

Received September 3, 2021, accepted September 22, 2021, date of publication September 23, 2021, date of current version October 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3115339

Research on Workers Integration in Smart Factories With Multi-Agent Control System

ZEQUN ZHANG¹, (Graduate Student Member, IEEE), DUNBING TANG¹, AND QINGWEI NIE¹

College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Corresponding author: Dunbing Tang (d.tang@nuaa.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1710500 and Grant 2018YFE0177000, and in part by the National Natural Science Foundation of China under Grant 52075257.

ABSTRACT In today's modern manufacturing environments, Multi-Agent Manufacturing System (MAMS) has been a fundamental approach for developing industrial applications that can cope with complexity, uncertainty, and dynamicity. It has become a significant resource for improving smart factories. However, little attention has been paid to integrate human workers into these overall control systems efficiently. This article is intended to take a step forward and propose a human worker integration scheme under MAMS. The multi-skilled feature of workers and the collaboration between workers are considered in this research. First, a Worker Agent (WoA) model with wearable devices is proposed as a critical design building block. After that, based on the conventional hybrid manufacturing control architecture (HyMCA), a WoA integration interface is investigated to implement the MAMS model. Then, a working mechanism of WoA in the task allocation process is studied. After that, a matching algorithm that considers different scenarios between workers and tasks is proposed based on the Hungarian Algorithm and Genetic Algorithm. Finally, program simulations and actual experiments are carried out to verify the effectiveness of the worker integration mechanism proposed in this paper.

INDEX TERMS Multi-agent manufacturing system, smart factory, hybrid manufacturing control architecture, worker agent.

I. INTRODUCTION

As the manufacturing industry evolves toward socialization and individualization of markets, customer requirements have become personalized and dynamic. In this new scenario, enterprises continuously receive orders coming in small-batches with uncertain time, random quantity, and changing priorities. The uncertainties of orders are getting stronger. In order to make the manufacturing workshop adapt to the new market environment better, the Multi-Agent Manufacturing System (MAMS) approach was proposed [1]. In MAMS, the workshop physical components and functions are abstracted into entities (agents) with autonomous decision abilities and cooperation capabilities [2]. In this way, the decisional entities (agents) can work together to quickly and efficiently react to events instead of wasting time requesting control decisions from a central unit.

The associate editor coordinating the review of this manuscript and approving it for publication was Tiago Cruz¹.

However, little attention has been paid to the efficient integration of human workers in the emerging context of MAMS. This is entirely in agreement with [3] that they draw attention to human factors as critical elements in addressing new and unpredictable behaviours in smart factories. Although the improvement of automation and intelligence has reduced manual intervention and can implement unmanned factories in specific environments, the efficient integration of workers in smart factories is still an open problem.

For a long time in the past, although the level of automation and intelligence of equipment within the shop floor has been improved, the working environment and scheduling system of workers are still the same as traditional processing workshops, especially in developing countries. In recent years, more and more people realized that the worker began to detach from the smart factory system control and, sometimes, became a constraint on the subsequent development of smart factories. To solve this problem, approaches such as "Made in China 2025", which is currently implemented in China, have started to focus on integrating workers into the smart

factory system control efficiently. In this plan, an approach is to equip workers with wearable mobile devices, which usually have image output and voice service functions, along with positioning and health monitoring sensors that monitor the workers. This method is typically used to assist workers in better-performing tasks in smart factories, such as improving efficiencies, reducing fatigues, and reducing work intensities [4]. In the research field of MAMS, this approach can also be used to integrate workers into the system control by developing Work Agents (WoAs) associated with each worker at the software level [5], [6].

WoAs are different from traditional agents, which can collect machines' status autonomously and control the machines. A WoA itself cannot control the worker's behaviour, and it is challenging to manage worker status autonomously. WoAs rely on wearable devices to dispatch tasks to the corresponding workers, but how and when the workers perform the tasks are not determined. When the workers complete the tasks, they can inform WoAs through the interactive interfaces of the wearable devices. In this way, WoAs can be regarded as black-box agents and integrated into the MAMS. Due to the uncontrollability of human behaviour, how a WoA participates in the interaction and collaboration with controllable agents (such as machine agents and part agents) needs further research. However, there are currently few articles on integrating WoA into the dynamic scheduling mechanism of MAMS. Most investigations focus on how to provide workers with an immersive experience through wearable devices to improve their work efficiency [7].

Furthermore, we need to consider that workers' functions also evolve with the improvement of the manufacturing industry. This is mainly manifested in the following two aspects:

- 1) With the improvement of the intelligence of the workshop, the amount of labour positions is gradually reduced, and the quality of workers continues to increase, which leads to workers having more skills in modern manufacturing systems. For example, in a processing workshop composed of CNC machine tools, almost all workers can use these machine tool systems proficiently and complete the pre-processing preparations.
- 2) Due to order uncertainty, random production events (e.g., fixture replacement, equipment maintenance) sometimes require multiple workers' cooperation to solve them. By investigating some part-processing workshops in China, we have found that workers usually form a temporary team to handle a task in many cases. After completing the task, the quick team will be disbanded.

The multi-skilled feature of workers prevents them from being limited to a single job. The collaboration between workers makes arrangements more complicated. In many cases, tasks that need to be handled by multiple workers are difficult to find enough workers immediately. It is necessary to pre-select suitable candidates for the tasks in advance.

In order to deal with the above problem, this article proposes a multi-agent interaction mechanism integrated with WoAs to assign tasks to workers.

This article is organized as follows. The research work related to this article is presented in section 2. Section 3 introduces a worker integration mechanism in MAMS. Moreover, a feasible solution to integrate WoAs under HyMCA is also proposed. After that, in section 4, the effectiveness of this integration mechanism is verified through an actual experiment. Section 5 summarizes the work presented in this article.

II. RELATED WORK

The main goal of this article is to study the efficient integration of workers into manufacturing systems based on traditional MAMS. This integration must ensure the correct operation of the original MAMS system, including task allocation between machines and parts, autonomous operation of the device, and initial optimal scheduling. Simultaneously, the integrated system should be able to improve the efficiency of workers.

A. MULTI-AGENT MANUFACTURING SYSTEM

The concept of Agent evolved from the research of Distributed Artificial Intelligence (DAI) systems in the 1990s [8]–[10]. An agent is a computational mechanism that exhibits a high degree of autonomy, performing actions based on information received from the environment [11]. A network of agents will create a multi-agent system that aims to provide both principles for constructing complex systems involving multiple agents and mechanisms to coordinate independent agents' behaviours [9], [12]. The entities in the workshop can be abstracted into agents and given different characteristics. The Machine Agent (MA) and Part Agent (PA) proposed by Krothapalli and Deshmukh [1] are the two most widely used agents in the construction of agent-based smart factories.

In MAMS, autonomy and negotiation are fundamental features to promote workshop operations [13], [14]. These two features are inherent to agent-based approaches. Thanks to their autonomous quality: agents combine their calculation power with local knowledge to decide individual behaviour. Simultaneously, thanks to the agents' negotiation capability (through the interaction between individuals): agents can jointly promote a specific behaviour, such as part operation, tooling distribution, and logistics planning. As a result, the workshop layer can realize a self-organized process through MAMS. MAMS is an essential means to realize smart factories [9], especially in parts processing workshops.

B. CONTROL ARCHITECTURE

The traditional Computer Integrated Manufacturing (CIM) system adopts a layer-based centralized architecture, which is named hierarchical manufacturing control architecture (HiMCA) [15]. The main characteristics of such systems are the top-down structure, fixed multi-layers of hierarchies (parent and child), and a centralized decision-making unit.

These systems present good optimization in mass production, and it is also possible to find the optimal set-up parameters for specific production equipment. However, the drawbacks of fragility and lack of flexibility limit the production performance in short series manufacturing [16]. In this kind of environment, production technology is often changed. Production tools have to be adjusted to specific products, and the process organization must follow these changes to avoid reduction or losses resulting from non-productive time gaps. Furthermore, with the increasing size and scope of Hierarchical/centralized planning systems, the structural complexity of these systems grows rapidly [17].

Duffie [18] proposed the heterarchical control architecture (HeMCA). In this architecture, the physical components (e.g., machines and parts) and functions (e.g., monitoring and task distribution) in the workshop are abstracted into entities with independent decision ability. This architecture is very similar to the design idea of MAMS and has become the most commonly used system architecture of MAMS for some time. Shen *et al.* [19] and Leitão [20] conducted several high-quality surveys for this architecture. Compared with HiMCA, HeMCA is no longer for finding the optimal solution of the system but for better adaptability. This architecture has strong robustness. It can efficiently deal with disturbances and is more adaptable to the modern manufacturing environment. However, HeMCA lacks the means to achieve high performance, limiting its application in large enterprises [9], [15].

Hybrid manufacturing control architecture (HyMCA), which is also called semi-HeMCA in some works [15], is a mixture of hierarchical and heterarchical controls. It applies the benefits of both systems while avoiding their shortcomings. This type of structure usually has an additional layer as the system's supervisor, in which the primary function is to maintain overall performance [21]. For instance, in [22], the supervisor level manages the coordination between the fundamental entities and ensures global performance. If a perturbation occurs, the low level enters into a reactive mode and manages their routing with pheromones without asking the supervisor level. The work described in [21] and [23] proposes a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling. The authors in [9] believe that HyMCA will become the mainstream control architecture used by smart factories in the future.

