neural network to fit the policy function. Such kinds of algorithms have their own exploration mechanisms, but the sample utilization rate is low and it is easy to cause local

optimization with large variance. At present, the policy-based RL has not been widely used for scheduling problems. The typical algorithms include REINFORCE, PPO, and Trust Region Policy Optimization (TRPO).

For permutation flow-shop scheduling problem, Wang and Pan[5] proposed a novel pointer network, and adopted REINFORCE method to train the network. The superior performance of the algorithm was demonstrated by comparing with other algorithms on the benchmark. Rummukainen and Nurminen[40] applied PPO to solve the stochastic economic lot scheduling problem. Zhang et al.[12] used disjunctive graph to describe the job shop scheduling and input it into GNN. PPO is used to train the network. Experiments showed that the algorithm has good performance. Considering the characteristics of smaller batch size, larger product variety, and complex processes in production system, Kuhnle et al.[41] designed an autonomous dispatching algorithm based on TRPO. The effectiveness of the proposed scheduling algorithm was verified by using the real-world case in the semiconductor industry.

Among the three policy-based RL methods, REINFORCE belongs to policy gradient based Monte Carlo algorithm with good stability but low sample utilization. The performances of the PPO and TRPO are not strongly dependent on super parameters, but their sampling rates are low with strong running environment support.

In addition to the value-based RL and policy-based RL, there are other types of RL methods, such as actor-critic method. To solve the job shop scheduling problem, Liu et al.[42] proposed a parallel training method to train the model using asynchronous update and DDPG. Hubbs et al.[43] applied advantage actor-critic to solve chemical production scheduling. Experiments showed that the speed and flexibility of RL are helpful to realize the real-time optimization of scheduling systems. For online job scheduling, Chen and Tian[44] proposed NeuRewriter to learn the policy and used actor-critic method to train the neural network. The experimental results showed the effectiveness of the proposed algorithm.

Figure 3 shows the statistics of the RL-based algorithm for scheduling. Although there are many works on the value-based RL for solving production scheduling, the study of policy-based RL for scheduling remains a large space.

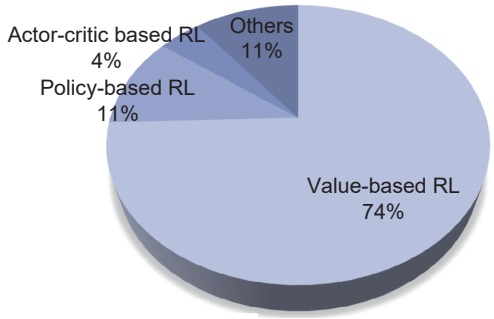


**Fig. 3 Statistics of the RL-based algorithm for scheduling.**

## 4 RL Applications for Scheduling

Regarding the applications of RL for different types of scheduling problems, it shows the number of papers in Fig. 4. It can be seen that RL is mostly used to solve job shop scheduling problems. The applications for flow shop, parallel machine, and single machine scheduling problems need further study. Next, we summarize the typical works about RL for different types of scheduling problems.

### 4.1 RL for single machine scheduling

The constraint of single machine scheduling is relatively simple, and it only needs to decide the job process sequence of the jobs. Currently, RL is mostly used to solve single machine scheduling problems under stochastic, dynamic, or online conditions.

Wang et al.[45] designed and compared two RL-based methods to address a stochastic economic lot scheduling problem for a single machine make-to-stock production system. Xie et al.[46] adopted Q-learning to solve the online single machine scheduling problem. The states of the machine and queue were selected as the state of the environment. Wang et al.[47] divided 23 states and two virtual states according to the situation of the buffer. Three scheduling rules, Earliest-Due-

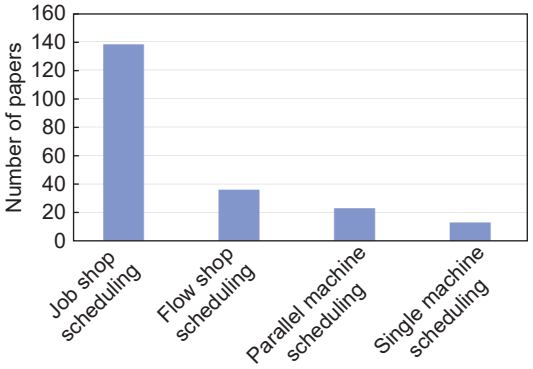


**Fig. 4 Number of papers of RL in different types of scheduling.**

Date (EDD), SPT, and FIFO, were selected to form the action set to optimize three objectives, including the maximum delay time, the number of delayed jobs, and the average flow time. For multi-state single machine production scheduling with degradation processes, Yang et al.[48] proposed a novel heuristic RL method to deal with the problem more efficiently. For production scheduling and preventive maintenance in multi-state production systems, Yang et al.[49] transformed the problem into Markov decision process, and designed a model-free RL algorithm to solve the problem. Considering that the arrival time of jobs obeys Poisson distribution, Wang and Usher[50] proposed a Q-learning to dynamically select three scheduling rules to minimize the average delay time for single machine scheduling problem.

