Scheduling Optimization and Adaptive Decision-Making Method for Self-organizing Manufacturing Systems Considering Dynamic Disturbances

*ZHANG Yi**, *QIAO Senyu*, *YIN Leilei*, *SUN Quan*, *XIE Fupeng*

School of Automation, Nanjing Institute of Technology, Nanjing 211167, P. R. China

(Received 25 November 2024; revised 23 January 2025; accepted 23 February 2025)

Abstract: The production mode of manufacturing industry presents characteristics of multiple varieties, small-batch and personalization, leading to frequent disturbances in workshop. Traditional centralized scheduling methods are difficult to achieve efficient and real-time production management under dynamic disturbance. In order to improve the intelligence and adaptability of production scheduler, a novel distributed scheduling architecture is proposed, which has the ability to autonomously allocate tasks and handle disturbances. All production tasks are scheduled through autonomous collaboration and decision-making between intelligent machines. Firstly, the multi-agent technology is applied to build a self-organizing manufacturing system, enabling each machine to be equipped with the ability of active information interaction and joint-action execution. Secondly, various self-organizing collaboration strategies are designed to effectively facilitate cooperation and competition among multiple agents, thereby flexibly achieving global perception of environmental state. To ensure the adaptability and superiority of production decisions in dynamic environment, deep reinforcement learning is applied to build a smart production scheduler. Based on the perceived environment state, the scheduler intelligently generates the optimal production strategy to guide the task allocation and resource configuration. The feasibility and effectiveness of the proposed method are verified through three experimental scenarios using a discrete manufacturing workshop as the test bed. Compared to heuristic dispatching rules, the proposed method achieves an average performance improvement of 34.0% in three scenarios in terms of order tardiness. The proposed system can provide a new reference for the design of smart manufacturing systems.

Key words: intelligent manufacturing; adaptive scheduling; self-organizing manufacturing system; reinforcement learning

CLC number: TH166 **Document code:** A **Article ID:** 1005-1120(2025)03-0297-13

0 Introduction

Affected by economic globalization, consumers' demand for products has significantly diversified, personalized and dynamic characteristics, and the update speed of product has accelerated. In order to meet customer needs and adapt to market competition, the manufacturer's production mode has transformed from large-batch assembly lines to small-batch customized production[1]. In customized production, manufacturing workshop needs to realize the flexible allocation of manufacturing resources and the rapid reconstruction of the production line[2]. There are various machine tools in the workshop, and machine tools can fulfill different multiple processing technologies. The production tasks of workpiece include a variety of operations. When the workpiece is assigned to different machines, the time and cost of each operation are not consistent. If the production task is not well matched with the machine tool, it will cause huge waste of resources in the workshop under customized production and may affect the delivery date of products. In workshop en-

*Corresponding author, E-mail address: y.zhang@njit.edu.cn.

How to cite this article: ZHANG Yi, QIAO Senyu, YIN Leilei, et al. Scheduling optimization and adaptive decision-making method for self-organizing manufacturing systems considering dynamic disturbances[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2025, 42(3): 297-309.

http://dx.doi.org/10.16356/j.1005-1120.2025.03.003

vironment, the types of orders, machines, and processing status are diversified and complicated, making it difficult for manual management to effectively control and allocate manufacturing resources.

In order to optimize the workshop production efficiency, the workshop scheduling has become an important research issue. The key of workshop scheduling is to rationally match and sort the production tasks with manufacturing resources, and meet some overall optimization goals (e. g., maximum completion time, minimum production cost, and minimum energy consumption). In the field of workshop scheduling, job shop scheduling problem (JSP) is a classic NP-hard problem. JSP assumes that each type of operation for a workpiece can only be processed on a specific machine tool. However, in actual production, such assumptions are difficult to establish. At present, there are many types of machine tools in the workshop, and multiple machine tools are available for each operation to choose from. With the deepening of research, many flexible extensions of JSP have been developed. Among them, the flexible JSP (FJSP) is more in line with the characteristics of actual workshop. In FJSP, one type of operation can be processed on multiple machine tools. The advantage of deep reinforcement learning (DRL) algorithm is to optimize the sequential decision-making problem, and the DRL model can be trained and optimized without feeding the labeled data. The focus of this paper is to solve the FJSP, which can be described as a type of sequential decision-making problem.

From the literature review, heuristic algorithms (e.g., PSO[3], GA[4-6]), game theory[7], and priority dispatching rules[8] have been investigated to solve the FJSP. Heuristic algorithms obtain the optimal or near optimal solution to the scheduling problem through population iteration and search strategy. However, they are faced with the problem of rescheduling when encountering unpredictable events, which will take up a lot of computing time and lead to poor real-time performance. It is hard to handle dynamic events where lots of real-time decisions are needed to be made quickly. Priority dispatching rules have good generalization and adaptability in solving various kinds of scheduling problems and handling unpredictable events, while the quality of the obtained solutions is not good. After training, the DRL model can be reused to obtain high-quality solutions in a short computing time. When dealing with abnormal events, the DRL model has good generalization, and its accuracy is close to that of meta-heuristic algorithm[9]. To sum up, compared with traditional methods, DRL has the advantages of fast response speed, accurate decision-making, online learning, and strong generalization in solving FJSP. Therefore, DRL is chosen as the decision-making strategy of the proposed system. The comparison of different methods in solving FJSP are shown in Table 1.

Table 1 Advantages of DRL in solving FJSP compared with traditional methods

Method	Heuristic algorithms	Priority dispatching rules	DRL[10-12]
Speed of response	Slow	Fast	Fast
Quality of solution	Good	Poor	Good
Generalization of model	Poor	Good	Good

In addition, previous research has mostly focused on centralized scheduling architecture. All scheduling decision results need to be requested from the central server. This will the non-real-time nature of the decision-making process, which is not conducive to dealing with dynamic events in the workshop. The research contributions of this paper are as follows.

(1) A distributed scheduling architecture based on multi-agent technology is proposed to effectively organize and manage various manufacturing resources. This architecture promotes interaction between machine tools in the workshop, enabling intelligent machines to autonomously generate optimal production strategies to adapt to environmental changes.

(2) An innovative collaborative decision-making mechanism for multi-agent manufacturing system is proposed, where all scheduling decisions are made near physical devices, enhancing the system's self-organizing ability and effectively completing dy-

namic scheduling of small-batch and multi-variety order tasks.

1 System Architecture

In smart workshops, manufacturing equipment supports digital and networked functions, and its manufacturing data can be obtained in real-time. However, there are information islands between manufacturing equipment, and there is a lack of active information exchange between them. During the production process, order schedule can only be passively adjusted to meet changing customer demands. In Industry 4.0, self-organizing capacity is a key indicator for evaluating the intelligence of manufacturing system[13]. In self-organizing manufacturing systems, various machines can autonomously carry out task division and action collaboration, thereby efficiently completing high-volume-multiple-variety order tasks. By embedding industrial personal computers into devices, manufacturing equipment can be intelligently transformed and upgraded to possess environmental awareness, data analysis, and information exchange capabilities.

