Bi-Level Dynamic Scheduling Architecture Based on Service Unit Digital Twin Agents

Author: Jian Zhang[1], Tingming Deng[1], Haifan Jiang[1], Haojie Chen[1], Shengfeng Qin[2], Guofu Ding[1]*

[1]Institute of Advanced Design and Manufacturing, Southwest Jiaotong University, Chengdu 610031, People's Republic of China

[2]School of Design, Northumbria University, Newcastle upon Tyne NE1 8ST, UK

Corresponding author: Guofu Ding

E-mail address: dingguofu@swjtu.edu.cn

Abstract

Pure reactive scheduling is one of the core technologies to solve the complex dynamic disturbance factors in real-time. The emergence of CPS, digital twin, cloud computing, big data and other new technologies based on the industrial Internet enables information acquisition and pure reactive scheduling more practical to some extent. However, how to build a new architecture to solve the problems which traditional dynamic scheduling methods cannot solve becomes a new research challenge. Therefore, this paper designs a new bi-level distributed dynamic workshop scheduling architecture, which is based on the workshop digital twin scheduling agent and multiple service unit digital twin scheduling agents.

Within this architecture, scheduling a physical workshop is decomposed to the whole workshop scheduling in the first level and its service unit scheduling in the second level. On the first level, the whole workshop scheduling is executed by its virtual workshop coordination (scheduling) agent embedded with the workshop digital twin consisting of multi-service unit digital twins. On the second level, each service unit scheduling coordinated by the first level scheduling is executed in a distributed way by the corresponding service unit scheduling agent associated with its service unit digital twin. The benefits of the new architecture include (1) if a dynamic scheduling only requires a single service unit scheduling, it will then be performed in the corresponding service unit scheduling without involving other service units, which will make the scheduling locally, simply and robustly. (2) when a dynamic scheduling requires changes in multiple service units in a coordinated way, the first level scheduling will be executed and then coordinate the second level service unit scheduling accordingly. This divide-and-then-conquer strategy will make the scheduling easier and practical.

The proposed architecture has been tested to illustrate its feasibility and practicality.

Keywords:

Bi-level dynamic scheduling architecture; Service unit; Digital twin agent, Pure reactive scheduling, Divide-and-then-conquer strategy

1 Introduction

The traditional static scheduling cannot meet the production requirements generally due to complex production situations and dynamics. Therefore, dynamic scheduling becomes a hot research topic. The modelling and utilizing real-time status information of job shop facilities and resources have always been puzzling the dynamic scheduling in production. It is because: 1) obtaining the best scheduling scheme, cyber space and physical space should be integrated and synchronized in real time, and the mapping between the cyber space and physical space should be highly consistent. However, at present, basic process data are usually inaccurate, incomplete, asymmetric and non-real time; 2) the production process is often interfered by strong real-time related information such as external order insertion, task change, scheduling adjustment and equipment failure. Obviously, it is a great challenge to organize and utilize all available information to form an ordered and agile production.

Similar to this dynamic job shop scheduling problem, large-scale optimizations [1] share very much similar challenges. For reducing the computational burden and traffic under the centralized framework, existing large-scale optimizations often decompose a complex system into some subsystems and adopt a distributed control strategy. It is observed that only one or a few subsystems are needed at each time to solve the whole system optimization problem [2]. In dealing with large-scale scheduling problems, the existing research mainly focuses on the distributed implementation of the scheduling algorithms [3][4]. However, it is also important to build an overall architecture of a dynamic scheduling system with reasonable decomposition and rapid reconfiguration of sub-scheduling problems. Under this architecture, how to obtain and utilize real-time information of the production resources and their states in the physical space and feed them back into the virtual scheduling space is a prerequisite for ideal real-time dynamic scheduling. This requirement is not easily met with the traditional manufacturing technology, but it could be enabled by the emerging digital twin technology.

Digital twin provides an enabling infrastructure to solve those problems because it makes the operation and information mapping between the physical system and its virtual system (twin) synchronized, bidirectional and in real-time. However, taking a workshop as an example, during the dynamic mapping of the physical workshop and its virtual twin, huge data and information are generated, which puts forward another new challenge to the traditional scheduling architecture. It is difficult to establish a mathematical model that can be used for real-time and effective calculation or approximation with extra complex information. It is also difficult for many intelligent algorithms dealing with dynamic scheduling of a flexible job shop due to the low calculation efficiency or accuracy in a short time, which often results in improper scheduling and affects the production efficiency and economic benefits.

Nevertheless, a workshop digital twin can be decomposed into different service unit digital twins, and each service unit can be constructed with seven basic digital twin building elements [5]. This new workshop digital twin construction method based on our team's up-front research works provides a foundation for a new bi-level dynamic scheduling scheme.

This paper proposes a new bi-level and distributed dynamic scheduling architecture (See Fig. 1). The dynamic scheduling can be systematically structured and associated with the digital twin model of a production system such as a workshop. For example, a workshop can be regarded as a configuration of several service units with proper resource allocations [5], its digital twin model can be described by the workshop configuration model at the first level and associated service unit models at the second level. On each level, a digital twin model can combine a level-specific dynamic scheduling agent to form a digital twin scheduling agent. For simplicity, we call it a DTSA hereafter. In this way, scheduling results generated by a virtual workshop scheduling agent can control the physical workshop. Therefore, the real-time optimal allocation of physical workshop resources could be realized, the cyber physical fusion for controlling the real space by virtual space could be completed, and closed loop of production could be formed.