C. HUMAN-MACHINE COOPERATION FOR SMART FACTORIES

A growing number of researchers are addressing human-machine cooperation designs in the domain of industrial engineering. Concerning the industrial state of the art in this domain, numerous studies are underway to integrate the human operator in manufacturing processes correctly. Still, they are conducted mainly at manufacturing ergonomics [7]. For instance, a framework that relied on immersive technologies and intelligent personal digital assistants is defined and implemented in [4] for augmented operators in Industry

4.0. In this framework, the personal digital assistant answers questions as an expert, and the immersive technologies are explored to provide interactive, absorbing experiences. In [24], it is proposed to adopt digital simulation set-ups to support the human-centred design of manufacturing workstations to enhance workers' interactive experience.

There is less research on integrating workers into the overall control system architecture, but they are constantly emerging. In many MAMS literature, authors may define Work Agent but essentially treat the worker as a machine described in detail in [3]. In response to these issues, researchers in [7] designed a human-oriented system control framework. In [25], it is discussed how to integrate human behaviour into factory simulation to make the simulation program results more in line with expectations. In [6], combined with the manufacturing situation of developing countries, it is proposed that workers can use mobile devices as the information access interface of the system, and the system control architecture is designed based on this.

D. RELATED WORK SUMMARY

In general, the theoretical development of MAMS has become more and more consummate, and the research focused has gradually shifted from how to obtain an excellent model to how to get better applications in practice and how to better conform to the development trend of manufacturing. There are still many issues worthy of discussion in these research directions [9].

The focus of this article is how to incorporate workers into the overall control system under MAMS. Workers are not the same as machines. They are more complex and more random. Under the research premise of this article, the WoA is different from other agents in the factory environment. It is no longer a controller but a worker's assistant. More detailed internal structure design will be discussed in the next section. Simultaneously, this article will use HyMCA as the system control architecture to study how WoAs assist workers to participate in the manufacturing process autonomously instead of a mode of artificially designated tasks.

III. A WORKER INTEGRATION MECHANISM IN MAMS

This article proposes a specific implementation to integrate worker agents into traditional MAMS. Under this implementation, WoAs are added as black box agents to the interactive system of MAs (Machine Agents) and PAs (Product Agents), assisting workers in undertaking tasks dynamically and ultimately improving overall worker efficiency. To this end, in this section, the life cycle of MAs, PAs, and workers is described first. Then, a multi-agent control system for processing workshops is designed. Under this control system, the integration scheme and the task allocation mechanism of the worker agent are proposed.

A. AGENTS COMBINED WITH PHYSICAL ENTITIES AND WORKERS

As shown in Fig. 1, machine tools of different brands are distributed discretely in the processing workshop and

are allowed for addition, removal, and interchange. Automatic Guided Vehicles (AGVs), along with robotic arms, are usually used to move materials, components, and tools between the buffer stations of the corresponding machine tool. Finished products, raw materials, and tooling are stored and managed by Automated Storage and Retrieval System (AS/RS). As introduced in section 1, in smart factories, workers are equipped with wearable mobile devices, which usually have image and voice interfaces and can be connected to the control layer.

1) MACHINE AGENT AND PART AGENT

Functional units (FUs), with a particular machine as the main body, are divided first on the shop floor. Fig. 2 shows a milling FU located in a processing workshop laboratory. The FU contains a CNC milling machine as its main body. It also contains buffer stations, RFID read/write equipment and sensors. We propose an Agent Computing Node (ACN) as the intelligent control node for this FU (see Fig. 2), which will give it the ability to make independent decisions and autonomous behaviours.

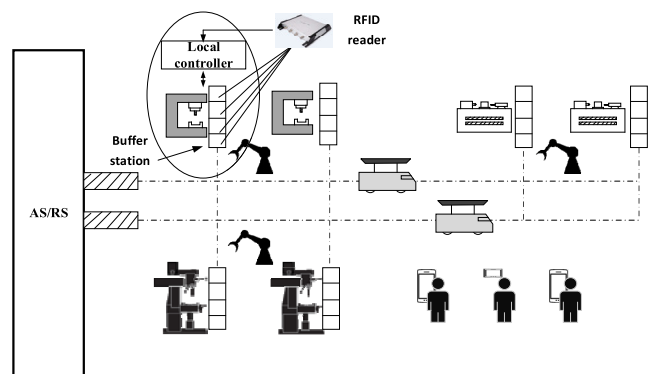


FIGURE 1. The processing workshop.

The ACN software consists of three layers, including the adaptation layer, intelligent analysis layer, and JADE-based information development layer. The adaptation layer is used to interconnect with the machines. The protocol for communicating with devices of different brands is encapsulated in the “Link library.” Simultaneously, the machine’s action drive and information collection modules are designed according to its type and function. The intelligent analysis layer is the core part of building intelligent individuals.

On the one hand, the information from FUs and the environment is analyzed here. On the other hand, it is the agent threads’ management container that sends information to the corresponding agents and transfers the agent decisions to the corresponding information interfaces. The JADE-based information exchange layer is used for interaction with other ACNs. JADE is an agent development framework based on the JAVA language [26]. It encapsulates the interface of message exchange.

Parts are the execution objects of the processing workshop. In our research environment, the antennas of RFID readers

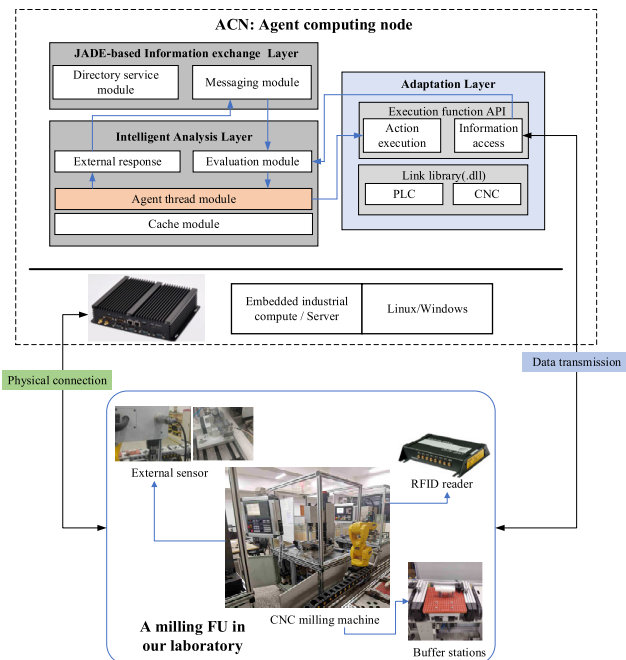


FIGURE 2. A milling FU located in a laboratory. FU: functional unit.

are installed in the buffer stations attached to the FUs. Pallets that hold the parts are equipped with RFID tags. The ACN of the AS/RS gets order information from the order database. Before the task is delegated, the AS/RS will negotiate with the processing FUs through ACNs to determine the location of the first processing step of the part. When parts leave the AS/RS, they will be written with relevant information. When parts arrive at each station, they can be quickly sensed by ACNs, and all information about the current task can be obtained simultaneously.

As shown in Fig. 3, ACNs will serve as containers for MA and PA execution. Taking a processing FU as an example, the MA program in the ACN is responsible for the operation of the unit itself. When a part arrives, the information of it is perceived as a task by the current ACN. At the virtual level, the agent program corresponding to this part is in a silent state at this time.

When there is no task in the processing machine, the task will be selected for processing from the buffer task list. If the part is determined, the part’s data will be combined with the supporting program to generate the PA program. The ACN acts as the container for this PA to run until this part is processed, and the next undertaking machine is determined through negotiation and interaction. After that, this PA program will go offline. At this moment, the part becomes a piece of data at the virtual level again.

The PA is usually a piece of data at the virtual level and will be generated by ACN when needed. The intelligent analysis layer of the ACN is responsible for managing the threads of MA and PA. In this mode, PA will not always use computing power and memory. The necessary data is stored in the RFID tag for generating PA threads and circulates along with the part in the workshop.

2) WORKER AGENT

Unlike MA and PA, WoAs cannot directly perceive or control objects (i.e., workers). It establishes connections with things through graphical interfaces or voice services. As mentioned in [3], the behaviour of workers is unpredictable and highly random. For example, the execution time of a manual task cannot be accurately predicted, and workers may temporarily leave their positions for any reason.

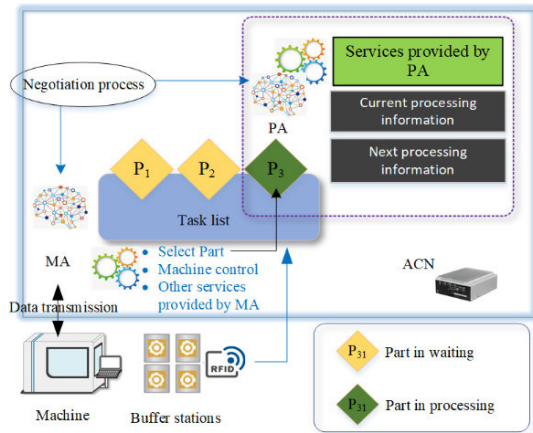


FIGURE 3. MA's and PA's Life Cycle in an ACN. MA: machine agent. PA: part agent. ACN: agent computing node.