The overall architecture of multi-agent manufacturing system is proposed as shown in Fig.1. In multi-agent manufacturing system, each of equipment, workpiece, and function module is constructed as an agent[14-17], including job agent (A_J), machine agent (A_M), AS/RS agent ($A_{AS/RS}$), and logistics agent (A_{AGV}). $A_{AS/RS}$ represents the warehouse and is responsible for managing raw materials and finished products. The key function of $A_{AS/RS}$ is to receive and manage personalized orders issued by the cloud platform, and sort these tasks according to the urgency rating to form a task scheduling queue. The logistics system is responsible for the transfer of raw materials and work-in-process between different stations, and is composed of multiple automated guided vehicle(AGV). A_{AGV} represents the functional abstraction of the logistics system, which is used to receive transportation requests from other agents and generate transportation schemes. Once the start and destination of transportation task are determined, A_{AGV} can intelligently choose the most suitable AGV to carry out the transportation operation of the workpiece. A_J records the processing process information and key information (e.g., the turning process is processed on Machine 1) of WIP through RFID tags. Among them, A_M is the key element of multi-agent manufacturing system, and it is the main executor of production tasks and the main initiator of collaborative requests. A_M is not only responsible for information interaction with equipment entities, but also for monitoring and controlling the execution of machining tasks. For example, controlling the robot to grab workpieces from buffer zone to

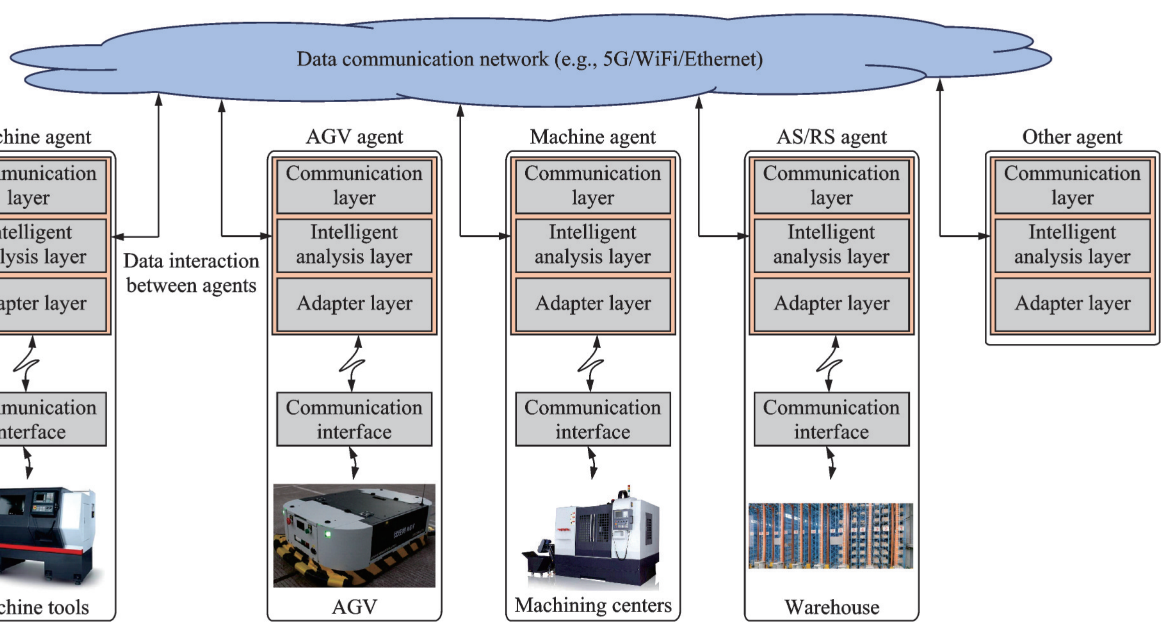

Fig.1 Overall architecture of multi-agent manufacturing system

workbench, as well as reading the processing progress of tasks and the workload of the machine. For other agents, A_M is the main participant in the cooperation and competition process of completing production tasks.

As shown in Fig.1, the internal structure of agent consists of communication layer, adapter layer and intelligent analysis layer. The role of communication layer is to interact with other agents through the Internet of Things and obtain the latest operation status data of other machines. The role of the analysis layer is to analyze and process the content of the acquired information during collaboration process, and produce decision results to control the implementation of production activities. The role of adaptation layer is to support agent to collect manufacturing data through different communication protocol interfaces and send control instructions to the machine based on the decision results.

The analysis layer of $A_{AS/RS}$ and A_M is deployed with intelligent decision-making module, called AI scheduler, which is used to generate scheduling decision after multi-agent negotiation. According to the perceived manufacturing status information and task information, AI scheduler can intelligently select appropriate machines to perform tasks.

The scheduling goal of multi-agent manufacturing system is to efficiently, flexibly, and intelligently complete multi-batch and multi-variety production tasks through negotiation and competition among multiple equipment agents. When the customer submits personalized order, the AS/RS agent decomposes the order into multiple tasks. These tasks are scheduled sequentially in queue order. At the same time, the negotiation process between agents will be triggered. The negotiation mechanism enables the equipment agent to observe the working status and ability of other manufacturing equipment in real time, and to obtain the state features of workshop environment. Then, AI scheduler selects appropriate machine tool to perform production tasks based on state features. In addition, A_{AGV} assigns an AGV to transfer the workpiece to be processed from current station to target station. Different agents continue to cooperate and compete until all tasks are completed.

2 Problem Formulation

In this paper, the application of multi-agent manufacturing system is to solve the dynamic job shop scheduling problem (DJSP) in flexible production processes. In DJSP, there are some jobs $J=\{j_1, j_2, \cdots, j_N\}$ that need to be assigned to different machines for processing. Each job contains multiple operations that need to be processed, and the processing capabilities and nominal operating time required for each operation are inconsistent. Correspondingly, there are M machines $M_t=\{M_1, M_2, \cdots, M_M\}$ available to be requested. Each operation $O_{i,j}$ of j-i can be processed on a set of machine tools $M_c=\{M_{i,1}, M_{i,2}, \cdots, M_{i,x}\}$. Meanwhile, the arrival time of jobs in the manufacturing system is unpredictable and dynamic. This further enhances the complexity of DJSP, which is a typical NP-hard problem. The optimization objective is to allocate the waiting manufacturing tasks to suitable machines at the scheduling time t to minimize the order tardiness. In this paper, the optimization objective is presented in the form of a reward function to optimize the decision model of AI scheduler, which is elaborated in detail in Section 3.3.