With this architecture, a physical workshop digital twin scheduling agent (WDTSA) at the first level will be responsible for optimal resource coordination and scheduling for each service unit while a service unit digital twin scheduling agent (SUDTSA) at the second level is responsible for the job scheduling within the unit. A service unit of the workshop is originally encapsulated with seven digital twin elements [5] namely the controller, processor, executor, buffer, material, logistics path and a virtual operation environment. To make it a DTSA, a unit digital twin job scheduling agent will be associated with it, thus a DTSA can communicate with the physical service unit through the supervisory control and data acquisition (SCADA) system and real-time manufacturing execution system (MES). Some of the digital twin elements of a service unit are smart elements in digital twin space, such as machines and caches, and we also call them digital twin agents (DTA). They feedback status information and perform assigned tasks intelligently. However, they are not digital twin scheduling agent (DTSA) without scheduling decision function. On the one hand, a SUDTSA can obtain physical real-time status information. On the other hand, it can control the physical service unit. The optimal scheduling results from a SUDTSA are transmitted to its corresponding physical space by twining mechanism to control the production activities of the service unit.

The contributions of this paper are threefold: 1) a new dynamic scheduling architecture is proposed, in which the digital twin scheduling agents can communicate with their physical spaces in two directions to realize real-time scheduling; 2) a distributed multiple digital twin scheduling agents-based system, model, implementation steps and techniques are proposed, which decompose a complex scheduling problem level by level, making the scheduling distributed, flexible and in parallel, 3) by blending the upper-level heuristic rule scheduling algorithm with the lower-level humoral immune negotiation scheduling mechanism; and combining the scheduling algorithm and simulation evaluation, a better scheduling scheme can be obtained, and then a way of implementing the proposed architecture and its application in scheduling is given with an example. It illustrates this architecture not only can quickly obtain a feasible scheduling scheme, but also schedule the internal resources of relevant service units under various dynamic factors to ensure the stability of the whole manufacturing system.

On this scheduling architecture, production tasks, the production resources and service units are first mapped by WDTSA according to the process information, bill of materials (BOM), resource information, etc. Then, the tasks are allocated to each service unit according to the delivery period, standard working hours, current production status and other constraints. At a SUDTSA level, tasks are assigned in a certain order to each machine through a multi-agent scheduling negotiation mechanism. In this way, a centralized workshop scheduling problem becomes a bi-level distributed scheduling, which reduces information processing greatly and makes the real-time dynamic reactive scheduling possible.

The remainder of this paper is organized as follows. Following the related work in Section 2, the dynamic scheduling problem definition and the overall solution framework for the bi-level distributed scheduling are built in Section 3. Then the bi-level dynamic scheduling realization steps and technologies are detailed in Section 4 and illustrated by an example in Section 5, then followed by the conclusion.

2 Related work

2.1 Digital twin and its application in manufacturing

Digital twin can integrate the new generation of information technologies, such as Internet of things, mobile communication, big data, cloud computing, and manufacturing technology to build a virtual model, which is highly consistent with the manufacturing physical system with real time two-way communication, so as to realize the control of the physical system through the operation of the virtual model and provide a specific solution and feasible way for the integration of the physical world with the cyber world. This concept was first proposed by Grieves at the University of Michigan in 2003 and was mainly applied in the field of military industry and aerospace in the early stage. The primary task of digital twin application is to create the digital twin model of an applied object[6]. At present, the digital twin model mostly uses the three-dimensional model originally defined by Professor Grieves[7], that is, physical entity, virtual entity and the connection between them. Developing and innovating the existing modeling theory, it is regarded as a new direction of modeling, simulation and optimization technology[8]. Its potential values and broad application prospects have attracted many attentions.

Firstly, some scholars study and develop the concept and definition of digital twin. Negri et al [9] summarized 16 definitions of digital twin. Kritzinger et al [10] reviewed and classified the research of digital twin in the manufacturing field. In these research literatures, more than half of them discussed the concept and definition, and mainly focused on the digital model and digital shadow, while the research of digital twin in the strict sense was relatively less. Zheng [11] expounded the concept of digital twin from two perspectives of broad and narrow senses. In a narrow sense, digital twin refers to a high fidelity model or digital equivalent of physical products that integrates product life cycle data. In a broad sense, digital twin is a kind of cyber physical system (CPS). Qi and Tao compared and analyzed the similarities, differences, and relations between big data [12], CPS [13] and digital twin.

Secondly, due to the characteristics of digital twin as summarized by Boschert et al. [14], many scholars have explored the possible applications of digital twin in smart manufacturing system [15] and smart workshop [16]. Tao et al. [17] summarizes the applications of digital twin in industry, covering the whole life cycle of products, and including product/production line design, production process and product/system operation and maintenance. Liu et al. [18] discusses the rapid personalized design and implementation of flow shop driven by digital twin. Tao et al. [15] put forward the "DTS" mode and discussed the operation mechanism and key technology of digital twin shop floor in the production process. Zhuang et al. [19] proposed a digital twin based smart production management and control framework for a complex product assembly shop floor. Sun et al. [20] discussed an assembly and commissioning method of high precision products driven by digital twin. Based on the digital twin technology, Liu et al. **Error! Reference source not found.** put forward a quad-play CMCO (Configuration design-Motion planning-Control development-Optimization decoupling) design architecture for the design of a flow-type smart manufacturing system in the Industry 4.0 context.