The WoA cannot directly make decisions. The goal/utility of WoA is to convey information and assist workers in making judgments. After the workers are integrated into the MAMS system, the tasks will be directly communicated to the workers through the graphical interface of the wearable device, which is different from the previously manual task assignment. At the same time, the details of the task are conveyed, and the workers can choose to accept or not according to their situation. Considering the above factors, this article proposes the WoA model shown in Fig. 4.

The WoA is designed by two modules, including “information processing module” and “interactive module”.

The “Interactive module” is responsible for the information interaction with workers. It presents the task information to the worker through a graphical interface or audio. The worker will respond to this information and be able to actively respond with the information required to the upper level, such as whether the task is currently active and whether the task undertaken has started or ended. When a particular event occurs, workers can also give feedback to the upper level through the voice service of the wearable device or the options of the graphical interface. The accessory functions of the wearable device itself will also be fed back through the “Interactive module”. This part of information does not come from the active report of the worker, but the sensors attached to the wearable device, such as the current worker’s location, worker’s health status information, etc.

The “Information processing module” is used to process the “Interactive module” information. On the one hand,

the task information from the outside will be filtered here, trigger the built-in event response program, and complete the content presented through the “Message pushing” part of the “Interactive module”. On the other hand, the information from the workers and their sensors will be classified and stored here, and essential historical data (e.g., the start and end time of each task of the worker, the length of work, etc.) will be synchronized with the upper database. The information confirmed by the worker will be encapsulated according to the specific content and fed back to the upper system through the “External information interface” in the “Interactive module”. It is also a mobile interface developed through the JADE framework.

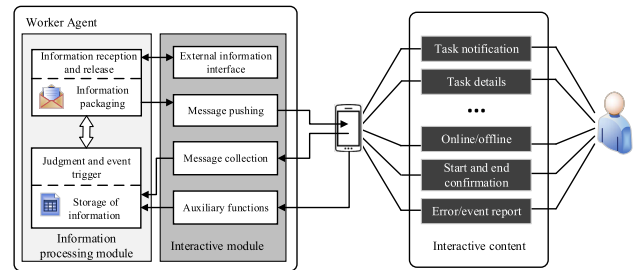


FIGURE 4. Worker agent model.

The WoA is similar to an information transfer and delivery platform when it communicates with the worker. Workers can access the overall system through this platform, handle tasks more flexibly, and report their status more dynamically.

B. THE INTEGRATION FOR WORKER AGENT UNDER HYMCA

As mentioned in section 2, HyMCA is applied as the control architecture for MAMS in this article. Using ACNs, intelligent improvement for a processing workshop can be achieved. The FUs together with ACNs constitute a heterarchical control level, as shown in Fig. 5, and we call this level a “self-organizing operation layer”. Individual autonomy and swarm negotiation are the main behaviours of this layer. In this model, the negotiation between MA and PA follows the contract net mechanism [27].

Above the self-organizing operation layer, a high-level is proposed regarding the general form of HyMCA [21]. We call this level the “supervisor layer”. Usually, the main responsibility of this layer is to optimize the system. Still, in this article, we mainly focus on how to integrate workers into the MAMS and how to improve the efficiency of workers. A worker management module is set up in the supervisor layer, and through WoAs, workers are registered in it. The registered workers can decide whether to be online/offline and whether to participate in the assignment of tasks through the graphical interface provided by the wearable device.

Because a WoA cannot make direct decisions, the worker management module also assumes the responsibility of manual task assignment. Taking preparatory work before part processing as an example. When the corresponding ACN selects

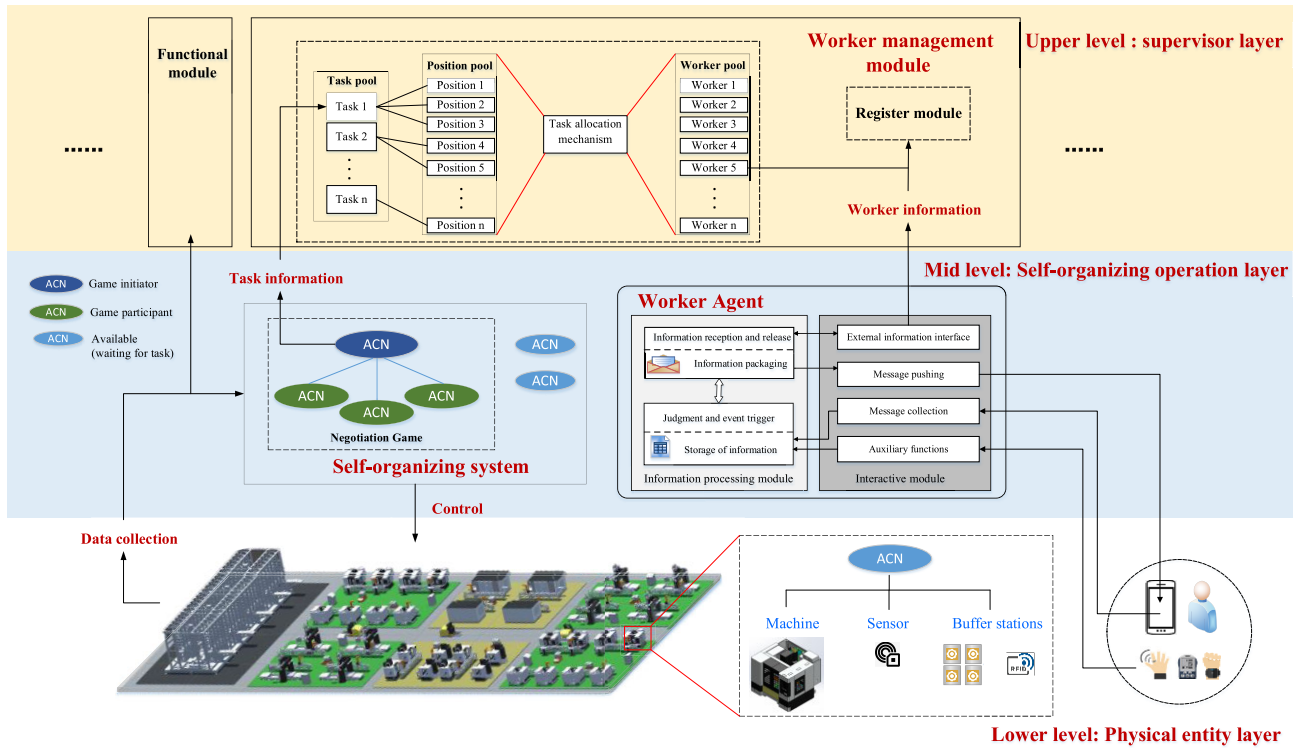


FIGURE 5. The architecture of the control system.

a certain batch of parts located in the buffer, the associated PA thread will be activated, and the processing information of the current step will be notified to the MA. MA will analyze the processing information. In addition to obtaining the NC program of the corresponding processing step, it will also release the preparatory tasks (e.g. fixture replacement, butted-knife, and confirmation of NC program, etc.) required by the current processing step to the “supervisor layer”. When a WoA is registered in the supervisor layer, the corresponding worker information is written into the relevant database. The worker status is updated to the management module in real-time during the system operation.

Task objects and worker objects will be divided into two lists, which are called “task pool” and “worker pool” in the “worker management module”. When the data in the task pool or worker pool changes, a new round of task allocation will be triggered. Since the task may need to be implemented by multiple workers, the task will be split into positions, forming a “position pool”. To further fit the actual situation, in this paper, the following two points are considered:

- 1) In actual operation, positions may have specific requirements for workers. For example, expensive CNC equipment requires the participation of high-level workers.
- 2) Different workers have different matching degrees for various positions. The matching degree of workers to other positions represents the proficiency of the

workers for the task. The more times the same type of task is assigned to a worker, the higher the ability for the worker. The WoA corresponding to the worker will record each task’s type and execution time for the worker, which calculates the matching degree of “worker-match-position”.

Considering the above situation, a dynamic worker task assignment mechanism is described as follows.

C. DYNAMIC TASK ALLOCATION MECHANISM FOR WORKERS

For the convenience of explanation, this article defines some mathematical symbols, which are explained in Table 1.

The worker task allocation process can be transformed into a bipartite graph matching problem [28], as shown in Fig. 6. $M = \{m_i | i = 0, \dots, I\}$ indicates workers’ collection. $N = \{n_j | j = 0, \dots, J\}$ indicates positions collection. Assigning workers to appropriate positions can be seen as a match between these two collections. Positions are split from the tasks, and different tasks require different numbers of positions. Only one worker can be assigned to one position, and a task can only be activated when all the positions in this task are assigned. After one task is started, those workers assigned to this task will go to the corresponding workstation according to the instructions issued by their WoAs.

Suppose the worker is qualified for a specific position. In that case, there is a connection between the two, which

TABLE 1. Mathematical symbols defined in this article.