The constraints of the DJSP model are as follows

$$\mathrm{st}_{i,j} + x_{i,j,m} \times t_{i,j,m} \leqslant \mathrm{et}_{i,j} \tag{1}$$

$$\mathrm{et}_{i,j} \leqslant \mathrm{st}_{i,j+1} \quad j=1,2,\cdots,h_i-1 \tag{2}$$

$$\sum_{m=1}^{M} x_{i,j,m}=1 \quad i=1,2,\cdots,N; j\in h_i \tag{3}$$

where $x_{i,j,m}$ is the decision variable, representing whether the operation $O_{i,j}$ is assigned to machine M_m for processing; $\mathrm{st}_{i,j}$ and $\mathrm{et}_{i,j}$ are the start and end processing times of operation $O_{i,j}$, respectively; $t_{i,j,m}$ is the actual processing time of operation $O_{i,j}$ on machine M_m, and h_i the set of all operations of Job i. Eqs.(1) and (2) ensure that the subsequent operation of a workpiece can only begin after the previous operation is completed. Eq.(3) represents that at a certain moment, each operation of the workpiece can only be processed by one machine tool.

3 Algorithm Design

This section introduces the key algorithms, including self-organizing collaboration and adaptive decision-making methods. Self-organizing collaboration section elaborates how multiple agents can collaborate and compete during workshop operations. Adaptive decision-making section explains how to generate the optimal scheduling strategy under dynamic environment.

3.1 Self-organizing collaboration among multiple agents

Collaborative intelligence is a key feature of self-organizing manufacturing system[18-20]. In self-organizing manufacturing system, various machines perform cooperation and competition through negotiation mechanisms to efficiently complete task allocation and execution. In this paper, the contract network protocol is applied to design a multi-agent negotiation mechanism to guide self-organizing collaboration among multiple agents. The collaborative behavior occurs between machine agent and logistics agent, as well as between different machine agents. When a workpiece needs to be transported between stations, cooperative behavior between machine agent and logistics agent is required. The production task of a workpiece usually consists of different types of operations, requiring cooperation between machine agents to complete the production task. The competitive behavior occurs between machine agents with the same function. For example, when a "milling" task needs to be scheduled, machine agents with machining capability compete with each other through negotiation mechanisms, and ultimately the most suitable machine is selected to perform the processing task by AI scheduler based on performance metrics. There are three types of scheduling events (i.e. operation completion, urgent job insertion, and machine failure) that can trigger self-organizing collaboration among multiple agents. Among them, "operation completion" is a routine production event, while the rest are disruptive production events.

When the scheduling event is "operation completion", machine agent $\mathrm{A_{M2}}$ that has completed the operation task needs to allocate a new machine tool for processing the next operation of the workpiece (i.e., Job 1), as shown in Figs.2 and 3. Firstly, it is necessary to check whether all operations of Job 1 have been completed. If completed, $\mathrm{A_{M2}}$ will send a transportation request to the logistics agent $\mathrm{A_{AGV}}$. Then, $\mathrm{A_{AGV}}$ assigns an optimal AGV to transport the workpiece to the finished product warehouse. Otherwise, $\mathrm{A_{M2}}$ obtains the attribute information of the next operation (i.e., operation type, expected completion time, and order urgency) by identifying the code of RFID tags, encapsulates the information into a task-announcement, and initiates a new round of negotiation requests to the available machine agents. After receiving the bidding request, machine agents analyze the content of the task-announcement, obtain the operation attributes of the workpiece to be scheduled, and evaluate the task.

The process of task evaluation is divided into two parts. The first part is the preliminary evaluation, whose result determines whether the machine agent will participate in this round of bidding. Based on the remaining buffer length and health status, it is evaluated whether the machine tool is suitable for receiving a new task under current situation. If the buffer is full or machine tool breaks down, the result of task evaluation is that the machine tool is not suitable for undertaking a new task, and the corresponding machine agent will not participate in the bidding. The second one is performance evaluation, in which the machine agent participating in the bidding measures the performance of completing the production task based on the workload and processing capacity of the machine tool and its buffer occupancy. The final evaluation result is encapsulated into a bidding document and fed back to the bid initiator. As shown in Figs. 2 and 3, the evaluation results show that $\mathrm{A_{M1}}$ and $\mathrm{A_{M3}}$ decide to participate in the bidding, while $\mathrm{A_{Mm}}$ decides to refuse the bidding. $\mathrm{A_{M1}}$ and $\mathrm{A_{M3}}$ measure the performance of completing the task based on machine tools' production capacity, and encapsulate the evaluation result into a bidding document and feed it back to $\mathrm{A_{M2}}$.

The content of a bidding document is as follows: $B=\langle \mathrm{op_{id}}, \mathrm{mac_{id}}, \mathrm{workoad}, t_{\mathrm{est}}\rangle$. Here, $\mathrm{op_{id}}$ de-