2.2 Dynamic scheduling

At present, most manufacturing enterprises generate an overall production plan according to production capacity through their enterprise resource planning (ERP) system, decompose it into detailed plan according to process information in product data management (PDM) system, and optimize resource allocation and determine the processing sequence according to the earliest completion, delivery time, standard working hours, etc. Once the MES system is used to dispatch the tasks to the stations, various disturbing factors will inevitably appear in the process of task execution, such as inserting orders[22], new job arrival[23] machine breakdown[24], task delay[25], etc. Therefore, dynamic scheduling is proposed to replace static scheduling. At present, in the process of modeling and solving dynamic scheduling problems, robust scheduling, pre-reaction scheduling (rescheduling) and pure reactive scheduling are mainly used.

Robust scheduling is a kind of dynamic scheduling method which takes some emergencies into account in advance. It generates scheduling schemes based on the current known information or the predictable future information. Scholars mostly focus on how to improve the robustness of scheduling schemes [26]. That is to assure that in the process of real production scheduling, if there is a dynamic event, the generated scheduling scheme will not be deteriorated in a wide range or cannot be implemented. Since many preventive measures are taken into robust scheduling, it has a certain degree of robustness for dynamic events that could be occurred [27]. However, robust scheduling with some predicted events results in the processing equipment with more idle time, which is a great waste of workshop resources [28]. In addition, because there is a great deviation between the occurrence of dynamic events and the predicted events, the performance of robust scheduling may be reduced greatly [29].

Pre-reaction scheduling is also called rescheduling. It is the process of rescheduling jobs when new disturbances are generated. The main research issues are when to start rescheduling and how to reschedule. Church et al. [30] proposed three ways to start rescheduling: event driven, cycle driven, and event and cycle hybrid driven. Adil et al. [31] presents a constructive algorithm, greedy randomized adaptive search procedure (GRASP), which can solve FJSP and DFJSP with machine capacity constraints and sequence-dependent setup times. Moreover, many scholars have done a lot of work on the frequency of rescheduling [32], which shows that it has a great impact on the system performance. In addition, in order to reduce the computational complexity of rescheduling, researchers have proposed rolling window technology [33][34], which can transform a dynamic scheduling problem into a sub-problem of static scheduling in each window. For static scheduling, there are many mature methods that can be applied directly. But, how to determine the size of rolling window is a very difficult problem, and it has a great impact on the quality and efficiency of scheduling [35]. In the actual production, the window is usually set to a shift, a day or a cycle and other fixed time periods.

Pure reactive scheduling is different from the above two scheduling methods. It selects an optimized local strategy by considering the current state and some local information that can be referenced when a scheduling is needed. It can respond to dynamic events quickly and economically without producing a pre-scheduling scheme. Pure reactive scheduling is also called online scheduling by some researchers [36].

Priority scheduling is a kind of pure reactive scheduling, which can get a scheduling scheme according to the local information of a system. Generally speaking, priority scheduling technology is a method to deal with scheduling by the highest priority in the system according to a certain strategy. When there are multiple optional elements waiting to be arranged in the scheduling system, selected jobs need to be screened, evaluated and sorted. Due to a small amount of calculation, this method has the ability of fast response to dynamic events. In addition, this method neither needs to modify the original scheduling, nor causes the increase of production instability. At present, many researchers focus on scheduling rules [37] and heuristic algorithms [38][39], which are the key factors to determine the scheduling performance. But this method suffers from that rules cannot evolve by themselves when states change. In order to support dynamic scheduling, a better pure reactive scheduling is needed.

Multi-agent technology **Error! Reference source not found.**[41] is one of the most representative methods in pure reactive scheduling. Researchers have proposed an agent-based scheduling system [42][43][44]. Each agent tries to maximize its own interest, thus causing conflicts with other different agents. Therefore, a coordination agent is needed to solve the global problem. Agent based scheduling system has good adaptability and flexibility, especially in the open and dynamic production environment, so it is very suitable for the frequently changing workshop manufacturing environment. The difficulty of this technology is to coordinate the contradictions among agents, which can maximize the interest of each agent and balance the contradictions among them, so as to maximize the overall interest of the system [45]. The popular negotiation mechanisms include humoral immunity [46], CNP [47], game theory [48], and so on. If a centralized negotiation algorithm is used, its calculation cost will be very high. In addition, it requires a lot of knowledge, rich scheduling experience [46], or the ability of self-learning and self-adaptation.

2.3 Multi-agent scheduling based on digital twin

Since a manufacturing execution system is a dynamic process that comprises of production plans, machines and resources, it is the twin of physical and virtual spaces. The great significance of the scheduling strategy is the rapid response to both orders and shop floor processing variations. In recent years, many scholars have begun to solve the above dynamic scheduling problems with digital twin related methods. Cupek et al [50] introduced a multi-agent system based manufacturing execution system architecture for small batch production support, including system model, application communication mechanism and agent instance. Guizzi et al [51] introduced a new manufacturing scheduling architecture leading to a mixture of proactive and reactive approaches to the Job-shop Scheduling Problem (JSP), taking advantages of both the decentralised and centralised methodologies. Du et al. [52] presented the development of reactive and proactive scheduling methods on both order-oriented level and processing oriented level. Further, an integrated scheduling strategy is devised based on them. Zhang et al[53] developed a DT-enhanced dynamic scheduling methodology with the DT-based machine breakdown prediction, disturbance detection and performance evaluation methods.