### 4.2 RL for parallel machine scheduling

Compared with the single machine scheduling, the parallel machine scheduling needs to consider machine assignment. The design of the state and action of the agent is complex. RL-based parallel machine scheduling optimization algorithms are mainly designed for the dynamic scheduling problem.

For the dynamic parallel machine scheduling problem with sequence-dependant setup times and machine-job qualification consideration, Zhang et al.[51] adopted Q-learning to minimize the mean weighted tardiness, and selected five heuristics as actions. For the dynamic scheduling problem in smart manufacturing, Zhou et al.[52] proposed a deep RL based method to minimize the maximum completion time. The target network and prediction network are used to cooperate in the training process to improve the stability. For the parallel machine scheduling with dynamic arrival of job, Zhang et al.[53] converted the problem into a Semi-Markov Decision Process (SMDP). Two heuristics were selected as actions, and R-learning was adopted to the problem. Considering different types of jobs arriving dynamically in independent Poisson processes, Zhang et al.[54] applied an on-line R-learning with function approximation to solve the unrelated parallel machine scheduling. The performance of the algorithm was better than four heuristics.

### 4.3 RL for flow shop scheduling

Flow shop scheduling needs to consider the processing of multiple stages. In order to realize flexible manufacturing, there are several parallel machines at some stages, i.e, hybrid or flexible flow shop scheduling. Obviously, it is more complex than parallel machine scheduling.

For the permutation flow shop scheduling, Zhang and Ye[15] transformed the problem into sequential decision problem, and proposed a Q-learning based scheduling algorithm. The effectiveness of the algorithm was verified by using the benchmark instances. For the non-permutation flow shop scheduling problem, Xiao et al.[55] proposed a deep temporal difference RL network. Several scheduling rules were selected according to the environment state. Zhang et al.[56] converted the non-permutation flow shop scheduling into an SMDP by constructing state features, actions, and reward function. Moreover, an on-line TD ($\lambda$) algorithm was applied to solve the problem.

For the hybrid flow shop scheduling problem, Han et al.[57] designed an effective Q-learning algorithm. Boltzmann exploration policy was adopted to trade-off the exploration and exploitation. Experiments demonstrated the effectiveness of the method. For the flow shop scheduling with sequence dependent setup time, Fonseca-Reyna and Martínez-Jiménez[58] presented an improved Q-learning to minimize the completion time of all jobs. For the distributed assembly no-idle flow shop scheduling problem, Zhao et al.[59] proposed a cooperative water wave algorithm with RL, and adopted Q-learning to balance the exploration and exploitation capabilities of the algorithm.

### 4.4 RL for job shop scheduling

Compared with the above three kinds of scheduling problems, job shop scheduling needs to consider different machine processing routes for jobs. For flexible job shop scheduling, the machine assignment should also be considered. Therefore, the design of the scheduling algorithm is more complex.

In the static scenario, Gabel and Riedmilier[60] transformed the classical job shop scheduling into a sequential decision problem, and introduced the neural network to approximate the value function. The simulation results showed that the performance of the designed algorithm was better than the existing rules. Combining the learning and optimization, Martínez et al.[61] proposed a two-stage method to solve the flexible job shop scheduling problem. In the first stage, Q-learning was used to realize machine assignment and job scheduling and the feasible solution can be

generated. In the second stage, several strategies were designed to optimize and improve the obtained feasible solution. The effectiveness of the algorithm was verified by comparing with the existing approaches.

In the dynamic scenario, Kardos et al.[62] designed a Q-learning based scheduling algorithm to solve the dynamic job shop scheduling problem effectively. For the job shop scheduling problem with random job arrivals, Luo et al.[63] proposed a double loop deep Q-network with exploration loop and exploitation loop. For each job, two ratios related to processing time were selected as the state variables. Zhao et al.[11] proposed an improved Q-learning algorithm to solve job shop scheduling problem. The concept of the urgency of remaining tasks was adopted to describe the state space, and several scheduling rules, such as FIFO and SPT, were selected as actions to minimize the total tardiness. For the flexible job shop scheduling problem with new job insertions, Luo[19] proposed six dispatching rules and developed a deep Q-network to minimize the total tardiness. For the dynamic job shop scheduling problem with random job arrivals and machine breakdowns, Shahrabi et al.[64] proposed an RL-based variable neighborhood search to minimize the mean flow time. Several states were defined by using the number of jobs and the average processing time of current jobs. Q-learning was used to learn the parameter selection in different states. Considering machine breakdown, new machine arrival, job cancellation, and new job arrival, Csáji et al.[65] proposed a triple-level learning mechanism to achieve adaptive behavior and search space reduction. The top level is composed of simulated annealing algorithm, and the middle level contains an RL system, and the lower level is a numerical function approximator. For the assembly job shop scheduling problem with uncertain assembly, Wang et al.[66] designed a dual Q-learning method to minimize the total weighted earliness penalty and completion time cost. The top level Q-learning was used to find the dispatching strategy and the bottom level Q-learning was used for global optimization. The experiments showed that dual Q-learning can yield better results than single Q-learning.