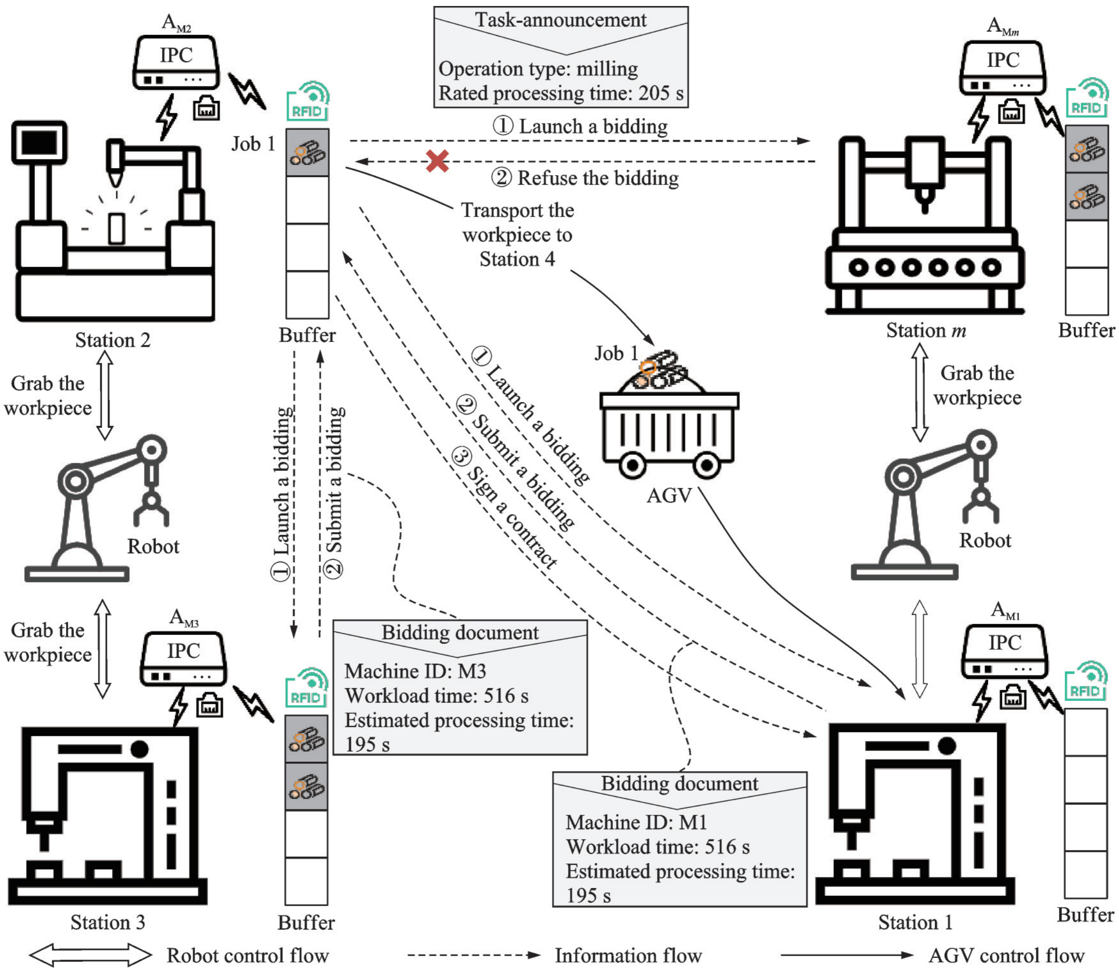

Fig.2 Self-organizing collaborative process between agents

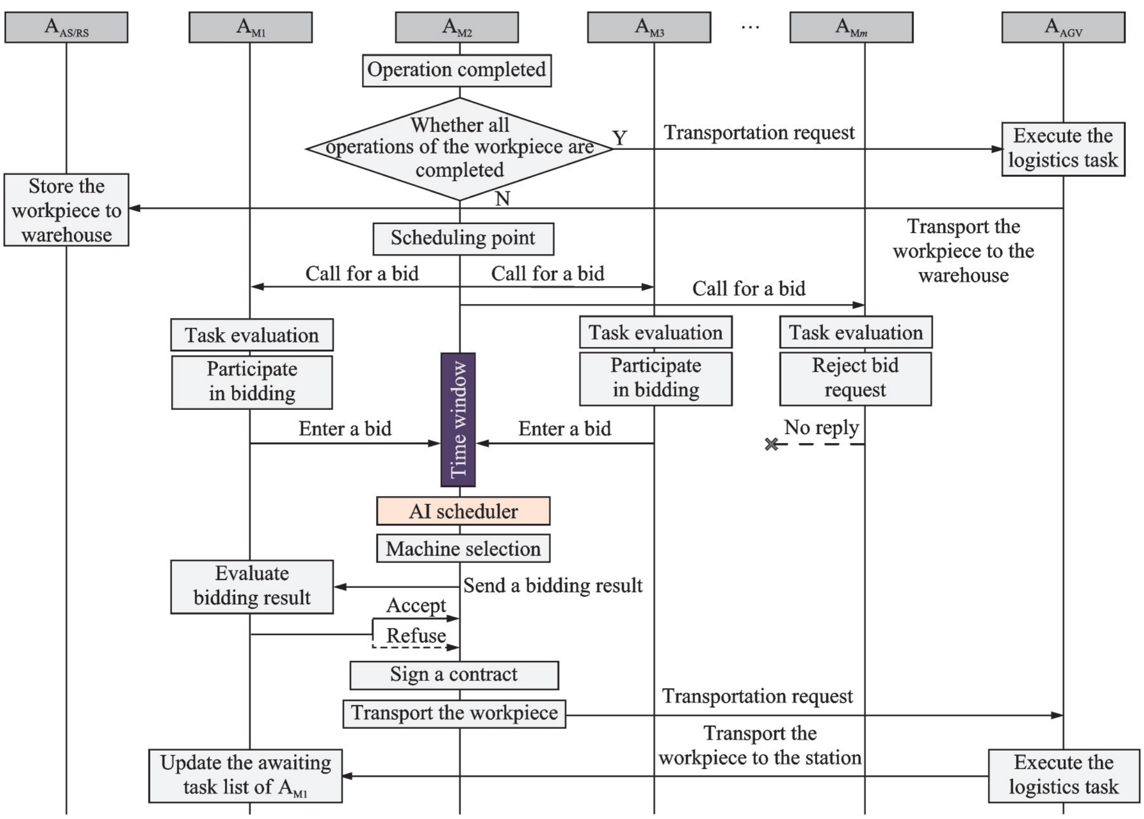

Fig.3 Cooperation and competition between equipment agents in the case of operation completion

notes the operation index of the workpiece to be scheduled, mac_{id} the index of the machine agent participating in the bidding, workoad the working time required for the remaining tasks of the machine, and t_{est} the processing time required for the machine to complete the operation task.

The AI scheduler of A_{M2} selects the most suitable machine agent (i.e., A_{M1}) to perform the task based on the performance metrics of bidding documents. Once the bidding is completed, A_{M1} initiates a request to A_{AGV} for transporting the workpiece. According to task requirements, the appropriate AGV is automatically chosen by A_{AGV} and transports the workpiece to the buffer of M1, and the awaiting task list of A_{M1} is updated subsequently. Then, A_{M1} continues to execute the unfinished tasks in sequence.

3.2 Rapid response strategy for dynamic disturbance events

Urgent job insertion, and machine failure are common disturbance events in manufacturing workshops. It is necessary for manufacturing system to respond and handle disturbance events quickly, otherwise it will reduce overall production efficiency. In this article, a rapid response strategy is designed for various disturbance events.

When the scheduling event is "machine failure", the process of cooperation and competition between equipment agents is shown in Fig.4. In this situation, machine agent A_{M2} checks whether there are any unprocessed operations in its task list. If the task list is empty, A_{M2} enters the standby mode and cannot participate in the negotiation process until Machine M2 resumes operation. Otherwise, A_{M2} initiates a negotiation operation to schedule unprocessed tasks in sequence. First, A_{M2} obtains the attribute information of the pending operation, encapsulates the information into a task-announcement and sends it to available machine agents. Upon receiving the task-announcement, all participator agents (i.e., A_{M1}, A_{M3}, and A_{Mm}) extract and analyze the production task contained in it, and evalu-