Zhang et al. [54] reviewed different kinds of scheduling methods since the last century and put forward a method to solve a dynamic real-time scheduling problem based on distributed multi-agent scheduling by using CPS, multi-agent technology, RFID technology, big data technology and other enabling technologies under industry 4.0. Zhou [55] proposed a general framework for knowledge-driven digital twin manufacturing cell (KDTMC) towards intelligent manufacturing, which could support autonomous manufacturing by an intelligent perceiving, simulating, understanding, predicting, optimizing and controlling strategy. Fang[56] put forward a digital-twin based job shop scheduling towards smart manufacturing, and applied digital twin to a scheduling problem the first time. Compared with the traditional job shop scheduling methods, the digital-twin-based scheduling method can well adapt to the physical workshop and sufficiently meet the workshop production needs in terms of real-time response, robustness, and accuracy. Leng et al. [57] proposes a novel digital twin driven joint optimisation approach to maximising the utilisation and efficiency of a large-scale automated high-rise warehouse product-service system continuously.

The above research shows that when the data can flow between the physical space and the virtual space in a digital twin, the virtual and real agents can communicate smoothly, a distributed multi-agent scheduling driven by digital twin becomes possible. Without doubt, digital twin becomes the key technology to solve this problem. Therefore, this paper proposes a bi-level distributed dynamic scheduling with a WDTSA and SUDTSAs. The unified modeling method is first applied to map the production factors of a workshop and configure the service unit digital twin scheduling agents. The virtual workshop is constructed quickly according to the production task and process information. Then the two levels work together to organize the production scheduling and reduce the calculation cost.

3 The Dynamic scheduling problem definition and the overall solution framework

3.1 Bi-level dynamic scheduling mathematic model

A workshop typically consists of several service units, thus a job shop scheduling problem can be defined at two levels: the workshop level and service unit level. All service units work in the same manner and in parallel, so this bi-level scheduling system can accommodate multiple service unit scheduling agents in parallel to support multi-agent flexible job shop scheduling.

At the workshop level, the problem is defined as follows. When a dynamic event(e) is triggering a job scheduling, given a set of service units $\boldsymbol{U}$ ($\boldsymbol{U}=\{1,2,...,l,...,u\}$, $l\in\{1,2,...,u\}$) and their current utility statuses as resource constraints (detailed in resource information section 3.3) and a set of new jobs (n_e) to be scheduled as task, the goal is to achieve the best production performance such as the minimum makespan by decomposing the overall jobs into sub-jobs and assigning proper new sub-jobs to corresponding service units.

At the service unit level, a service unit job scheduling problem is defined in response to the workshop level scheduling. The resource constraints include the number of machine/storage device/logistics equipment, the number of buffer/handling equipment/code scanning equipment, other resources such as measure resources and their utility statuses within the unit. The assigned new sub-jobs to the unit are tasks to be completed in the unit on schedule. The goal is to achieve the best unit production performance such as the minimum makespan.

This bi-level job shop scheduling based on service units (SUFJSP) can be described as follows.

Goal: the best production performance such as the minimum makespan.

Constraints- six basic element sets:

$n_e\times u\times\{m_l, b_l, a_l, r_l\}$, where n_e is the new job number of the eth dynamic event, u is the service unit number, m_l is the number of machine/storage device/logistics equipment within the lth service unit, b_l *is* the number of buffer/handling equipment/code scanning equipment within the lth service unit, a_l is the assigned job number to the lth service unit, and r_l is the number of other resources in the lth service unit. Each job i in the lth service unit consists of lni ordered operations.

The service unit set $\boldsymbol{U}$ ($\boldsymbol{U}=\{1,2,...,l,...,u\}$, $l\in\{1,2,...,u\}$), the job set $\boldsymbol{J}$ ($\boldsymbol{J}=\{J_{1,1}, J_{1,2},...,J_{i,l},..., J_{a'u+au,u}\}$, $i\in\{1,2,..., a'_l+a_l\}$), where a'_l is the new assigned job number to the lth service unit, the operation set $\boldsymbol{O}$ ($\boldsymbol{O}=\{O_{1,1,1},O_{1,1,2},\ldots,O_{l,i,j},\ldots,O_{u,a'u+au,un(a'u+au)}\}$, $j\in\{1,2,...,lni\}$,), the machine/storage location/logistics equipment set in lth service unit $\boldsymbol{M_l}$ ($\boldsymbol{M_l}=\{M_{l,1},M_{l,2},\ldots,M_{l,k},\ldots,M_{l,ml}\}$, $k\in\{1,2,...,m_l\}$), the buffer/handling equipment/code scanning equipment set in lth service unit $\boldsymbol{B_l}$ ($\boldsymbol{B_l}=\{B_{l,1},B_{l,2},\ldots,B_{l,k'},\ldots,B_{l,bl}\}$, $k'\in\{1,2,...,b\}$), and the other resource set in lth service unit $\boldsymbol{R_l}$ ($\boldsymbol{R_l}=\{R_{l,1},R_{l,2},\ldots,R_{l,k''},\ldots,R_{l,rl}\}$, $k''\in\{1,2,...,r\}$).

The processing time of operation $O_{l,i,j}$ on machine/storage location/logistics equipment

M_{lk} is denoted by $t_{i,j,l,k}$. The occupied time of operation $O_{l,i,j}$ on the buffer/handling equipment/code scanning equipment $B_{lk'}$ is denoted by $t_{i,j,l,k'}$. The other time of operation $O_{l,i,j}$ on other resource $R_{lk''}$, such as measuring equipment/personnel, is denoted by $t_{i,j,l,k''}$. The start sever time of operation $O_{l,i,j}$ in lth service unit is denoted by $S_{l,i,j}$, and the end sever time of operation $O_{l,i,j}$ in lth service unit is denoted by $C_{l,i,j}$. According to the sequence of operations, the start processing time of the operation immediately after the operation $O_{l,i,j}$ in l'th ($l' \in \{1,2,...,u\}$) service unit is denoted by $S_{l',i',j'}$.