### 4.5 RL for other scheduling problems

RL has also been applied for some other types of scheduling problems, such as distributed scheduling, energy efficiency scheduling, and multi-objective scheduling. Moreover, RL has made progress in several real production scenarios, such as edge computing task scheduling and agricultural irrigation scheduling.

For the multi-site companies scheduling problem, Aissani et al.[30] proposed a multi-agent method based on RL. Each company was composed of an observer agent, many inventory agents, and resource agents. Compared with GA and mixed integer linear programming, the effectiveness of the proposed algorithm was verified. To deal with the high-dimensional data in the distributed system, Zhou et al.[67] presented a new cyber-physical integration method in smart factories for online scheduling. RL was introduced to improve the decision-making ability of the scheduling algorithm.

For the problem of energy efficiency scheduling in virtual machines, Wang et al.[68] proposed a deep RL model based on Quality of Service (QoS) feature learning. Extensive experiments showed that the proposed method can effectively reduce the energy consumption. For reducing energy consumption of machining job shops, He et al.[69] proposed an improved Q-learning algorithm to optimize total energy consumption of task, makespan, and workload of machine simultaneously.

For the dynamic multi-objective job shop scheduling problem with just-in-time constraint, Hong and Prabhu[70] modelled the problem as SMDP and introduced a novel scheduling algorithm by using Q-learning. The performance of the algorithm was significantly better than other scheduling rules. For the multi-objective scheduling in semiconductor industry, Kuhnle et al.[41] used a weighted method to deal with the two objectives optimization problem. By introducing RL, the scheduling solution can be generated automatically. For the multi-objective scheduling problem with uncertain, Zhou et al.[71] presented a new RL-based scheduling method with composite reward functions to optimize makespan, production cost, balance workloads, and other indicators. For the multi-objective scheduling problem in heterogeneous cloud environment, Yuan et al.[72] designed a multi-objective reinforcement learning based on analytic hierarchy process to optimize execution time, energy consumption, and execution cost.

For the offloading scheduling in vehicle edge computing, Zhan et al.[73] transformed the problem into a Markov decision process and introduced CNN to approximate both policy and value function. Moreover,

PPO was adopted to yield better performance than many heuristic algorithms. For the irrigation scheduling, Yang et al.[74] introduced a deep RL algorithm to realize highly accurate water scheduling. For the supply chain ordering management system, Mortazavi et al.[75] used RL to design the scheduling algorithm and achieved good performance.

Figure 5 illustrated the statistics of the RL for static scheduling and dynamic scheduling. It can be found that RL-based production scheduling optimization are mainly adopted to solve dynamic scheduling problems. The reason is that after learning the properties and knowledge of the problems from the interaction with environment, the RL agent can continue to solve the following dynamic scheduling problem under the same problem scenario to obtain the solution quickly. However, for the static scheduling problem, when the trained RL agents are extended to solve the instances in different scenarios, the solution obtained by RL are often not as good as those obtained by the meta-heuristics. Thus, the research on RL with end-to-end model for static scheduling needs further research. Moreover, RL is mainly applied to scheduling problems in simple scenarios even with single objective. Thus, the study of RL for solving complex scheduling problems and multi-objective optimization should be stressed.

## 5 Integration of RL and Meta-Heuristic for Scheduling

The RL applications for scheduling are very promising and still need to be discussed and studied. In recent decades, as an important branch of artificial intelligence, computational intelligence, especially meta-heuristic, has made great advances in the production scheduling. However, meta-heuristics with single search mode are difficult to deal with complex scheduling problems effectively and efficiently, such as distributed scheduling and green scheduling. It is necessary to introduce a variety of mechanisms, such as learning mechanism, to assist meta-heuristics to improve search efficiency. Thus, the integration of RL and meta-heuristic is a promising way to improve the algorithm performance. In this section, we discuss the integration mode of RL and meta-heuristics.

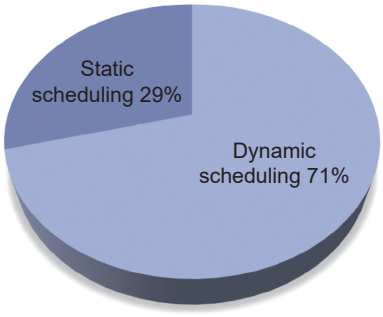


**Fig. 5 Statistics of the RL for static scheduling and dynamic scheduling.**

(1) RL and meta-heuristics are regarded as two stages of the algorithm. This is a simple and easy way to combine the advantages of RL and meta-heuristics to improve the solution quality. For the flow shop scheduling problem, Wang and Pan[5] proposed a new network to model the problem and it was trained by RL. After the network output a solution, an iterative greedy algorithm was adopted to improve the result.

(2) RL is used to guide the parameter selection of meta-heuristics. Through interaction with the environment, RL can learn the knowledge of parameter setting. The meta-heuristics can realize the adaptive adjustment by using the guidance of the trained agent.