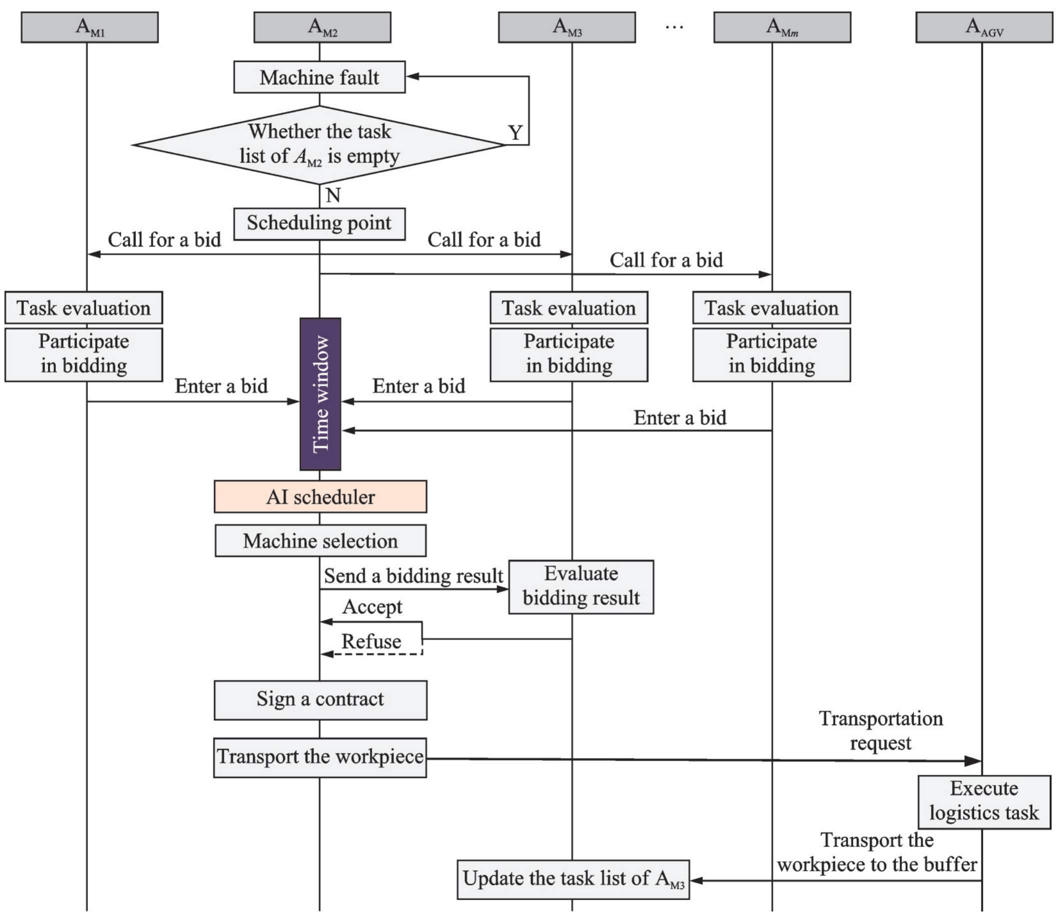

Fig.4 Cooperation and competition between equipment agents in the case of machine failure

ate the task based on the machine tool's processing capacity, and the number of remaining buffers. According to the evaluation results, the machine agents (i.e., A_{M1}, A_{M3}, and A_{Mm}) decide whether to participate in the bidding. As shown in Fig.4, the evaluation results show that A_{M1}, A_{M3}, and A_{Mm} decide to participate in the bidding. A_{M1}, A_{M3}, and A_{Mm} measure the performance of completing the production task based on the workload, processing capacity, and buffer occupancy of corresponding machine tools, and encapsulate the evaluation results into bidding documents and feed them back to A_{M2}. A_{M2} receives bidding documents within the specified time window. The AI scheduler of A_{M2} selects the most suitable machine agent (i.e., A_{M3}) to perform the task based on the performance metrics in the bidding documents. A_{M3} initiates a request to A_{AGV} for transporting the workpiece between stations. According to transportation requirements, the appropriate AGV is automatically chosen by A_{AGV}. The selected AGV transports the workpiece to the buffer of machine M3, and the task list of A_{M3} is updated subsequently. Then, A_{M3} executes the unfinished tasks in the task list in sequence.

When the scheduling event is "urgent job insertion", the process of cooperation and competition between equipment agents is shown in Fig.5. Due to high urgency of the scheduled workpiece, it has the highest priority. Firstly, $A_{AS/RS}$ obtains the first operation attributes of the workpiece and initiates a negotiation request to available machine agents. The obtained attribute information is encapsulated into a task-announcement and sent to the machine agents (i. e., A_{M1}, A_{M2}, A_{M3}, and A_{Mm}). Machine agents conduct task analysis and evaluation based on their own health status and processing capabilities, and provide feedback on the evaluation results to $A_{AS/RS}$ in the form of bidding document. $A_{AS/RS}$ merely receives the bidding documents within the predetermined time window. Then, the AI scheduler of $A_{AS/RS}$ makes decisions for machine selection based on the performance metrics. As shown in Fig.5, Machine M2 is selected to perform production tasks. Finally, $A_{AS/RS}$ collaborates with A_{AGV} to transport the workpieces to the buffer of M2 and update its awaiting task list. When M2 completes the first operation of the workpiece, its next process task is executed in a multi-agent collaborative manner, as shown in Fig.3.

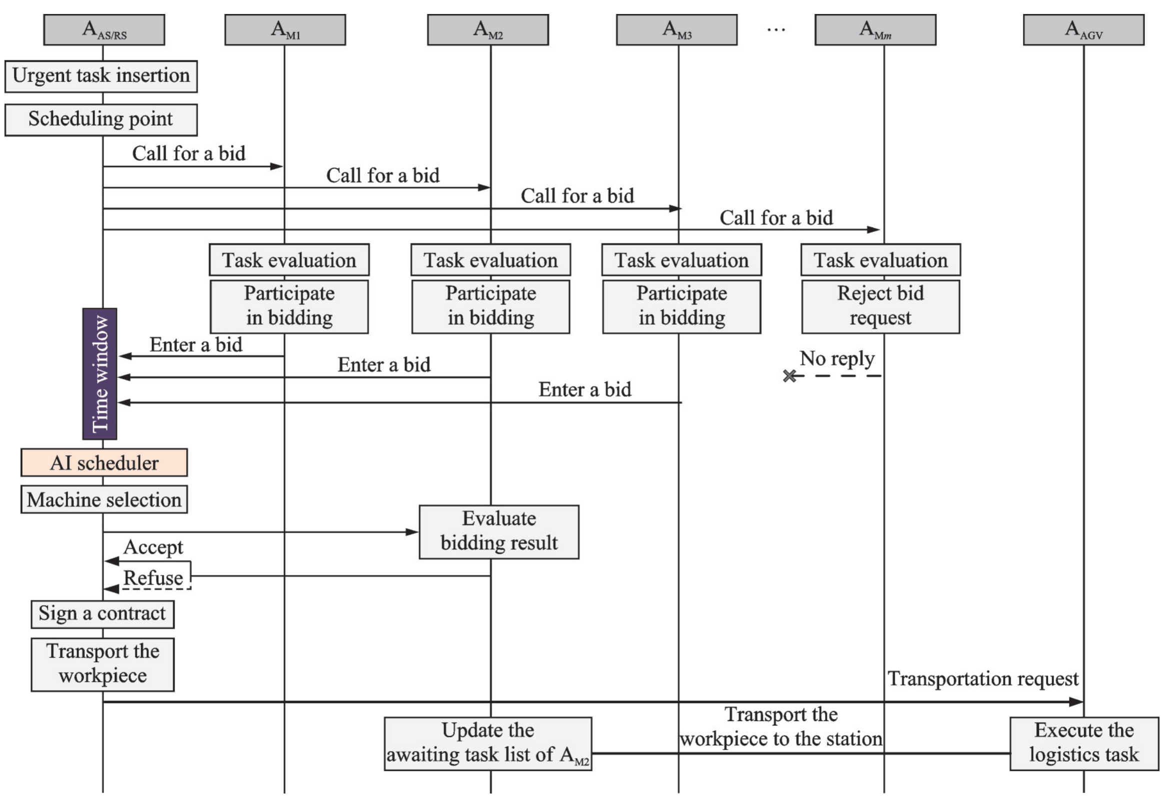

Fig.5 Cooperation and competition between equipment agents in the case of urgent job insertion

3.3 Adaptive decision-making method for dynamic scheduling

In the process of self-organizing collaboration, machine agents can perceive the working status of other machines and obtain on-site manufacturing data. Their internal AI scheduler then generates the optimal production strategy based on environmental status. The decision generation is a key issue during self-organizing production processes, that is, multi-task dynamic scheduling problem.