The orderly arrival time of new job(s) or the occurrence time of abnormal production event is E_e,. Each job includes one or more operations, and the sequence of operations is predetermined. Each operation can be processed on several different machines/storage location/logistics equipment supported by proper services in the corresponding service units. The processing/storage/transportation times vary with the performance of the corresponding machine/storage location/logistics equipment. The typical objectives are minimum makespan, the total tardiness of all jobs, minimum maximal machine workload and minimum total machine workload, especially the first one. The goal of scheduling is to select and assign the most suitable service units with corresponding resources and available statues for processing appropriate sub-jobs when the dynamic event at the time E_e is triggering a rescheduling, and then each service unit will determine its own best processing/storage/transportation sequences and start time of each sub-job on the corresponding machine/storage location/logistics equipment, so that some performance indexes of the whole system can be optimized. At the time of E_e, once the arrival of new jobs, machine failure or other dynamic events are occurred, taking the goal of the minimum makespan as an example, the proposed mathematical model is shown as follows.

The objective function is:

$$f = \min(\max(C_{l,i,lni})) \tag{1}$$

Eq. 1 represents the objective function, which is minimizing the makespan of the last operation of the ith job in the lth service unit.

The top-level constraints:

$$n_e = a'_1 + a'_2 + ... + a'_l + ... + a'_u \tag{2}$$

$$\sum_{l=1}^{u} y_{i,l} = 1 \tag{3}$$

Eq. 2 represents the sum of new jobs/tasks decomposed/assigned into each service unit is equal to the total new job/tasks at the eth dynamic event time. Eq. 3 is a decision variable, it also means a job/task can only be serviced by one service unit at the same time. In the equation, $y_{i,l}$ represents the ith job/task is assigned to the service unit l if the load of the lth service unit is the minimum at that time.

The second-level constraints include internal constraints in a service unit and operation constraints among service units.

$$S_{l,i,j} + x_{l,i,j,k} \times t_{l,i,j,k} + x_{l,i,j,k'} \times t_{l,i,j,k'} + x_{l,i,j,k''} \times t_{l,i,j,k''} = C_{l,i,j} \tag{4}$$

$$C_{l,i,j} \le S_{l,i,j+1} \tag{5}$$

$$C_{l,i,j} \leq S_{l',i',j'} \tag{6}$$

$$\sum_{i=1}^{M_{lk}} x_{l,i,j,k} = 1 \tag{7}$$

$$\sum_{i=1}^{B_{lk'}} x_{l,i,j,k'} = 1 \tag{8}$$

$$\sum_{i=1}^{R_{lk''}} x_{l,i,j,k''} = 1 \tag{9}$$

Eq. 4 represents that if any job on any service unit is being processed, the current scheme cannot be interrupted/rescheduled until the processing procedure is finished. Eq. 5 and 6 represent service order of the $O_{l,i,j}$th operation in the lth service unit and among service units. Eq. 7, 8 and 9 are decision variables, they also mean that any resource can only be occupied by one job/task at the same time.

A scheduling agent is a computing cell which can control the mechanism of problem solving, and this cell may be a program or a machine [43]. When solving a complex scheduling problem, the computing ability of a single agent is limited. Thus, a complex flexible job shop scheduling problem needs to be decomposed into many sub-scheduling problems, the sub-scheduling problems are handled by various agents individually, and then these individual agents are combined into a distributed parallel scheduling computing system according to a certain organizational relationship and architecture. The cooperation between agents is used to solve the complex FJSP problems. Therefore, the basic idea of multi-agent flexible job scheduling system is to divide a complex manufacturing system into several independent autonomous entities, and to distribute a single heavy production scheduling task to the corresponding agents within each autonomous entity. At the same time, the negotiation mechanism between agents is established to achieve the common optimization goal by coordinating their independent behaviors [46].

Note that the utility statues of various resources need to be updated in dynamic job scheduling in real time. For example, any machine broken down, major abnormal execution of scheduled production activities and dynamically diagnosed machine fault information from the physical world need to be fed into the virtual job scheduling. This is the reason why digital twin can help the dynamic job scheduling.

3.2 The overall framework

The unified modeling method [5] is used to create the virtual models of an actual physical workshop's elements and their control logics (see the left hand side of Fig. 1), such as human, machine, material, job and other production elements. Then the virtual models are mapped onto their physical counterparts including resource information mapping and logic mapping, which form information model and logic model in the virtual space (digital twin) respectively (See the middle of Fig. 1). Thus, real-time states of resources in the production process of a physical workshop can parametrically control the corresponding virtual space states to realize physical-to-virtual twinning, and the data model in digital twin could be updated. In the reverse direction, the changes in the digital twin can drive the physical workshop running to realize the virtual-to-physical twinning. The workflow is shown in blue arrows.

How to make changes in the virtual twin is closely related to digital twin applications. In the research, our application is focused on the workshop job scheduling. Our strategy is to add a workshop job scheduling application on the virtual twin (see the right hand side of Fig 1). The added workflow is shown in green arrows. From the top, task/job information could be obtained from the workshop's upper management system such as ERP. Driven by jobs/tasks, guided by information model, logic model and data model in the virtual twin, a series of digital twin agents with learning or decision-making ability can be formed through information fusion and data interaction technology.