Mathematical symbol	Description
$M = \{m_i i = 0, \dots, I\}$	The collection of workers waiting to be assigned
$N = \{n_j j = 0, \dots, J\}$	The collection of positions waiting to be assigned
$\{T_k k = 0, \dots, K\}$	The collection of tasks waiting to be assigned
a_{ij}	The connection between m_i and n_j
W_{ij}	The weight value of the connection a_{ij}
D_k	Task matching index
V_k	Task value index
λ_k	The effectiveness of the k th task
U_k	The urgency of the k th task
st_k	The estimated time when the k th task can start
et_k	The estimated difference between the finish date and the delivery date of the k th task
wt_k	The waiting time of the k th task
rt_k	The estimated remaining processing time of k th task
dt_k	The delivery date of k th task
ct_k	The current date of k th task
NDI_k	Near-delivery index (illustrate the impact of delivery date on task urgency)

is represented by a_{ij} . W_{ij} indicates the weight value of the connection between worker i and position j . This suitability means the ability of the worker to handle the corresponding task. WoAs will record the worker’s performance for each task (usually the time to complete the task), and these data will be the basis for scoring W_{ij} . The value of W_{ij} depends on these historical data and is derived from experience. It can be calculated by WoA or directly scored by the workshop director.

When task assignment is triggered, two scenarios need to be considered: scenario 1, the number of idle workers meets the position requirements; scenario 2, the number of idle workers is not enough to fill the position requirements.

1) SCENARIO 1: THE NUMBER OF IDLE WORKERS MEETS THE POSITION REQUIREMENTS

In scenario 1, the ultimate goal of the allocation is to get the right workers to the right place. That is, the final matching sum is the largest. The optimization goals are as follows:

$$\begin{aligned}
 &\max \sum_{i=0}^I \sum_{j=0}^J W_{ij} \cdot a_{ij} \\
 &s.t. \sum_{i=0}^I a_{ij} = 1, \quad j = 1, \dots, J \\
 &\quad \sum_{j=0}^J a_{ij} = 1, \quad i = 1, \dots, I
 \end{aligned} \tag{1}$$

where:

$$a_{ij} \begin{cases} 1 & \text{if position } j \text{ is assigned to worker } i \\ 0 & \text{if position } j \text{ is not assigned to worker } i \end{cases}$$

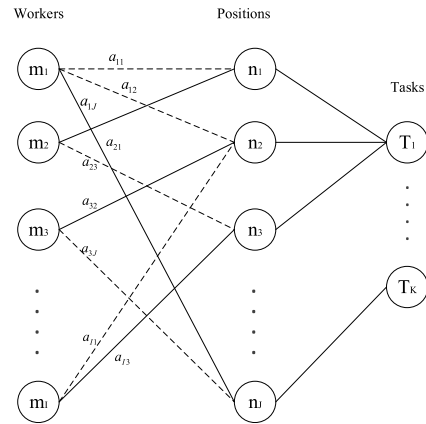


FIGURE 6. Workers-positions-tasks.

At this time, the problem is transformed into finding the optimal matching in the bipartite graph. The Kuhn-Munkres (KM) algorithm can be perfectly applied to the solution process in this case [29]. Using the KM algorithm, we can easily find the optimal solution for the allocation in this situation. After that, the supervisor layer will push the tasks to the workers. When the workers confirm, this round of allocation is completed. If some workers choose not to accept it, a new round of assignments will be performed.

2) SCENARIO 2, THE NUMBER OF IDLE WORKERS IS NOT ENOUGH TO FILL THE POSITION REQUIREMENTS

Scenario 2 mainly occurs when the workload is high. If only idle workers are considered at this time, matching according to the ability value may result in a situation where no task can be performed. If the first-come-first-served rule is adopted, it may lead to optimal local problems. It is supposed that three tasks need to be allocated urgently, and the existing idle workers can only meet the requirements of one of the tasks. After the tasks are allocated, a new batch of workers completes the current tasks and enters the worker allocation pool, but the newly arrived workers cannot match the next two tasks. And if the previous allocation can wait for a period of time, the three tasks can be allocated simultaneously. From the resulting point of view, the latter scheme is better.

This article proposes a new distribution mechanism based on wearable devices and worker autonomy to avoid the above problem. When an assignment event occurs, the supervisor layer will query the worker’s status in working state, and get the average completion time and the time the task has been in progress. When the task progress enters a specific time window, the supervisor layer sends a confirmation invitation to the workers to participate in the next step, and the workers will reply to the invitation according to the actual situation. After confirmation, the workers who are working on this task will also enter the pool of workers to be assigned.

In order to make the distribution more reasonable, this article uses the following two indexes:

The first is the “matching index”. Like scenario 1, the system always hopes that workers can be fully utilized for each task assignment. However, because workers cannot meet the needs of all tasks, some tasks cannot make up the required workers, which means that these assignments are invalid and the tasks cannot be started. The matching index of the k th task (i.e. T_k) is represented by D_k . If the task is not activated, D_k is zero; if the task is activated, D_k is the sum of the weights of the matching workers. Fig. 7 shows an example, the task T_1 is not assigned enough workers, $D_1 = 0$. The task T_2 is allocated enough workers, $D_2 = a_{13} + a_{34}$.

The second is the “value index”. When a round of allocation is completed, the system always hopes that enough tasks can be allocated. At the same time, the allocated tasks enable the system to reap the greatest value. The value index of the k th task (i.e. T_k) is represented by V_k , and the calculation method is as follows:

$$V_k = \lambda_k \cdot U_k \quad (2)$$

In equation (2), λ_k represents the effectiveness of the k th task, and its value range is 0~1. If the task can be executed immediately, the value is 1. If the task cannot collect the required workers, the value is 0. In scenario 2, there are idle workers involved in assigning tasks and workers who can complete the task immediately, and the execution of the task may not start immediately. The estimated time when the task can start (i.e. st_k) determines this value. We can directly design the value of λ_k when the st_k is in different intervals based on experience. For example: after an assignment, if the assigned workers of a task need 100s to be in place, $\lambda_k = 0.7$; if it takes 200s to be in place, $\lambda_k = 0.5$; if it is more than 200s to be in place, $\lambda_k = 0.3$.

In equation (2), U_k represents the urgency of the task. Tasks with higher urgency should have a higher value. In different working environments, there will be different definitions for U_k . Here, this article gives a more general equation to describe its specific value. The delivery date of the part involved in the k th task and the time waiting by the k th task will be determined by the delivery date of the part involved in the k th task (i.e. wt_k).

We first use a designed index, “near-delivery index (NDI)”, to illustrate the impact of delivery date on task urgency. The NDI is used to determine whether there are overdue risks and the specific value of the overdue risk in production activities.

$$NDI_k = \begin{cases} 0 & et_k \leq 0 \\ \frac{et_k}{rt_k} & et_k > 0 \end{cases} \quad (3)$$

$$et_k = dt_k - (ct_k + \alpha \cdot rt_k) \quad (4)$$

In equations (3) and (4), the actual meaning of each parameter is shown in Fig. 8. et_k represents the estimated difference between the finish date and the delivery date of the k th task.

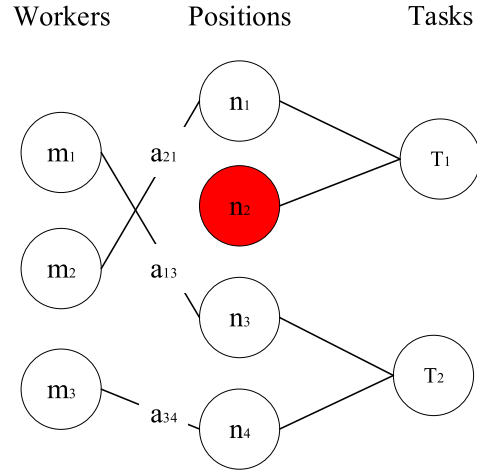


FIGURE 7. The example for “matching index” calculation.

rt_k represents the estimated remaining processing time of the k th task. dt_k represents the delivery date. ct_k represents the current date, and α is the amplification factor of the rt_k . Since the workpiece may be idle due to insufficient productivity, the predicted execution time needs to be enlarged. Usually the value of α is 1.5 to 2. It can be seen from Fig. 12 that when et_k is less than 0, the task is overdue, and $NDI_k = 0$. $\frac{et_k}{rt_k}$ can represent the urgency of the delivery date. The larger the $\frac{et_k}{rt_k}$, the smaller the risk of overdue is. This article uses the method in equation (3) to normalize it and finally obtains NDI_k .