For the dynamic job shop scheduling problem, Shahrabi et al.[64] adopted Q-learning to learn the selection of core parameters of VNS. The core parameters of VNS can dynamically adjust in the iterative search process to improve the performance. Xing and Liu[76] designed an adaptive particle swarm optimization algorithm based on RL. A Q-learning algorithm was introduced to change the inertia weight dynamically, which effectively improves the efficiency of the algorithm.

(3) RL is adopted to guide the search of meta-heuristics. In this way, the advantages of RL and meta-heuristics can be used to realize the adaptive selection of the search strategies and the adaptive adjustment of the search directions. It is an effective way to cope with the complex scheduling problems, such as green scheduling and distributed scheduling.

For the energy-aware distributed hybrid flow shop scheduling, the authors in Ref. [77] proposed a collaborative memetic algorithm and designed an RL-based method to assist in the selection of search operators. Li et al.[78] proposed an improved genetic algorithm combined with RL. The gene space of genetic algorithm is regarded as the action strategy space and the Q-learning algorithm can assist in the search of genetic algorithm. For the additive manufacturing machine scheduling, Alicastro et al.[79] proposed an iterative local search algorithm based on

RL. In the search process, RL can assist in the selection of neighbourhood structures. Zhao et al.[59] proposed a cooperative water wave algorithm to solve the flow shop scheduling problem, and introduced Q-learning to balance the exploration and exploitation of the algorithm.

To sum up, existing researches show that the integration of RL and meta-heuristic can effectively improve the performance of the algorithm. However, related works are still not plentiful enough, which need to be further discussed. Therefore, it is necessary to explore new modes for the integration of RL and meta-heuristics.

## 6 Discussion and Conclusion

Production scheduling is the core of the manufacturing system and has attracted much concern. In view of the large scale and the real-time requirements, the existing scheduling algorithms are facing huge challenges. With the development of artificial intelligence, RL has made breakthroughs in many combinatorial optimization problems and provides a new way for scheduling optimization. In this paper, the RL for the production scheduling is reviewed to provide a guideline for RL intelligent optimization for production scheduling.

From the existing researches about RL-based scheduling, RL algorithms have particular advantages, such as convenience and rapidity in solving shop scheduling problems, especially dynamic scheduling problems. However, relevant research is still in its infancy and remains to be further explored in problem, algorithm, and application domains.

**(1) Problem domain**

The existing works mainly focus on the RL to solve the single objective scheduling problem. Meanwhile, the rare studies about multi-objective optimization mainly consider the economic and time index. On the one hand, according to different requirements, it is necessary to study the machine load balance, the number of delayed jobs, and other scheduling indicators; on the other hand, the proposal of carbon peak and carbon neutrality targets promotes the green transformation of industry and accelerates the integration of intelligent manufacturing and green manufacturing. Thus, it is of practical significance to explore the RL algorithm to optimize economic and green objectives simultaneously.

In addition, most literatures about the RL for the production scheduling problems are simplified and traditional. At the same time, many real-life constraints should be considered, such as no-idle, no-wait, sequence-dependent setup time, and machine deterioration effect. It is of great practical value to study RL algorithm in solving production scheduling problems with complex process constraints.

**(2) Algorithm domain**

Currently, the existing RL algorithms lack the theoretical analysis and support for solving scheduling problems. Besides, the absence of systematic methods to guide the designs of state and action is also unfavorable for the promotion and application of RL in solving the production scheduling problems. Therefore, the research on the theory and method of RL algorithms for production scheduling optimization has very important academic value.

At present, policy-based RL algorithms seldom used for production scheduling problems can search optimal policy and generate the schedule in an end-to-end way, which can cope with the challenges of real-time scenarios effectively. Therefore, it is important to solve the production scheduling problem in an end-to-end way and realize the adaptive generation of scheduling rules via research of policy-based RL algorithms, such as PPO and TRPO.

Considering the synergy with meta-heuristics, the studies about the cooperative RL are relatively rare. It is a promising research direction to explore effective fusion mechanisms of the RL and meta-heuristics. It is expected to find associated knowledge and improve search efficiency via giving full play to the advantages of RL to decide the search direction and search step length, and adaptively adjust search operations and parameters setting.

**(3) Application domain**

At present, most researches on RL for the scheduling problems stay at the academic level. Relevant theories and methods are only tested and analyzed through simulation, lacking the application of practical problems. Therefore, it is necessary to strengthen the understanding and refinement of practical problems, emphasize problem modeling, and algorithm design, and promote the application of RL algorithm for solving shop scheduling.

In short, the research of production scheduling optimization based on RL is promising, while many areas need to be improved and explored. With the development of RL technology, it is believed that the theory, method, and application research can be

comprehensively developed and enhanced.

## Acknowledgment

This work was supported in part by the National Science Fund for Distinguished Young Scholars of China (No. 61525304) and the National Natural Science Foundation of China (No. 61873328).

### References

[1] N. Dilokthanakul, C. Kaplanis, N. Pawlowski, and M. Shanahan, Feature control as intrinsic motivation for hierarchical reinforcement learning, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3409–3418, 2019.

[2] Y. Y. Jia and S. G. Ma, A coach-based bayesian reinforcement learning method for snake robot control, *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2319–2326, 2021.