The whole workshop is divided into different service units, and the configuration package is used to build a variety of digital twin agents into a service unit. When the digital twin agent of the workshop and the digital twin agents of service units are endowed with scheduling decision-making function, a bi-level dynamic scheduling architecture is formed. When a new task/job is arrived, the workshop digital twin scheduling agent quickly allocates resources and a sub-task/job to different service unit digital twin scheduling agents, and each service unit agent performs distributed negotiation scheduling within itself. Finally, the scheduling scheme is formed and simulated in the virtual workshop, and a reasonable scheme is chosen and sent to the physical workshop. All kinds of disturbance factors in a service unit of the physical workshop inform the digital twin scheduling agent through information fusion. When the service unit digital twin scheduling agent is unable to coordinate within the service unit, the unit scheduling agent will inform the workshop digital twin scheduling agent to allocate new resources to the unit or cooperate with other service unit digital twin agents.

In this way, the workshop production logic can be reconstructed quickly according to the production task/job and process information and the distributed production scheduling is then organized and executed mainly within service units. Under the technologies of physical information fusion, all kinds of production information and scheduling interference factors are acquired by multi-agents in real time, which is convenient for optimizing scheduling strategy.

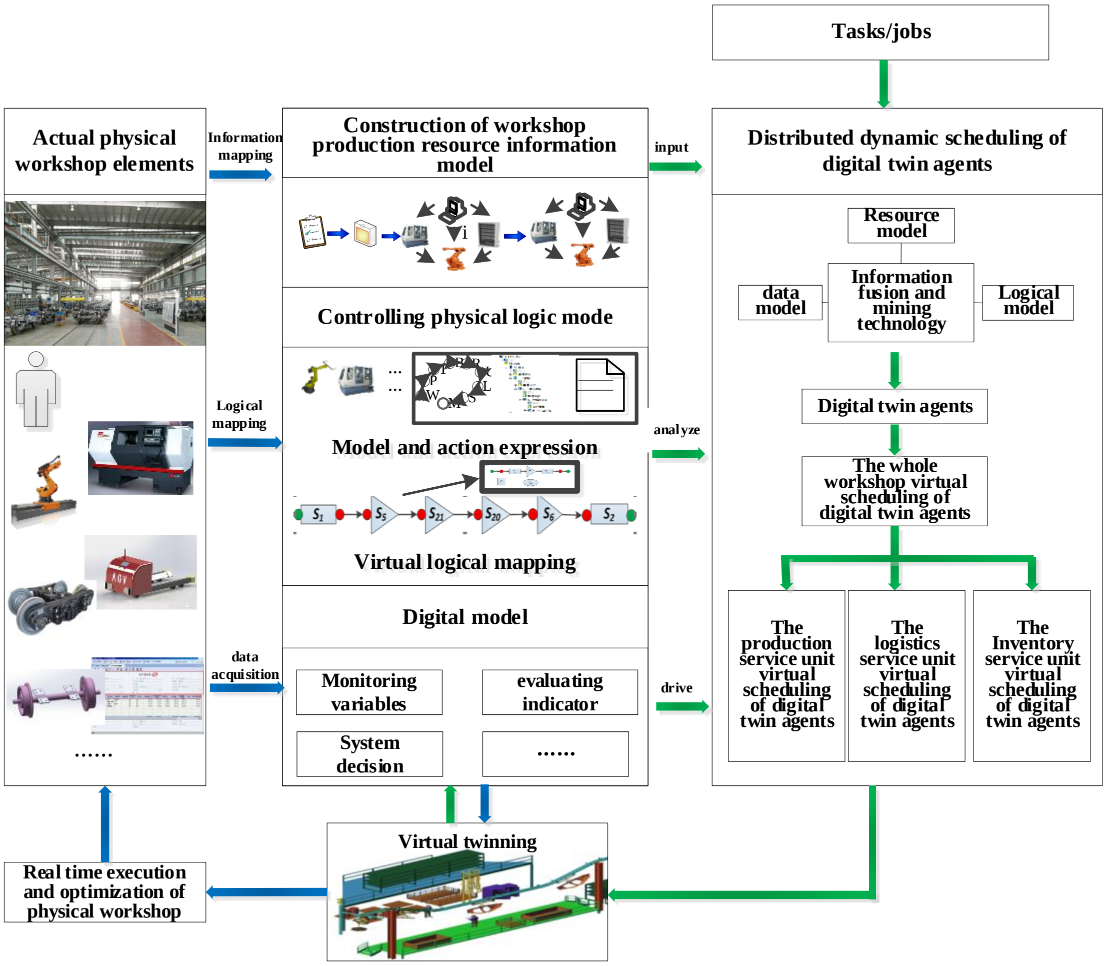

Fig. 1 Dynamic scheduling framework driven by digital twin

3.3 Resource information, control logic and data model of digital twin workshop

The workshop digital twin model includes information model, logic model and data model. The information model characterizes and encapsulates the workshop resources, defines the internal operation mechanism and its external interface. The logic model describes the operation mechanism of the physical workshop on this basis, which describes the process of the parts to be processed into finished products or semi-finished products after logistics and processing. The data model establishes a two-way communication channel between the physical workshop and the virtual workshop.

Firstly, the resource information model of digital twin workshop is built with the method described in [5]. Seven basic logic elements (seven elements) of controller, processor, actuator, buffer, logistics path, flow entity and virtual service node are used to express workshop key components, such as control equipment, processing equipment, logistics equipment, storage equipment, logistics path, production object, external interface, etc. The attributes and behaviors of each element are described graphically and formally. Furthermore, the input / output interface of seven elements is defined and associated with their corresponding CAD models and physical equipment data monitoring points to realize the encapsulation of seven elements and form the workshop production resource information model library.