We normalize the NDI_k and wt_k of the tasks in the task pool to obtain U_k .

$$U_k = \mu \cdot \frac{1}{NDIn_k} + \nu \cdot \frac{1}{WTn_k} \quad (5)$$

where:

$$NDIn_k = \frac{NDI_k}{NDI_{max}} \quad (6)$$

$$WTn_k = \frac{wt_{max}}{wt_k + wt_{max}} \quad (7)$$

In equation (5), μ and ν are the weights of $\frac{1}{NDIn_k}$ and $\frac{1}{WTn_k}$ respectively. They are usually valued according to the actual situation of the factory. When a round of allocation is completed, we can get the “matching index” and “task value index” corresponding to each task. The optimization goal can be obtained by multiplying these two items in each task and then summarising everything. The optimization goal for scenario 2 is described in equation (8). The larger the optimization goal, the more tasks are assigned, and it also means that the more urgent tasks are more likely to be assigned to the skilled workers.

$$\max ([D_k][V_k]^T), \quad k = 1, \dots, K \quad (8)$$

The control system always expects that worker resources can be fully utilized, that is, the final matching result should be the maximum matching of the bipartite graph. Maximum matching refers to the largest connection between workers and positions. We can find the optimal solution

of equation (8) in the maximum matching. The Hungarian method can be used to find the largest match of the bipartite graph. It is a combinatorial optimization algorithm that solves the assignment problem in polynomial time. In the case where the weight is not considered, this method can quickly help us to find the maximum match, that is, the matching connection is the most. The process of the Hungarian algorithm is carefully described in [30].

However, when the arrangement order is fixed, the Hungarian algorithm of the same model can only get one solution. This article considers disrupting the arrangement order and uses genetic algorithms to better match results [31]. A case shown in Fig. 9 is used to illustrate the execution process of the algorithm. The related procedures and data used in this case can be obtained in (<https://github.com/zhjj370/MyNewPaper2>). The CPU used in the test computer is AMD Ryzen 3700U, and the memory is 8Gb.

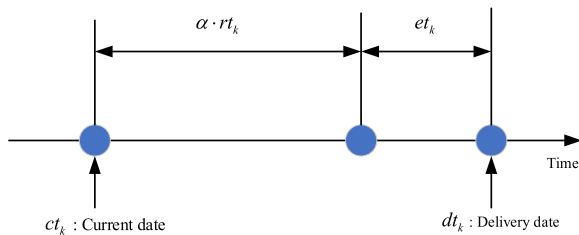


FIGURE 8. The actual meaning of each parameter when calculating the “near-delivery index.”

In the example, there are 6 tasks to be assigned, which are split into 10 positions. At this time, idle workers do not meet the job requirements. After the supervisor layer interacts with WoAs, the two groups of workers who meet the conditions are added to the worker pool to be assigned. If no task assignment event is triggered afterwards, the tasks that require the participation of workers in this part will be performed directly after the personnel is in place. If there are tasks that enter the task pool before this part of the workers are in place, the event will be redistributed.

For the convenience of explanation, the NDI and wt_k of each task are directly given in the simulation test. Tasks will be divided into positions according to requirements. The numbers on the table in Fig. 6 indicate the degree of fit between workers and positions. It can be automatically calculated by each WoA based on the historical completion of related tasks or by the workshop director based on the performance score of the workers during a specific period. The task in the factory may be set to a level, and only workers who reach the corresponding level can operate. If the level of the worker is not enough or the worker does not have the skills for this task, the fit is “0”. Our ultimate goal is to find the optimal solution or better solution that satisfies the equation (8) in the table. The solution in this article is to find the maximum match between workers and positions as much as possible and select the optimal solution in these maximum

matches. The combination of the Hungarian algorithm and genetic algorithm can help us solve this problem. The key steps are shown in Fig. 10.

–Step1. *Population initialization.* When the assignment event starts, the system will orderly assign specific ID numbers to the tasks and workers participating in this round of assignments. The chromosome in the population is composed of two parts, including the position gene and worker gene, which are the combination of the task ID and the worker ID.

–Step2. *Fitness calculation.* This article adopts equation (8) as the fitness function. According to the arrangement order of the positions and workers provided by the chromosome, the Hungarian algorithm can be used to obtain the largest match in this arrangement order. This process is equivalent to the decoding process in the genetic algorithm. We can get the fitness value of the chromosome though substituting the matching result into equation (8).

–Step3. *Selection.* This article uses the roulette method to select a new generation of chromosomes.

–Step4. *Crossover.* When the parent chromosomes are crossed, position genes and worker genes will be crossed separately and then combined into a new chromosome. The “Order Crossover” algorithm, proposed by Davis [32], is used in this step.

–Step6. *Mutation.* The chromosomes will have a certain chance of mutation, and the values of the two positions in the position gene and the worker gene will be randomly exchanged.

In this case, the population size is set to 30, and the mutation probability is 0.01, and the maximum genetic generation is 100. The fitness convergence curve and the better allocation result obtained by the algorithm are shown in Fig. 11. In this case, the number of workers and tasks is small, so the results quickly converge. When the end condition is set to 100 for genetic generation, it takes 23ms.

To confirm the workshop scale for which the algorithm can be used and further prove that the algorithm is suitable for the dynamic allocation of worker tasks, this article expands the number of workers and tasks to be allocated and conducts

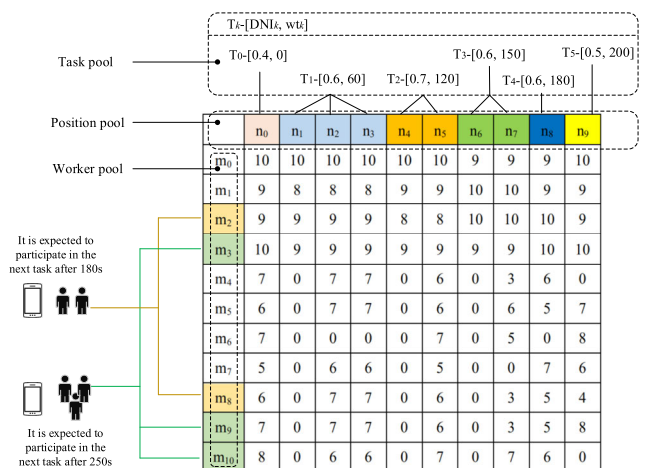


FIGURE 9. An example is used to illustrate the algorithm process.

a larger-scale algorithm experiment. Three different sizes of workers and positions were tested. In each case, 20 tests were carried out, and the final results are shown in Table 2. The convergence curve in each case is shown in Fig. 12. The results show that the allocation mechanism and its supporting algorithms designed in this article can meet the needs of processing workshops with hundreds of workers. When the size of the workshop is further expanded, the task allocation mechanism designed in this article can be used by dividing the workshop area and specifying the work area for workers.

IV. IMPLEMENTATION IN THE ACTUAL ENVIRONMENT

A processing workshop laboratory located in Nanjing University of Aeronautics and Astronautics is used to verify the worker integration method under the multi-agent system proposed in this article and determine its effectiveness in actual use. The laboratory is shown in Fig. 13.

For the case used in this article, the shop floor physical entities can be classified into multiple types of units, including an AS/RS unit, two milling units, two turning units, an AGV unit, two robot units, and a detection unit. Two types of parts are used as processing objects. In the experiment, manual tasks mainly refer to the preparatory work of these parts, including replacing fixtures and tools. These tasks usually require 2-3 people to work together.

Several students participated in this experiment. They had experienced training and were able to implement the

part-processing preparatory work skillfully. Their ability values for various tasks in the experiment had approached a stable state, reaching their ability bottleneck, so these ability values could be calibrated directly before the experiment.

The workshop layer has been deployed in multi-agent control systems based on ACN technology. The WoA program proposed in this article is embedded in the mobile phone as an application and connected to the control system. The workshop layer and the workers finally form the control architecture shown in Fig. 5. In the actual experiment, ACNs publish worker task information based on current processing requirements. WoAs receive task information and notifies workers (i.e. students) for processing. Although worker-task time has strong randomness, the WoA program assists workers to join the production process orderly through the interaction between agents. It means that the integrated approach for workers designed in this article can realize the dynamic processing of production tasks.

In the worker-task allocation method test, we used two sets of schemes, including first-come-first-served method and the method designed in section III. In this test, we let the factory maintain a full-load production state and gradually reduce the number of workers. We counted the overdue rate of processing-tasks and the vacancy rate of worker-tasks under the number of different workers (as shown in Figure 14). The overdue rate refers to the sum of the final outstanding tasks exceeding the specified time to the total working time. The