[3] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, Neural combinatorial optimization with reinforcement learning, in *Proc. 5th Int. Conf. Learning Representations*, Toulon, France, 2017, pp. 1–13.

[4] W. Kool, H. Van Hoof, and M. Welling, Attention, learn to solve routing problems!, in *Proc. 7th Int. Conf. Learning Representations*, New Orleans, LA, USA, 2019, pp. 1–12.

[5] L. Wang and Z. X. Pan, Scheduling optimization for flow-shop based on deep reinforcement learning and iterative greedy method, (in Chinese), *Control and Decision*, vol. 36, no. 11, pp. 2609–2617, 2021.

[6] L. B. Wang, X. Hu, Y. Wang, S. J. Xu, S. J. Ma, K. X. Yang, Z. J. Liu, and W. D. Wang, Dynamic job-shop scheduling in smart manufacturing using deep reinforcement learning, *Comput. Netw.*, vol. 190, p. 107969, 2021.

[7] S. H. Qu, J. Wang, S. Govil, and J. O. Leckie, Optimized adaptive scheduling of a manufacturing process system with multi-skill workforce and multiple machine types: An ontology-based, multi-agent reinforcement learning approach, *Procedia Cirp*, vol. 57, pp. 55–60, 2016.

[8] S. Luo, L. X. Zhang, and Y. S. Fan, Dynamic multi-objective scheduling for flexible job shop by deep reinforcement learning, *Comput. Ind. Eng.*, vol. 159, p. 107489, 2021.

[9] Y. C. Wang and J. M. Usher, Application of reinforcement learning for agent-based production scheduling, *Eng. Appl. Artif. Intell.*, vol. 18, no. 1, pp. 73–82, 2005.

[10] H. F. Wang, Q. Yan, and S. Z. Zhang, Integrated scheduling and flexible maintenance in deteriorating multi-state single machine system using a reinforcement learning approach, *Adv. Eng. Inform.*, vol. 49, p. 101339, 2021.

[11] Y. J. Zhao, Y. H. Wang, J. Zhang, and H. X. Yu, Application of improved Q learning algorithm in job shop scheduling problem, (in Chinese), *Journal of System Simulation*, https://kns.cnki.net/kcms/detail/11.3092.V.20210423.1823.002.html, 2021.

[12] C. Zhang, W. Song, Z. G. Cao, J. Zhang, P. S. Tan, and C. Xu, Learning to dispatch for job shop scheduling via deep reinforcement learning, arXiv preprint arXiv: 2010.12367, 2020.

[13] B. A. Han and J. J. Yang, Research on adaptive job shop scheduling problems based on dueling double DQN, *IEEE Access*, vol. 8, pp. 186474–186495, 2020.

[14] L. Hu, Z. Y. Liu, W. F. Hu, Y. Y. Wang, J. R. Tan, and F. Wu, Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network, *J. Manuf. Syst.*, vol. 55, pp. 1–14, 2020.

[15] D. Y. Zhang and C. M. Ye, Reinforcement learning algorithm for permutation flow shop scheduling to minimize makespan, (in Chinese), *Comput. Syst. Appl.*, vol. 28, no. 12, pp. 195–199, 2019.

[16] M. A. L. Silva, S. R. de Souza, M. J. F. Souza, and A. L. C. Bazzan, A reinforcement learning-based multi-agent framework applied for solving routing and scheduling problems, *Expert Syst. Appl.*, vol. 131, pp. 148–171, 2019.

[17] C. C. Lin, D. J. Deng, Y. L. Chih, and H. T. Chiu, Smart manufacturing scheduling with edge computing using multiclass deep Q network, *IEEE Trans. Ind. Inform.*, vol. 15, no. 7, pp. 4276–4284, 2019.

[18] S. L. Yang, Z. G. Xu, and J. Y. Wang, Intelligent decision-making of scheduling for dynamic permutation flowshop via deep reinforcement learning, *Sensors*, vol. 21, no. 3, p. 1019, 2021.

[19] S. Luo, Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning, *Appl. Soft Comput.*, vol. 91, p. 106208, 2020.

[20] A. M. Kintsakis, F. E. Psomopoulos, and P. A. Mitkas, Reinforcement learning based scheduling in a workflow management system, *Eng. Appl. Artif. Intell.*, vol. 81, pp. 94–106, 2019.

[21] Y. Y. Li, E. Fadda, D. Manerba, R. Tadei, and O. Terzo, Reinforcement learning algorithms for online single-machine scheduling, in *Proc. 2020 Federated Conf. Computer Science and Information Systems*, Sofia, Bulgaria, 2020, pp. 277–283.

[22] R. S. Williem and K. Setiawan, Reinforcement learning combined with radial basis function neural network to solve Job-Shop scheduling problem, in *Proc. 2011 IEEE Int. Summer Conference of Asia Pacific Business Innovation and Technology Management*, Dalian, China, 2011, pp. 29–32.

[23] K. Arviv, H. Stern, and Y. Edan, Collaborative reinforcement learning for a two-robot job transfer flow-shop scheduling problem, *Int. J. Prod. Res.*, vol. 54, no. 4, pp. 1196–1209, 2016.