The scheduling problem can be modelled as a Markov decision process[6]. In this paper, DRL is used to solve the DJSP, that is, the assignment of jobs to machines in a dynamic environment. In DRL models, state features, optional actions and reward functions are key factors for successful implementation. State features $S=\{s_t, t\in T\}$ are an accurate representation of environment and the key basis for AI schedulers to make decisions. The factors that affect the task allocation need to be considered when designing the state features of workshop environment. The assignment of jobs to machines has a direct relationship with the information of jobs and available machines. If one operation needs to be scheduled, it is necessary to investigate the operation attributes of the job (i.e., operation type, expected completion time, and order urgency) and the working status of available machines (i.e., machine type, processing speed, energy consumption, occupation time, and buffer occupancy). Based on attribute information of the job and machines, the AI scheduler can intelligently make an optimal scheduling decision to complete the task allocation, as shown in Fig.6, where DQN represents the deep Q-network.

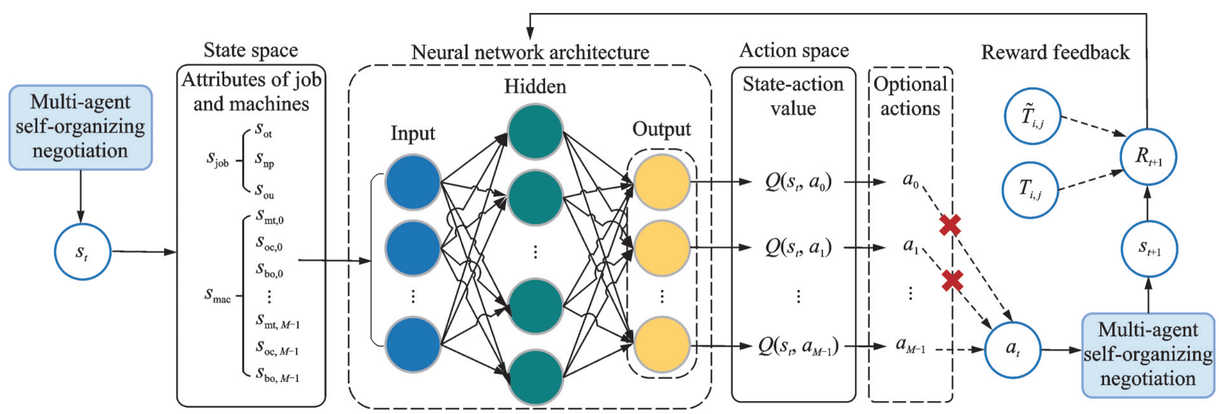

Fig.6 Running process of DQN-based adaptive scheduling method

Thus, the state $s_t=\{s_{\text{job}}, s_{\text{mac}}\}$ of workshop environment consists of the task attributes of jobs and the performance attributes of available machines. $s_{\text{job}}=\{s_{\text{ot}}, s_{\text{np}}, s_{\text{ou}}\}$ includes operation type, nominal processing time, and order urgency. $s_{\text{mac}}=\{s_{\text{mt}}, s_{\text{oc}}, s_{\text{bo}}\}$ includes machine type, occupation time, and buffer occupancy. The action space $A=\{a_t, t\in T\}$ consists of all optional machines $a_t=\{a_i, i\in M_t\}$ in shop floor, where M_t denotes the collection of machines. The production action a_i is to select a machine M_i to perform the task to be scheduled. The reward function R corresponds to the optimization objective of scheduling problem. As shown in Eq.(4), the reward function represents the difference between estimated completion time $\tilde{T}_{i,j}$ and actual completion time $T_{i,j}$ for operation $O_{i,j}$. The completion time represents actual machining time consumed by one operation task. If $\tilde{T}_{i,j}$ is greater than $T_{i,j}$, the reward value will be multiplied by an amplification factor, making the incentive for positive rewards higher. On the contrary, the reward value will be divided by a reduction factor, weakening the suppression of the reverse reward. The above process can be expressed as

$$R=\begin{cases}(\tilde{T}_{i,j}-T_{i,j})\times\varepsilon_{\max} & \tilde{T}_{i,j}>T_{i,j}\\(\tilde{T}_{i,j}-T_{i,j})/\varepsilon_{\min} & \tilde{T}_{i,j}\leqslant T_{i,j}\end{cases}\tag{4}$$

The operation process of task scheduling intelligent decision method based on DQN algorithm is shown in Table 2. The core of the proposed method is the DQN algorithm, and its complexity typically involves two aspects, including time and spatial complexity. The calculation of complexity depends on the algorithm network structure, environmental characteristics, and problem size. The time complexity of the proposed algorithm includes the forward and backward propagation time of the neural network, which is typically expressed as $O(T\times P)$. Here T is the number of iterations and P the total number of parameters in the neural network.

Table 2 DQN-based decision-making method

Algorithm 1: DQN-based scheduling method
(1) Initialize the experience replay memory and workshop environment
(2) Initialize the Q-value function Q with random weights θ
(3) Initialize the target Q-value function Q' with random weights θ
(4) **for** episode$=1$, M **do**
(5) Observe the initial state s of workshop environment
(6) **for** step$=1$, T **do**
(7) AI scheduler selects a random action with probability ε
(8) Otherwise, select $a_t=\operatorname{argmax}_a Q(s_t,a;\theta)$
(9) Perform the action a_t, that is, assign machine a_t to perform the processing task
(10) Observe rewards r_t and new state s_{t+1} after state transition
(11) Store scheduling data (s_t,a_t,r_t,s_{t+1}) in the experience memory
(12) Randomly sample scheduling data from experience memory
(13) Calculate evaluation value $y_i=\begin{cases} r_t & s_{t+1} \text{ is the termination status} \\ r_t+\gamma\times\max\limits_{a_{t+1}} Q(s_{t+1},a_{t+1};\theta') & \text{Otherwise}\end{cases}$
(14) Apply $(y_i-Q(s_t,a_t;\theta))^2$ as loss function to train neural network parameters
(15) Update the parameters of neural network every C step $\theta\leftarrow\theta'$
(16) Continue until all tasks are scheduled

Neural network parameters and experience replay memory are applied to estimate the spatial complexity of the proposed algorithm from a general perspective. Assuming that the neural network has L layers, each layer has N neurons, and the size of experience replay is M. Therefore, the approximate estimation of the spatial complexity is:

(1) The spatial complexity of neural network parameters: $O(L\times N^2)$ (the connections of each layer of neurons are fully connected).

(2) The spatial complexity of experience replay memory: $O(M\times S)$, where S is the size of the space occupied by each experience (state, action, reward, next state).