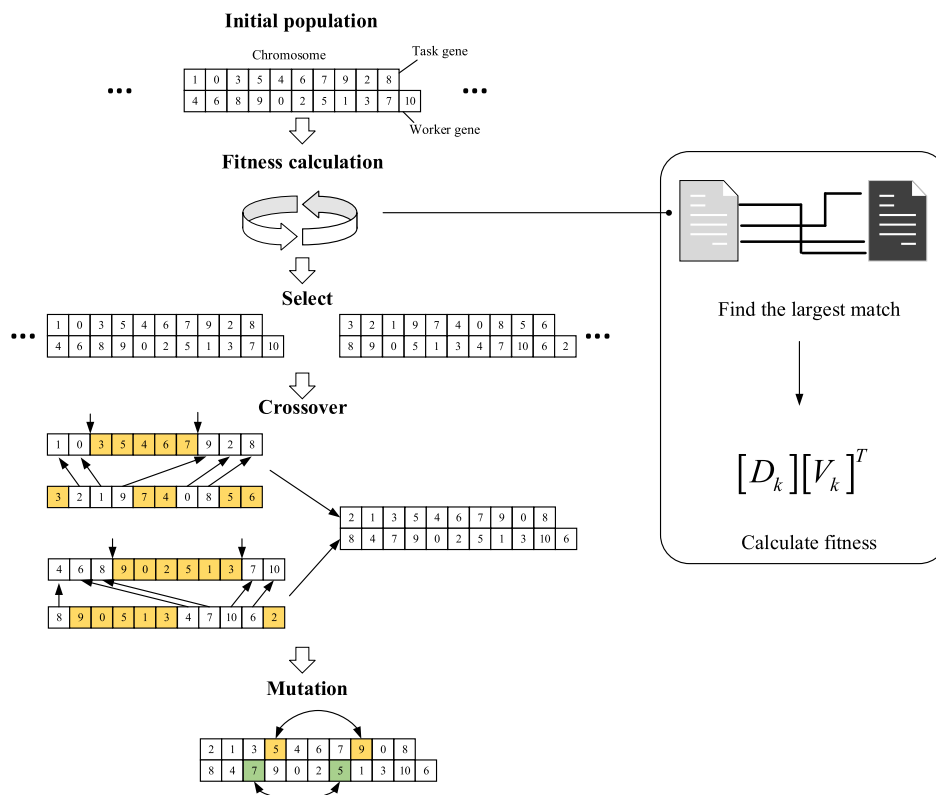
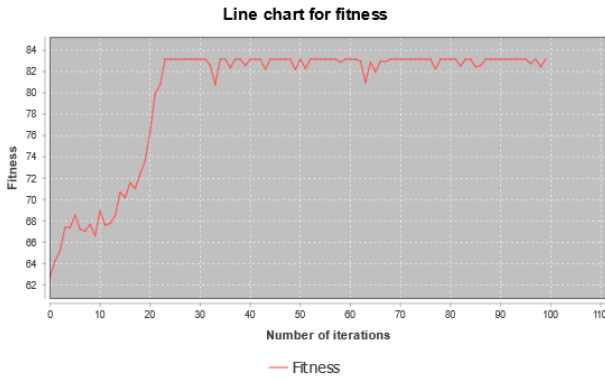


FIGURE 10. Key steps in this genetic algorithm.



(a) Convergence curve of fitness

	n_0	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9
m_0	10	10	10	10	10	10	9	9	9	10
m_1	9	8	8	8	9	9	10	10	9	9
m_2	9	9	9	9	8	8	10	10	10	9
m_3	10	9	9	9	9	9	9	9	10	10
m_4	7	0	7	7	0	6	0	3	6	0
m_5	6	0	7	7	0	6	0	6	5	7
m_6	7	0	0	0	0	7	0	5	0	8
m_7	5	0	6	6	0	5	0	0	7	6
m_8	6	0	7	7	0	6	0	3	5	4
m_9	7	0	7	7	0	6	0	3	5	8
m_{10}	8	0	6	6	0	7	0	7	6	0

(b) Convergence curve of fitness. Excellent task assignment results (The orange background in the picture represents the matching item.)

FIGURE 11. Simulation results of the case.

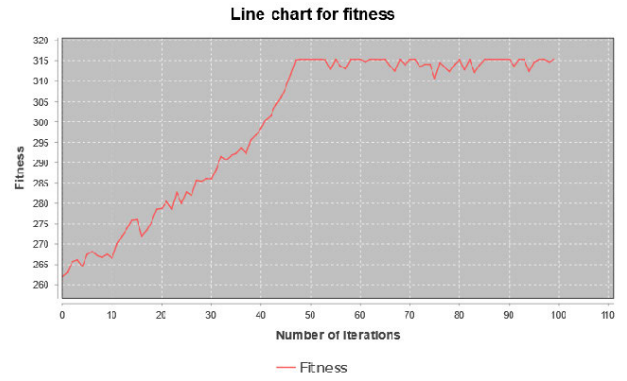
TABLE 2. Simulation result.

Worker size	Position size	Population size	Average genetic generation for fitness convergence	Average convergence time (ms)
44	40	40	40	57
110	100	100	205	1305
220	200	100	610	3905

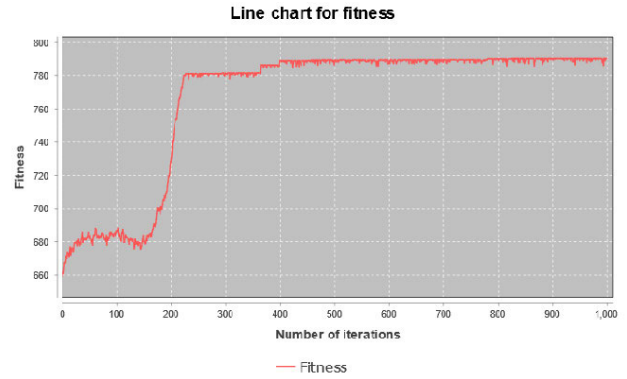
worker-task vacancy rate refers to the average proportion of the task that failed to be assigned in each round of assignments.

In this test, when the number of workers is 10, the full load production can be satisfied. At this time, the part delivery period can be guaranteed under these two allocation methods. As the workers are reduced, the part overdue rate and the worker-task vacancy rate will increase. It represents that workers become busy, and the worker-resource has become a production bottleneck.

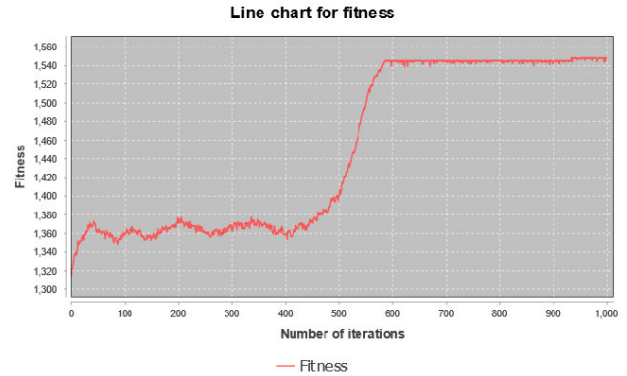
As shown in Fig. 14, the allocation method designed in section III will slow down the growth trend of the overdue rate. This is because when the worker-resource becomes a



(a) 44*40



(b) 110*100



(c) 220*200

FIGURE 12. Fitness convergence curve under different sizes of workers and positions.

bottleneck, equation (8) will make urgent tasks more easily access to workers with high capability. Simultaneously, from the change of worker-task vacancy rate, we can clearly find that the allocation method designed in this paper has a higher allocated efficiency in each round of task-allocation, because we adopt a communication appointment during the allocation.

In order to further verify the effectiveness, we have a long-term performance observation. This article carried out the following experiments. We keep the number of workers at 7.

–Case 1: The system published the tasks that workers need to perform on the electronic screen, and a teacher played as a workshop director to assign these tasks.

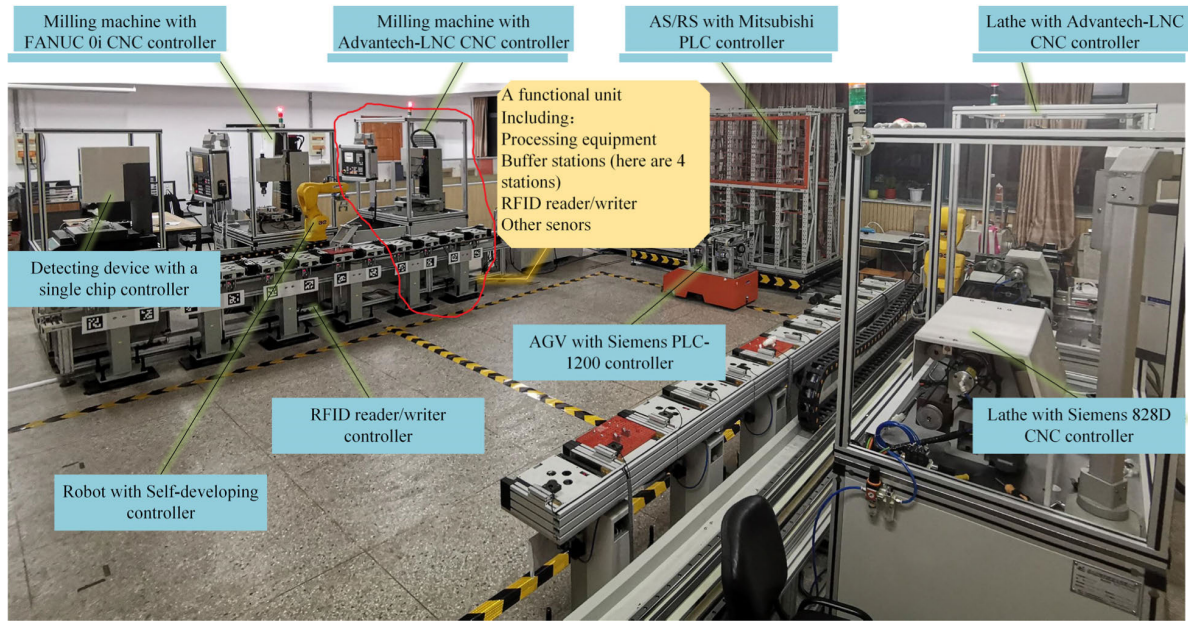
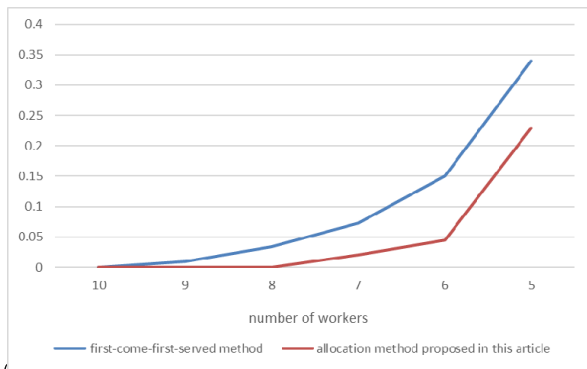
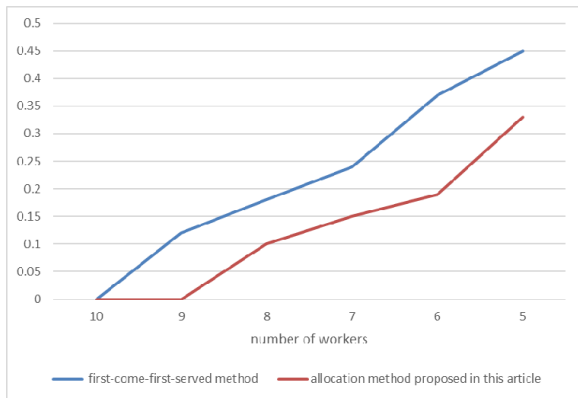