[24] I. B. Park, J. Huh, J. Kim, and J. Park, A reinforcement learning approach to robust scheduling of semiconductor manufacturing facilities, *IEEE Trans. Automat. Sci. Eng.*, vol. 17, no. 3, pp. 1420–1431, 2020.

[25] J. Wang, J. Hu, G. Y. Min, W. H. Zhan, Q. Ni, and N. Georgalas, Computation offloading in multi-access edge computing using a deep sequential model based on

reinforcement learning, *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 64–69, 2019.

[26] Z. H. Qin, N. Li, X. T. Liu, X. L. Liu, Q. Tong, and X. H. Liu, Overview of research on model-free reinforcement learning, (in Chinese), *Computer Science*, vol. 48, no. 3, pp. 180–187, 2021.

[27] J. Palombarini, J. C. Barsce, and E. Martinez, Generating rescheduling knowledge using reinforcement learning in a cognitive architecture, arXiv preprint arXiv: 1805.04752, 2018.

[28] R. H. Chen, B. Yang, S. Li, and S. L. Wang, A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem, *Comput. Ind. Eng.*, vol. 149, p. 106778, 2020.

[29] A. I. Orhean, F. Pop, and I. Raicu, New scheduling approach using reinforcement learning for heterogeneous distributed systems, *J. Parallel Distrib. Comput.*, vol. 117, pp. 292–302, 2018.

[30] N. Aissani, A. Bekrar, D. Trentesaux, and B. Beldjilali, Dynamic scheduling for multi-site companies: A decisional approach based on reinforcement multi-agent learning, *J. Intell. Manuf.*, vol. 23, no. 6, pp. 2513–2529, 2012.

[31] W. Bouazza, Y. Sallez, and B. Beldjilali, A distributed approach solving partially flexible job-shop scheduling problem with a Q-learning effect, *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15890–15895, 2017.

[32] Y. F. Wang, Adaptive job shop scheduling strategy based on weighted Q-learning algorithm, *J. Intell. Manuf.*, vol. 31, no. 2, pp. 417–432, 2020.

[33] N. Stricker, A. Kuhnle, R. Sturm, and S. Friess, Reinforcement learning for adaptive order dispatching in the semiconductor industry, *CIRP Annals*, vol. 67, no. 1, pp. 511–514, 2018.

[34] H. X. Wang and H. S. Yan, An interoperable adaptive scheduling strategy for knowledgeable manufacturing based on SMGWQ-learning, *J. Intell. Manuf.*, vol. 27, no. 5, pp. 1085–1095, 2016.

[35] H. X. Wang, H. S. Yan, and Z. Wang, Adaptive assembly scheduling of aero-engine based on double-layer *Q*-learning in knowledge manufacturing, (in Chinese), *Computer Integrated Manufacturing Systems*, vol. 20, no. 12, pp. 3000–3010, 2014.

[36] J. W. Liu, F. Gao, and X. L. Luo, Survey of deep reinforcement learning based on value function and policy gradient, (in Chinese), *Chinese Journal of Computers*, vol. 42, no. 6, pp. 1406–1438, 2019.

[37] B. Waschneck, A. Reichstaller, L. Belzner, T. Altenmüller, T. Bauernhansl, A. Knapp, and A. Kyek, Optimization of global production scheduling with deep reinforcement learning, *Procedia CIRP*, vol. 72, pp. 1264–1269, 2018.

[38] H. Hu, X. L. Jia, Q. X. He, S. F. Fu, and K. Liu, Deep reinforcement learning based AGVs real-time scheduling with mixed rule for flexible shop floor in industry 4.0, *Comput. Ind. Eng.*, vol. 149, p. 106749, 2020.

[39] J. A. Palombarini and E. C. Martínez, Closed-loop rescheduling using deep reinforcement learning, *IFAC-PapersOnLine*, vol. 52, no. 1, pp. 231–236, 2019.

[40] H. Rummukainen and J. K. Nurminen, Practical reinforcement learning-experiences in lot scheduling application, *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 1415–1420, 2019.

[41] A. Kuhnle, N. Röhrig, and G. Lanza, Autonomous order dispatching in the semiconductor industry using reinforcement learning, *Procedia CIRP*, vol. 79, pp. 391–396, 2019.

[42] C. L. Liu, C. C. Chang, and C. J. Tseng, Actor-critic deep reinforcement learning for solving job shop scheduling problems, *IEEE Access*, vol. 8, pp. 71752–71762, 2020.

[43] C. D. Hubbs, C. Li, N. V. Sahinidis, I. E. Grossmann, and J. M. Wassick, A deep reinforcement learning approach for chemical production scheduling, *Comput. Chem. Eng.*, vol. 141, p. 106982, 2020.

[44] X. Y. Chen and Y. D. Tian, Learning to perform local rewriting for combinatorial optimization, arXiv preprint arXiv: 1810.00337, 2019.