Secondly, the logic model of digital twin workshop is built. The information model describes the static structure of the workshop from the aspects of the components and organizational structure of the workshop, while the logical model describes the correlations among the elements from the dynamic operation mechanism of the system [5]. Through virtual service node, the seven elements are associated into the manufacturing execution unit, namely the *service unit*. By encapsulating the internal behavior and external interface, the internal details are hidden to form a digital twin agent of the service unit. The combination of service units can reconstruct all kinds of production organization forms, such as series, parallel connection, assembly, disassembly, etc., and realize the mapping of various production service units and production organization relations. Furthermore, binding a logistics path, logistics equipment and virtual service node can form a logistic serve unit, and a logistics path network model can be formed to realize the mapping of logistics association and logistics organization relationship between service units. Finally, based on the flow entity, service unit and logistics path network model, the process, production and logistics are dynamically related to build a production logic model to realize the orderly logical flow of materials in the virtual workshop, and map the operation mechanism of the system. On this basis, the task simulation, process operation simulation and analysis evaluation of production planning can be realized, and then the production organization optimization and control of physical workshop can be realized [58].

On the basis of the logical model, the data model integrates the real-time status and operation parameters of resources in the production process from the physical workshop, real-time operation data, and virtual data, including simulation process data and results obtained from simulation system and scheduling system, etc. Moreover, through the analysis and evaluation of the virtual and real data fusion to assist production decision-making, the virtual workshop can realize the feedback control from the virtual workshop to physical workshop. The data model can be divided into three levels: basic parameters, evaluation index and system decision [58].

In this paper, basic parameters generally refer to all the available physical workshop resource/job status and operation parameters, as well as the simulated data generated by the virtual workshop. For example, a real-time job can be defined by the job identity number, job status, job executor, job service unit, job start time, job completion time, NC code for the job processing, job exception number, exception content, job exception sender and handler, exception start and end time and the identity number of job changed. In order to characterize the status and performance of the equipment or system, the virtual and real data are counted and analyzed to form the evaluation indexes or constrain functions. The job's constrain functions include earliest completion time, delivery delay rate, delay cost, equipment or worker utilization rate, equipment or worker balance rate and robustness. System decision-making refers to the adjustment of system parameters according to the objectives. The decision-making basis is the evaluation index, which finally acts on the monitoring variables. A job shop scheduling objective decision set includes the shortest time, the lowest cost, the highest efficiency, the best labor balance, the best equipment balance and the best robustness. The Information fusion and interaction for creating digital twin agents are detailed in section 4.

3.4 Bi-level dynamic scheduling structure

The workshop bi-level dynamic scheduling is a distributed multi-agent system composed of three kinds of DT agents. Its structure is shown in Fig. 2, including service unit digital twin scheduling agent (SUDTSA), job digital twin agent (JDTA) and workshop digital twin scheduling agent (WDTSA). The SUDTSA agent unit is composed of production unit digital twin scheduling agents (PUDTSAs), logistics unit digital twin scheduling agents (LUDTSAs) and inventory unit digital twin scheduling agents (IUDTSAs). It is a sub bi-level structure. WDTSA is in the first level, and SUDTSA and JDTA are in the second level. A service unit's SUDTSA and JDTA can communicate with each other and feedback the state information to WDTSA. WDTSA is responsible for allocating resources and tasks to each service unit's job digital twin agent according to the special rules or strategies. Any new jobs issued from a unit JDTA will trigger the corresponding service unit scheduling agent (SUDTSA) to work.

In a service unit digital twin scheduling layer, the SUDTSA is in the upper level, and the digital twin agents of machine, buffer, jobs assigned to the service unit and other resources (tools, tooling, etc.) are connected in the lower level. Each digital twin agent in the service unit, for example, a machine digital twin agent (MDTA), a buffer digital twin agent (BDTA), a assigned job digital twin agent (AJDTA) and other resource digital twin agent (ORDTA), can communicate with each other and feedback the state information to the PUDTSA. According to the negotiation mechanism, the behavior of each digital twin agent and scheduling result are finally determined by the SUDTSA. Through the communication between the virtual digital twin agents and the physical space, the instruction is transmitted to physical space.

Therefore, the whole scheduling structure with a multi digital twin agent system is divided into two levels, which forms a hierarchical scheduling structure with a multi digital twin agent system. In the first level, WDTSA allocates resources and sub-tasks/jobs to the service units. In the second level, the SUDTSA arranges the specific service resources and sequence of sub-tasks/jobs.

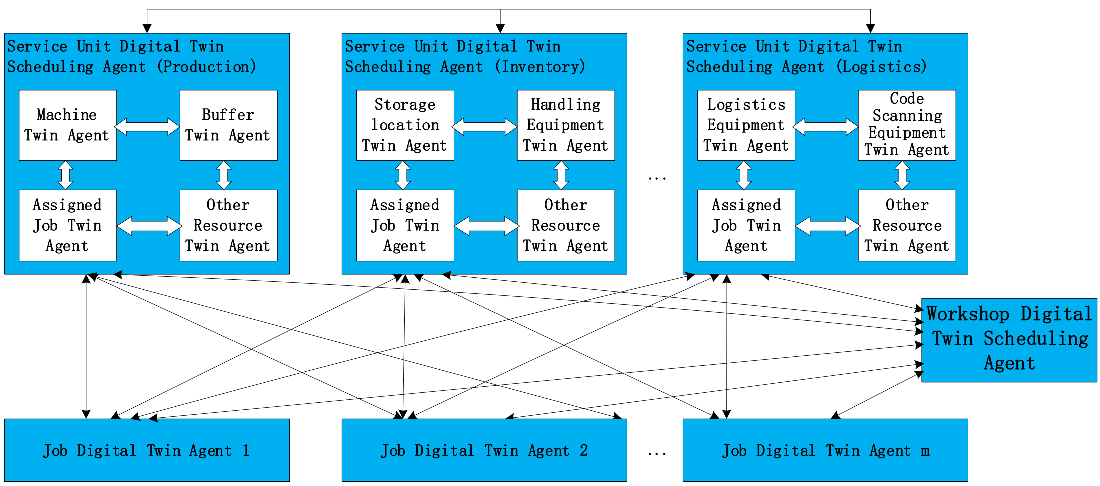

Fig. 2 bi-level dynamic scheduling structure with multi digital twin agent system