FIGURE 13. A processing workshop laboratory.



(a) Overdue rate of processing tasks



(b) Vacancy rate of worker-tasks

FIGURE 14. The performance of two allocation methods with the reduction of workers. The two allocation methods are the first-come-first-served method and the method proposed in section III. The overdue rate refers to the sum of the final outstanding tasks exceeding the specified time to the total working time. The worker-task vacancy rate refers to the average proportion of the task that failed to be assigned in each round of assignments.

–Case 2: Workers obtained the tasks through the WoA app, and the allocation of the supervisor layer follows the principle of first-come-first-served.

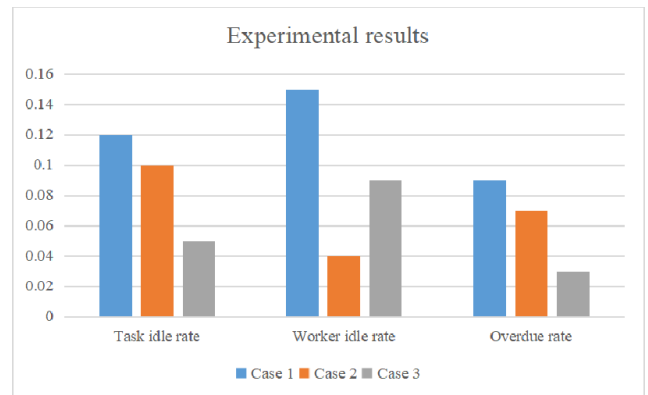


FIGURE 15. Experimental result.

–Case 3: Workers obtained the tasks through the WoA app, and the allocation method followed the mechanism designed in this article.

The experiment compares the “Task idle rate”, “Worker idle rate”, and “Overtime rate” under the three cases. Task idle rate refers to the ratio of the total idle time of the parts due to worker-tasks cannot be processed in time during processing to the total working time. The idle rate of workers refers to the ratio of the total idle time of workers due to the lack of partners to the total working time. Several experiments have been implemented. The results after each experiment are recorded, and the average values are shown in Fig 15.

From the results of “Task idle rate”, after using WoA and wearable devices to integrate workers, the waiting time of the workpiece due to the lack of personnel is significantly reduced. This means that the overall efficiency of the system is improved. Simultaneously, judging from the results, following the worker task dynamic allocation mechanism designed in section III is better than the first-come-first-served principle.

Comparing the “Worker idle rate” results, following the first-come-first-served principle, workers spend less time idle, while manual allocation makes workers idle more time. After following the dynamic mechanism designed in this article, workers’ idle time is middle. Combining the two indicators of “Task idle rate” and “Overdue rate”, the result of case 3 is better than the result of case 2. This shows that the use of workers by the system in Case 3 is at a reasonable level - workers are less busy, but the system is more efficient.

Therefore, it can be seen that the WoA integration scheme proposed in this article can significantly improve the work efficiency of workers and reduce the risk of task overdue due to personnel factors.

V. CONCLUSION

This article discusses how to integrate workers into the overall control system under MAMS and how to improve worker efficiency after integration. Wearable mobile devices provide hardware support for WoA programs.

The multi-skilled feature of workers and the collaboration between workers are fully considered in the integration process. In the current manufacturing environment, workers are not fixed in a position, and they can temporarily form a team to complete a task according to system requirements. The entire control system follows the structure of HyMCA. We have designed a special information interface to assist WoA integration. As a black box agent, the WoA participates in the operation of the control system and assists the corresponding worker in undertaking their tasks.

The interaction mechanism of WoA with other types of agents and modules is also studied in this article. A dynamic task allocation mechanism is investigated. Considering that a task may require multiple workers to complete, this article splits the task into multiple positions that require workers to match. We abstract the allocation problem into a bipartite graph matching problem and consider the algorithms and strategies used in different scenarios. A new task allocation algorithm is designed for complicated situations. In this algorithm, the guarantee of delivery time and the optimization of workers’ ability are fully considered. The effectiveness and timeliness of this algorithm under different scale problems are verified. It should be noted that we use the genetic algorithm in the allocation method, but the genetic algorithm is not the focus of this article, so it is not optimized. In actual use, there will be premature convergence. The method has a further optimized space. In future works, we will continue to optimize it. At the same time, how to make workers cooperate efficiently is the next problem we need to solve.