[45] J. Wang, X. P. Li, and X. Y. Zhu, Intelligent dynamic control of stochastic economic lot scheduling by agent-based reinforcement learning, *Int. J. Prod. Res.*, vol. 50, no. 16, pp. 4381–4395, 2012.

[46] S. F. Xie, T. Zhang, and O. Rose, Online single machine scheduling based on simulation and reinforcement learning, in *Proc. of the Simulation in Produktion und Logistik*, *Wissenschaftliche Scripten*, Auerbach, Germany, 2019, pp. 59–68.

[47] S. J. Wang, S. Sun, B. H. Zhou, and L. F. Xi, Q-learning based dynamic single machine scheduling, (in Chinese), *Journal of Shanghai Jiaotong University*, vol. 41, no. 8, pp. 1227–1232 & 1243, 2007.

[48] H. B. Yang, W. C. Li, and B. Wang, Joint optimization of preventive maintenance and production scheduling for multi-state production systems based on reinforcement learning, *Reliab. Eng. Syst. Saf.*, vol. 214, p. 107713, 2021.

[49] H. B. Yang, L. Shen, M. Cheng, and L. F. Tao, Integrated optimization of scheduling and maintenance in multi-state production systems with deterioration effects, (in Chinese), *Computer Integrated Manufacturing Systems*, vol. 24, no. 1, pp. 80–88, 2018.

[50] Y. C. Wang and J. M. Usher, Learning policies for single machine job dispatching, *Robot. Comput. -Integr. Manuf.*, vol. 20, no. 6, pp. 553–562, 2004.

[51] Z. C. Zhang, L. Zheng, and M. X. Weng, Dynamic parallel machine scheduling with mean weighted tardiness objective by Q-Learning, *Int. J. Adv. Manuf. Technol.*, vol. 34, no. 9, pp. 968–980, 2007.

[52] L. F. Zhou, L. Zhang, and B. K. P. Horn, Deep reinforcement learning-based dynamic scheduling in smart manufacturing, *Procedia CIRP*, vol. 93, pp. 383–388, 2020.

[53] Z. C. Zhang, L. Zheng, and X. H. Weng, Parallel machines scheduling with reinforcement learning, (in Chinese), *Computer Integrated Manufacturing Systems*, vol. 13, no. 1, pp. 110–116, 2007.

[54] Z. C. Zhang, L. Zheng, N. Li, W. P. Wang, S. Y. Zhong, and K. S. Hu, Minimizing mean weighted tardiness in unrelated parallel machine scheduling with reinforcement learning, *Comput. Operat. Res.*, vol. 39, no. 7, pp. 1315–1324, 2012.

[55] P. F. Xiao, C. Y. Zhang, L. L. Meng, H. Hong, and W. Dai, Non-permutation flow shop scheduling problem based on deep reinforcement learning, (in Chinese), *Computer Integrated Manufacturing Systems*, vol. 27, no. 1, pp. 192–205, 2021.

[56] Z. C. Zhang, W. P. Wang, S. Y. Zhong, and K. S. Hu, Flow shop scheduling with reinforcement learning, *Asia-Pac. J. Operat. Res.*, vol. 30, no. 5, p. 1350014, 2013.

[57] W. Han, F. Guo, and X. C. Su, A reinforcement learning method for a hybrid flow-Shop scheduling problem, *Algorithms*, vol. 12, no. 11, p. 222, 2019.

[58] Y. C. Fonseca-Reyna and Y. Martínez-Jiménez, Adapting a reinforcement learning approach for the flow shop environment with sequence-dependent setup time, *Revista Cubana de Ciencias Informáticas*, vol. 11, no. 1, pp. 41–57, 2017.

[59] F. Q. Zhao, L. X. Zhang, J. Cao, and J. X. Tang, A cooperative water wave optimization algorithm with reinforcement learning for the distributed assembly no-idle flowshop scheduling problem, *Comput. Ind. Eng.*, vol. 153, p. 107082, 2021.

[60] T. Gabel and M. Riedmiller, Scaling adaptive agent-based reactive job-shop scheduling to large-scale problems, in *Proc*. of *2007 IEEE Symp. Computational Intelligence in Scheduling*, Honolulu, HI, USA, 2007, pp. 259–266.

[61] Y. Martínez, A. Nowé, J. Suárez, and R. Bello, A reinforcement learning approach for the flexible job shop scheduling problem, in *Proc*. of *the 5th Int. Conf. Learning and Intelligent Optimization*, Rome, Italy, 2011, pp. 253–262.

[62] C. Kardos, C. Laflamme, V. Gallina, and W. Sihn, Dynamic scheduling in a job-shop production system with reinforcement learning, *Procedia CIRP*, vol. 97, pp. 104–109, 2021.

[63] B. Luo, S. B. Wang, B. Yang, and L. L. Yi, An improved deep reinforcement learning approach for the dynamic job shop scheduling problem with random job arrivals, *J. Phys.*: *Conf. Ser.*, vol. 1848, no. 1, p. 012029, 2021.

[64] J. Shahrabi, M. A. Adibi, and M Mahootchi, A reinforcement learning approach to parameter estimation in dynamic job shop scheduling, *Comput. Ind. Eng.*, vol. 110, pp. 75–82, 2017.