4 Experimental Verification

As shown in Fig.7, a manufacturing workshop test bed is used as the experimental environment and includes three lathes, three millers and three drillers. The test bed has the capability to handle a variety of operation technologies, including turning, milling, and drilling. Each machine has different attribute values in terms of machining speed. Two robots are used to grab workpieces between buffer and workbench. Each machine has four buffer units for the placement of workpieces. The logistics system consists of two AGVs for transporting material and work-in-progress between workstations. The order management system constantly receives orders from customers and breaks them down into production task queues to be sent to the scheduling system.

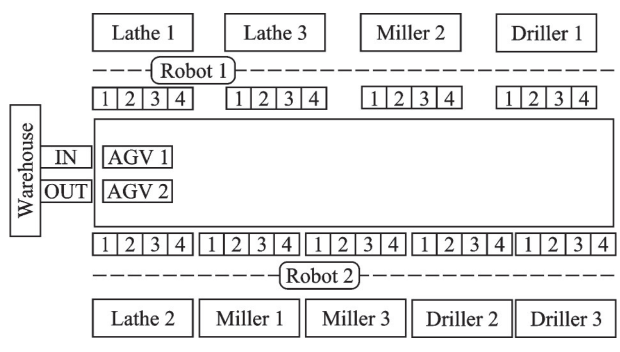

Fig.7 Layout of manufacturing workshop

There are three types of jobs that can be handled by the test bed. Type 1 consists of two operations: turning and milling. Type 2 consists of two operations: milling and drilling. Type 3 consists of three operations: turning, milling, and drilling. The warehouse is applied to store raw materials and finished products, and to initialize the RFID tag information adhered to the workpiece, and the RFID tag is responsible for recording and tracking the manufacturing process of workpiece.

In this case, machine agent is used to obtain the working status of other agents and the characteristics of workshop environment through self-organizing collaboration mechanisms, ultimately supporting AI scheduler to generate production strategies. The internal of AI scheduler is a deep learning model consisting of a three-layer neural network structure with the following parameter settings: an input layer of 31 neurons, a hidden layer of 200 neurons and an output layer of nine neurons. The specific values of ε_{max} and ε_{min} are set to 10 and 5, respectively. In the experiment, a customer order consisting of 20 different types of workpieces is randomly generated and fed into the order management system. To simulate the event of urgent order disturbance, two new orders arrive at 50 s and 100 s respectively during the scheduling process. To verify the effectiveness of the proposed method, three cases are designed in the experimental section: N20U0, N20U10, and N20U10U10. N20U0 indicates that a customer order consisting of 20 different types of workpieces arrives at the initial time and there are no urgent orders. N20U10 indicates that a customer order consisting of 20 workpieces arrives at the initial time, and an urgent order consisting of ten workpieces arrives at 50 s. N20U10U10 indicates that a customer order consisting of 20 workpieces arrives at the initial time, an urgent order consisting of ten workpieces arrives at 50 s, and another urgent order consisting of ten workpieces arrives at 100 s. The training curve of the DRL-based scheduler is shown in Fig.8 and eventually converges at 402, demonstrating the learning feasibility of the proposed method.

Five different methods are used to compare performance on three test cases, namely shortest queue (SQ), shortest processing time (SPT), shortest remaining (SR) processing time, Random and the proposed method. The cumulative tardiness time is used to evaluate the performance of the scheduling results, which is calculated as $\sum_{i\in N, j\in h_i} (\tilde{T}_{i,j} - T_{i,j})$.

Experimental results obtained by five different methods are shown in Figs.(9—11). In the N20U0 case, the proposed method achieves the best performance, while SQ, SPT, SR and Random achieve 95.5%, 22.4%, 93.3% and 97.3% performance, respectively. In the N20U10 case, the proposed method achieves the best performance, while SQ, SPT, SR and Random achieve 92.0%, 6.5%, 91.9% and 72.6% performance, respectively. In the N20U10U10 case, the proposed method achieves the best performance, while SQ, SPT,

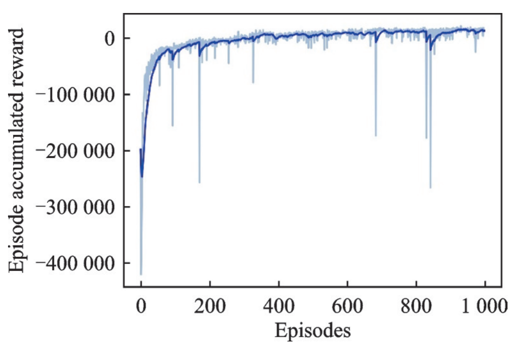

Fig.8 Training curve of DRL-based scheduler

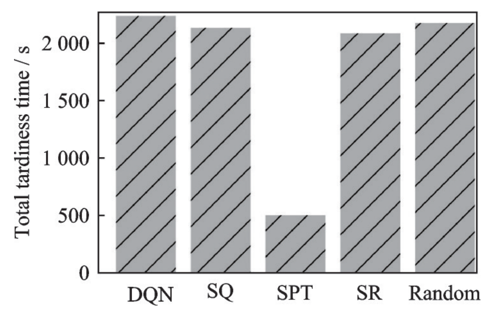

Fig.9 Comparison of scheduling results in N20U0

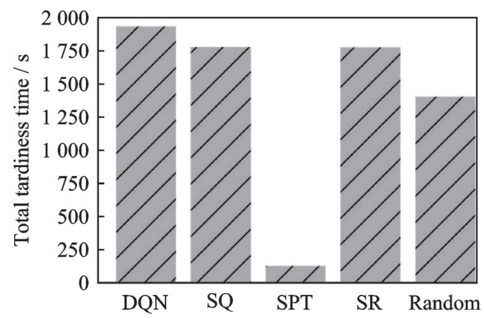

Fig.10 Comparison of scheduling results in N20U10

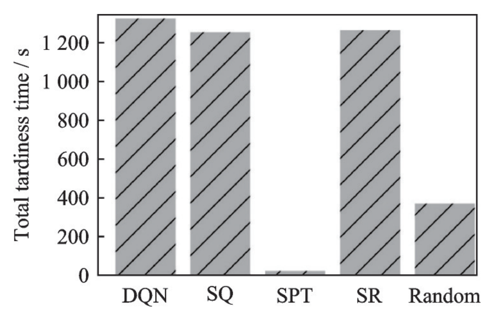

Fig.11 Comparison of scheduling results in 20U10U10

SR and Random achieve 94.8%, 1.7%, 95.5% and 28.0% performance, respectively. In summary, compared to four rule-based methods, the proposed method achieves the best performance in terms of total tardiness time, and its advantages are more pronounced under the interference of urgent order events. The proposed method has strong scalability in actual large-scale manufacturing systems. Due to the self-organizing nature of multi-agent manufacturing system, other machines can be easily and flexibly added. As the number of machines increases, the scheduling performance advantage of the proposed method will become more apparent.