REFERENCES

- [1] N. K. C. Krothapalli and A. V. Deshmukh, “Design of negotiation protocols for multi-agent manufacturing systems,” *Int. J. Prod. Res.*, vol. 37, no. 7, pp. 1601–1624, May 1999, doi: [10.1080/002075499191157](https://doi.org/10.1080/002075499191157).
- [2] Y. G. Kim, S. Lee, J. Son, H. Bae, and B. D. Chung, “Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system,” *J. Manuf. Syst.*, vol. 57, pp. 440–450, Oct. 2020, doi: [10.1016/j.jmsy.2020.11.004](https://doi.org/10.1016/j.jmsy.2020.11.004).
- [3] D. Trentesaux and P. Millot, “A human-centred design to break the myth of the ‘magic human’ in intelligent manufacturing systems,” in *Service Orientation in Holonic and Multi-Agent Manufacturing*, T. Borangiu, D. Trentesaux, A. Thomas, and D. McFarlane, Eds. Cham, Switzerland: Springer, 2016, pp. 103–113.
- [4] F. Longo, L. Nicoletti, and A. Padovano, “Smart operators in industry 4.0: A human-centered approach to enhance operators’ capabilities and competencies within the new smart factory context,” *Comput. Ind. Eng.*, vol. 113, pp. 144–159, Nov. 2017, doi: [10.1016/j.cie.2017.09.016](https://doi.org/10.1016/j.cie.2017.09.016).
- [5] Z. Kemény, J. Váncza, L. Wang, and X. V. Wang, “Human-robot collaboration in manufacturing: A multi-agent view,” in *Advanced Human-Robot Collaboration in Manufacturing*, L. Wang, X. V. Wang, J. Váncza, and Z. Kemény, Eds. Cham, Switzerland: Springer, 2021, pp. 3–41.
- [6] J. Leuvennink, K. Kruger, and A. Basson, “Architectures for human worker integration in holonic manufacturing systems,” in *Service Orientation in Holonic and Multi-Agent Manufacturing: Proceedings of SOHOMA 2018*, T. Borangiu, D. Trentesaux, A. Thomas, and S. Cavalieri, Eds. Cham, Switzerland: Springer, 2019, pp. 133–144.
- [7] M. P. Pacaux-Lemoine, D. Trentesaux, G. Zambrano Rey, and P. Millot, “Designing intelligent manufacturing systems through human-machine cooperation principles: A human-centered approach,” *Comput. Ind. Eng.*, vol. 111, pp. 581–595, Sep. 2017, doi: [10.1016/j.cie.2017.05.014](https://doi.org/10.1016/j.cie.2017.05.014).
- [8] L. Monostori, J. Váncza, and S. R. T. Kumara, “Agent-based systems for manufacturing,” *CIRP Ann. Technol.*, vol. 55, no. 2, pp. 697–720, 2006, doi: [10.1016/j.cirp.2006.10.004](https://doi.org/10.1016/j.cirp.2006.10.004).
- [9] L. Wang and A. Haghighi, “Combined strength of holons, agents and function blocks in cyber-physical systems,” *J. Manuf. Syst.*, vol. 40, pp. 25–34, Jul. 2016, doi: [10.1016/j.jmsy.2016.05.002](https://doi.org/10.1016/j.jmsy.2016.05.002).
- [10] S. Kraus, “Negotiation and cooperation in multi-agent environments,” *Artif. Intell.*, vol. 94, nos. 1–2, pp. 79–97, 1997, doi: [10.1016/S0004-3702\(97\)00025-8](https://doi.org/10.1016/S0004-3702(97)00025-8).
- [11] H. Lan, R. Liu, and C. Zhang, “A multi-agent-based intelligent STEP-NC controller for CNC machine tools,” *Int. J. Prod. Res.*, vol. 46, no. 14, pp. 3887–3907, Jul. 2008, doi: [10.1080/00207540701213494](https://doi.org/10.1080/00207540701213494).
- [12] S. Bussmann, “An agent-oriented architecture for holonic manufacturing control,” in *Proc. 1st Int. Workshop IMS*, Lausanne, Switzerland, 1998, pp. 1–12. [Online]. Available: <http://stefan-bussmann.de/downloads/ims98.pdf>
- [13] P. Stone and M. Veloso, “Multiagent systems: A survey from a machine learning perspective,” *Auto. Robots*, vol. 8, no. 3, pp. 345–383, Jun. 2000, doi: [10.1023/A:1008942012299](https://doi.org/10.1023/A:1008942012299).
- [14] Z. Jiang, Y. Jin, E. Mingcheng, and Q. Li, “Distributed dynamic scheduling for cyber-physical production systems based on a multi-agent system,” *IEEE Access*, vol. 6, pp. 1855–1869, 2018, doi: [10.1109/ACCESS.2017.2780321](https://doi.org/10.1109/ACCESS.2017.2780321).
- [15] Y. Sallel, T. Berger, S. Raileanu, S. Chaabane, and D. Trentesaux, “Semi-hierarchical control of FMS: From theory to application,” *Eng. Appl. Artif. Intell.*, vol. 23, no. 8, pp. 1314–1326, Dec. 2010, doi: [10.1016/j.engappai.2010.06.013](https://doi.org/10.1016/j.engappai.2010.06.013).
- [16] L.-C. Wang, C.-Y. Cheng, and S.-K. Lin, “Distributed feedback control algorithm in an auction-based manufacturing planning and control system,” *Int. J. Prod. Res.*, vol. 51, no. 9, pp. 2667–2679, May 2013, doi: [10.1080/00207543.2012.738944](https://doi.org/10.1080/00207543.2012.738944).
- [17] M. F. Zaeh, G. Reinhart, M. Ostgathe, F. Geiger, and C. Lau, “A holistic approach for the cognitive control of production systems,” *Adv. Eng. Informat.*, vol. 24, no. 3, pp. 300–307, Aug. 2010, doi: [10.1016/j.aei.2010.05.014](https://doi.org/10.1016/j.aei.2010.05.014).
- [18] N. A. Duffie, “Synthesis of heterarchical manufacturing systems,” *Comput. Ind.*, vol. 14, nos. 1–3, pp. 167–174, May 1990.
- [19] W. Shen, L. Wang, and Q. Hao, “Agent-based distributed manufacturing process planning and scheduling: A state-of-the-art survey,” *IEEE Trans. Syst., Man, Cybern., C (Appl. Rev.)*, vol. 36, no. 4, pp. 563–577, Jul. 2006, doi: [10.1109/TSMCC.2006.874022](https://doi.org/10.1109/TSMCC.2006.874022).
- [20] P. Leitão, “Agent-based distributed manufacturing control: A state-of-the-art survey,” *Eng. Appl. Artif. Intell.*, vol. 22, no. 7, pp. 979–991, 2009, doi: [10.1016/j.engappai.2008.09.005](https://doi.org/10.1016/j.engappai.2008.09.005).
- [21] C. Pach, T. Berger, T. Bonte, and D. Trentesaux, “ORCA-FMS: A dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling,” *Comput. Ind.*, vol. 65, no. 4, pp. 706–720, May 2014, doi: [10.1016/j.compind.2014.02.005](https://doi.org/10.1016/j.compind.2014.02.005).
- [22] P. Leitão and F. Restivo, “A holonic approach to dynamic manufacturing scheduling,” *Robot. Comput.-Integr. Manuf.*, vol. 24, no. 5, pp. 625–634, Oct. 2008, doi: [10.1016/j.rcim.2007.09.005](https://doi.org/10.1016/j.rcim.2007.09.005).

- [23] H. Najjari, M. Seitz, E. Trunzer, and B. Vogel-Heuser, "Cyber-physical production systems for SMEs—A generic multi agent based architecture and case study," in *Proc. 4th IEEE Int. Conf. Ind. Cyber-Physical Syst. (ICPS)*, May 2021, pp. 625–630, doi: [10.1109/ICPS49255.2021.9468232](https://doi.org/10.1109/ICPS49255.2021.9468232).
- [24] M. Peruzzini, M. Pellicciari, and M. Gadaleta, "A comparative study on computer-integrated set-ups to design human-centred manufacturing systems," *Robot. Comput.-Integr. Manuf.*, vol. 55, pp. 265–278, Feb. 2019, doi: [10.1016/j.rcim.2018.03.009](https://doi.org/10.1016/j.rcim.2018.03.009).
- [25] R. Riedel, E. Mueller, R. von der Weth, and N. Pflugradt, "Integrating human behaviour into factory simulation- a feasibility study," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage.*, Dec. 2009, pp. 2089–2093, doi: [10.1109/IEEM.2009.5373164](https://doi.org/10.1109/IEEM.2009.5373164).
- [26] F. Bellifemine, F. Bergenti, G. Caire, and A. Poggi, "JADE—A Java agent development framework," in *Multi-Agent Programming: Languages, Platforms and Applications*, R. H. Bordini, M. Dastani, J. Dix, A. El Fallah Seghrouchni, Eds. Boston, MA, USA: Springer, 2005, pp. 125–147.
- [27] Smith, "The contract net protocol: high-level communication and control in a distributed problem solver," *IEEE Trans. Comput.*, vol. C-29, no. 12, pp. 1104–1113, Dec. 1980, doi: [10.1109/TC.1980.1675516](https://doi.org/10.1109/TC.1980.1675516).
- [28] F. Glover, "Maximum matching in a convex bipartite graph," *Nav. Res. Logistics Quart.*, vol. 14, no. 3, pp. 313–316, 1967, doi: [10.1002/nav.3800140304](https://doi.org/10.1002/nav.3800140304).
- [29] H. Zhu, M. Zhou, and R. Alkins, "Group role assignment via a Kuhn–Munkres algorithm-based solution," *IEEE Trans. Syst., Man, Cybern., A, Syst. Humans*, vol. 42, no. 3, pp. 739–750, May 2012, doi: [10.1109/TSMCA.2011.2170414](https://doi.org/10.1109/TSMCA.2011.2170414).
- [30] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955, doi: [10.1002/nav.3800020109](https://doi.org/10.1002/nav.3800020109).
- [31] M. Gen and L. Lin, "Genetic algorithms," in *Wiley Encyclopedia of Computer Science and Engineering*. Atlanta, GA, USA: American Cancer Society, 2008, pp. 1–15.
- [32] L. Davis, *Handbook of Genetic Algorithms*. New York, NY, USA: Van Nostrand, 1991.



DUNBING TANG was born in Hunan, China, in 1972. He received the Ph.D. degree from Nanjing University of Science and Technology (NUST), in 2000. Then, he spent two years on his postdoctoral research with Tsinghua University, Beijing, and the City University of Hong Kong. He was supported by the renowned Alexander von Humboldt Foundation. He has conducted his research with RWTH Aachen University, Germany, as Alexander von Humboldt Research Scientist, from July 2002 to February 2004. Thereafter, he moved to Cranfield University, U.K., as a Research Fellow. He was offered a full-time professorship. He joined Nanjing University of Aeronautics and Astronautics, in December 2005. He has published over 100 academic articles. His research interests include intelligent manufacturing systems and automation, and manufacturing system modeling. He has conducted several research grants as a Principal Investigator (PI) or the Co-PI. His research outcome can be found in international high-quality academic journals, such as the *International Journal of Production Research*, the *International Journal of Computer Integrated Manufacturing*, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, Computers & Industrial Engineering*, *The International Journal of Advanced Manufacturing Technology*, *Computers in Industry*, the *Journal of Intelligent Manufacturing*, *Robotics & Computer Integrated Manufacturing*, and *Concurrent Engineering Research & Applications*.



ZEQUN ZHANG (Graduate Student Member, IEEE) was born in Anqing, Anhui, China, in 1991. He received the B.S. degree in mechanical engineering from Nanjing University of Science and Technology, Nanjing, China, in 2013, and the M.S. degree in mechanical engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, in 2016, where he is currently pursuing the Ph.D. degree in mechanical engineering.

His research interests include framework and core models of future intelligent industrial systems.



QINGWEI NIE was born in Yangzhou, Jiangsu, China, in 1993. He received the B.S. degree in mechanical engineering from Nanjing Institute of Technology, Nanjing, China, in 2015, and the M.S. degree in mechanical engineering from Yangzhou University, Yangzhou, in 2018. He is currently pursuing the Ph.D. degree with Nanjing University of Aeronautics and Astronautics.

His research interests include framework and core models of future intelligent industrial systems.

• • •