[65] B. C. Csáji, L. Monostori, and B. Kádár, Reinforcement learning in a distributed market-based production control system, *Adv. Eng. Inform.*, vol. 20, no. 3, pp. 279–288, 2006.

[66] H. X. Wang, B. R. Sarker, J. Li, and J. Li, Adaptive scheduling for assembly job shop with uncertain assembly times based on dual Q-learning, *Int. J. Prod. Res.*, vol. 59, no. 19, pp. 5867–5883, 2021.

[67] T. Zhou, D. B. Tang, H. H. Zhu, and Z. Q. Zhang, Multi-agent reinforcement learning for online scheduling in smart factories, *Robot. Comput. -Integr. Manuf.*, vol. 72, p. 102202, 2021.

[68] B. Wang, F. G. Liu, and W. W. Lin, Energy-efficient VM scheduling based on deep reinforcement learning, *Future Generation Computer Systems*, vol. 125, pp. 616–628, 2021.

[69] Y. He, L. X. Wang, Y. F. Li, and Y. L. Wang, A scheduling method for reducing energy consumption of machining job shops considering the flexible process plan, (in Chinese), *Journal of Mechanical Engineering*, vol. 52, no. 19, pp. 168–179, 2016.

[70] J. Hong and V. V. Parbhu, Distributed reinforcement learning control for batch sequencing and sizing in Just-In-Time manufacturing systems, *Appl. Intell.*, vol. 20, no. 1, pp. 71–87, 2004.

[71] T. Zhou, D. B. Tang, H. H. Zhu, and L. P. Wang, Reinforcement learning with composite rewards for production scheduling in a smart factory, *IEEE Access*, vol. 9, pp. 752–766, 2020.

[72] J. L. Yuan, M. C. Chen, T. Jiang, and C. Li, Multi-objective reinforcement learning job scheduling method using AHP fixed weight in heterogeneous cloud environment, (in Chinese), *Control and Decision*, doi: 10.13195/j.kzyjc.2020.0911.

[73] W. H. Zhan, C. B. Luo, J. Wang, C. Wang, G. Y. Min, H. C. Duan, and Q. X. Zhu, Deep reinforcement learning-based offloading scheduling for vehicular edge computing, *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5449–5465, 2020.

[74] Y. X. Yang, J. Hu, D. Porter, T. Marek, K. Heflin, and H. X. Kong, Deep reinforcement learning-based irrigation scheduling, *Trans. ASABE*, vol. 63, no. 3, pp. 549–556, 2020.

[75] A. Mortazavi, A. A. Khamseh, and P. Azimi, Designing of an intelligent self-adaptive model for supply chain ordering management system, *Eng. Appl. Artif. Intell.*, vol. 37, pp. 207–220, 2015.

[76] C. M. Xing and F. A. Liu, An adaptive particle swarm optimization based on reinforcement learning, (in Chinese), *Control and Decision*, vol. 26, no. 1, pp. 54–58, 2011.

[77] J. J. Wang and L. Wang, A cooperative memetic algorithm with learning-based agent for energy-aware distributed hybrid flow-Shop scheduling, *IEEE Trans. Evol. Comput.*, doi: 10.1109/TEVC.2021.3106168.

[78] Z. P. Li, X. M. Wei, X. S. Jiang, and Y. W. Pang, A kind of reinforcement learning to improve genetic algorithm for multiagent task scheduling, *Mathematical Problems in Engineering*, vol. 2021, p. 1796296, 2021.

[79] M. Alicastro, D. Ferone, P. Festa, S. Fugaro, and T. Pastore, A reinforcement learning iterated local search for makespan minimization in additive manufacturing machine scheduling problems, *Comput. Operat. Res.*, vol. 131, p. 105272, 2021.

**Ling Wang** received the BEng degree in automation and the PhD degree in control theory and control engineering from Tsinghua University, China in 1995 and 1999, respectively. Since 1999, he has been at the Department of Automation, Tsinghua University, where he became a full professor in 2008. His current research interests include intelligent optimization and production scheduling. He has authored five academic books and more than 300 refereed papers.

He is a recipient of the National Natural Science Fund for Distinguished Young Scholars of China, the National Natural Science Award (Second Place) in 2014, the Science and Technology Award of Beijing City in 2008, the Natural Science Award (First Place in 2003 and Second Place in 2007) nominated by the Ministry of Education of China. He now is the editor-in-chief of *International Journal of Automation and Control*, and the associate editor of *IEEE Transactions on Evolutionary Computation*, *Swarm and Evolutionary Computation*, *Expert Systems with Applications*, etc.

**Zixiao Pan** received the BEng degree from Wuhan University of Technology, China in 2019. He is currently a PhD candidate at Tsinghua University, Beijing, China. His main research interests include the distributed and green scheduling with intelligent optimization.

**Jingjing Wang** received the BEng and MEng degrees from Tsinghua University, China in 2015 and 2018, respectively. She is currently a PhD candidate at Tsinghua University. Her main research interests include the distributed and green scheduling with intelligent optimization.