5 Conclusions

In order to improve the organizational efficiency and restructuring flexibility of manufacturing workshops under customized production mode, this paper proposes a self-organizing manufacturing system based on multi-agent technology and deep reinforcement learning methods. In self-organizing manufacturing systems, each machine is modeled as an agent with intelligent analysis and information exchange capabilities, and can autonomously collaborate with other machines under dynamic disturbances. At a decision-making moment, each machine agent can perceive environmental features and independently generate the optimal production strategy. Ultimately, the allocation of all production tasks and the response to dynamic events are well achieved in a self-organizing form.

In future work, industrial knowledge should be incorporated into the scheduling decision-making process of manufacturing systems. In addition, novel multi-agent models and reinforcement learning methods deserve to be considered for the design of self-organizing manufacturing systems.

References

[1] QIN Z, LU Y. Self-organizing manufacturing network: A paradigm towards smart manufacturing in mass personalization[J]. Journal of Manufacturing Systems, 2021, 60: 35-47.

[2] ZHOU J, LI P, ZHOU Y, et al. Toward new-generation intelligent manufacturing[J]. Engineering, 2018, 4(1): 11-20.

[3] ZHANG Y, ZHU H, TANG D. An improved hybrid particle swarm optimization for multi-objective flexible job-shop scheduling problem[J]. Kybernetes, 2020, 49(12): 2873-2892.

[4] DAI M, TANG D, GIRET A, et al. Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints[J]. Robotics and Computer-Integrated Manufacturing, 2019, 59: 143-157.

[5] DAI M, WANG L, GU W, et al. Research on flexible flow-shop scheduling problem with lot streaming in IOT-based manufacturing environment[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2020, 37(6): 831-838.

[6] AI Y, WANG M, XUE X, et al. An efficient heuristic algorithm for flexible job-shop scheduling problem with due windows[C]//Proceedings of 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE). [S.l.]: IEEE, 2022: 142-147.

[7] WANG J, ZHANG Y, LIU Y, et al. Multiagent and bargaining-game-based real-time scheduling for internet of things-enabled flexible job shop[J]. IEEE Internet of Things Journal, 2018, 6(2): 2518-2531.

[8] ZHANG F, MEI Y, NGUYEN S, et al. Multitask multiobjective genetic programming for automated scheduling heuristic learning in dynamic flexible job-shop scheduling[J]. IEEE Transactions on Cybernetics, 2022, 53(7): 4473-4486.

[9] JOHNSON D, CHEN G, LU Y. Multi-agent reinforcement learning for real-time dynamic production scheduling in a robot assembly cell[J]. IEEE Robotics and Automation Letters, 2022, 7(3): 7684-7691.

[10] NGUYEN T T, NGUYEN N D, NAHAVANDI S. Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications[J]. IEEE Transactions on Cybernetics, 2020, 50(9): 3826-3839.

[11] LIN C C, DENG D J, CHIH Y L, et al. Smart manufacturing scheduling with edge computing using multiclass deep Q network[J]. IEEE Transactions on Industrial Informatics, 2019, 15(7): 4276-4284.

[12] ZHOU T, TANG D, ZHU H, et al. Reinforcement learning with composite rewards for production scheduling in a smart factory[J]. IEEE Access, 2020, 9: 752-766.

[13] XU X, LU Y, VOGEL-HEUSER B, et al. Industry 4.0 and Industry 5.0—Inception, conception and perception[J]. Journal of Manufacturing Systems, 2021,

61: 530-535.

[14] ZHANG Y, ZHU H, TANG D, et al. Dynamic job shop scheduling based on deep reinforcement learning for multi-agent manufacturing systems[J]. Robotics and Computer-Integrated Manufacturing, 2022, 78: 102412.

[15] BI M, CHEN G, TILBURY D M, et al. A model-based multi-agent framework to enable an agile response to supply chain disruptions[C]//Proceedings of 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE). [S. l.]: IEEE, 2022: 235-241.

[16] OWLIYA M, SAADAT M, ANANE R, et al. A new agents-based model for dynamic job allocation in manufacturing shopfloors[J]. IEEE Systems Journal, 2012, 6(2): 353-361.

[17] ZHU H, CHEN M, ZHANG Z, et al. An adaptive real-time scheduling method for flexible job shop scheduling problem with combined processing constraint[J]. IEEE Access, 2019, 7: 125113-125121.

[18] WANG L P, TANG D B, SUN H W, et al. Enabling technology of multiagent manufacturing system: A novel mode of self-organizing IoT manufacturing[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2021, 38(5): 876-892.

[19] HUANG J, HUANG S, MOGHADDAM S K, et al. Deep reinforcement learning-based dynamic reconfiguration planning for digital twin-driven smart manufacturing systems with reconfigurable machine tools[J]. IEEE Transactions on Industrial Informatics, 2024, 20(11): 13135-13146.

[20] QIN Z, JOHNSON D, LU Y. Dynamic production scheduling towards self-organizing mass personalization: A multi-agent dueling deep reinforcement learning approach[J]. Journal of Manufacturing Systems, 2023, 68: 242-257.

Ackowlegements This work was supported by the Scientific Research Foundation of Nanjing Institute of Technology (No.YKJ202425) and the National Natural Science Foundation of China (No.72301130).

Author

The first/corresponding author Dr. **ZHANG Yi** received his Ph.D. degree in mechanical engineering from Nanjing University of Aeronautics and Astronautics in 2024. He is currently an associate professor in Nanjing Institute of Technology. His research interests include the design of multi-agent manufacturing systems, and smart manufacturing control.

Author contributions Dr. **ZHANG Yi** conducted the study and wrote the paper. Mr. **QIAO Senyu** contributed to data and model components for the decision-making model. Dr. **YIN Leilei** designed the case study. Dr. **SUN Quan** contributed to the conclusion and background. Dr. **XIE Fupeng** contributed to the data analysis. All authors commented on the manuscript draft and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: WANG Jing)

考虑动态扰动的自组织制造系统调度优化与自适应决策方法

张 毅，乔森雨，殷磊磊，孙 权，谢富鹏

(南京工程学院自动化学院，南京 211167，中国)

摘要：制造业生产模式呈现出多品种、小批量和个性化的特点，导致车间扰动频发。传统的集中式调度方法难以在动态扰动下实现高效实时的生产管理。为提升生产调度的智能性和适应性，提出新型分布式调度架构，赋予制造系统自主任务分配与扰动处理能力。此架构下，生产任务通过智能机器间自主协作与决策得以高效调度。首先，利用多智能体技术构建自组织制造系统，使每台机器都具备主动信息交互和联合行动执行的能力。其次，设计多类型自组织协作策略，促进了多个主体之间自主协商交互，实现对全局环境状态的感知。为确保在动态环境中生产决策的适应性和优越性，采用深度强化学习来构建智能调度器，基于实时环境状态智能地生成最优生产策略，以指导任务分配和资源配置。最后以离散制造车间为试验台，通过3种场景验证所提出方法的可行性和有效性。与启发式调度规则相比，在订单按期交付方面，提出方法在3种场景下的平均性能提高了34.0%。所提出的自组织制造模式可为智能制造系统的设计提供新的参考。

关键词：智能制造；自适应调度；自组织制造系统；强化学习