4 The bi-level dynamic scheduling realization steps and technologies

The bi-level dynamic scheduling flowchart is shown in Fig. 3. When a new task/job according to the new orders or dynamic disturbances is arriving, a job (generation) agent will generate a job request with the resource information and communicates with the workshop scheduling agent. Workshop scheduling agent decomposes the overall jobs into sub-jobs for different service units. When each service unit receives a new sub-job, its own job generation agent will generate a service unit job request with the service unit resource information and its real-time interference factors. The service unit scheduling agent will then perform the service unit job scheduling, and the service unit digital twin can simulate and verify the implementation of scheduling scheme, and then choose the best scheduling scheme before commanding the service unit to execute the job. During the execution, if any abnormity of executive status of the service unit occurs, the abnormity will be noticed by the service unit job generation agent, which should decide whether it causes a service unit job online scheduling or not. And if the abnormity cannot be managed by the service unit alone, they will be sent to the workshop job generation agent for triggering a workshop job online scheduling.

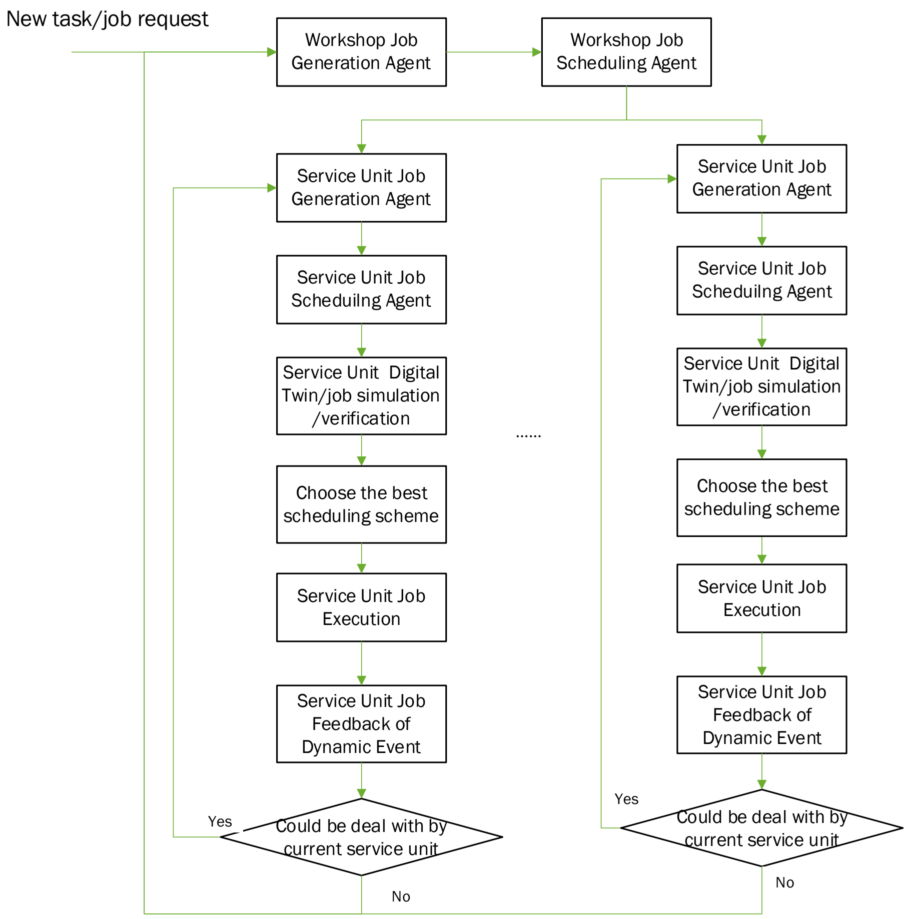

Figure 3 the flowchart of bi-level dynamic scheduling

There are five steps to realize the bi-level dynamic scheduling: 1) set up the dynamic scheduling problem; 2) implement the top-level scheduling based on heuristic rule allocation; 3) implement the dynamic scheduling based on the negotiation mechanism of the service unit level; 4) negotiate with the top-level to deal with the problems that cannot be solved in the service unit; 5) control the jobs of the service unit, obtain the dynamic scheduling problem in real time, go back to step 1 and cycle.

4.1 step 1- how to set up a dynamic scheduling problem

The information fusion and interaction between physical space and information space are key to realize digital twin. In a digital twin, its information fusion and interaction occur between physical objects in physical twin, between virtual models in digital twin, and between physical twin and digital twin, as shown in Fig. 4. Due to these information fusion and interaction technologies, it possible to obtain dynamic scheduling information in real time and then a dynamic scheduling problem can be set up.

In the physical space, the physical twin is divided into three levels, physical resources, physical service units and workshops. Physical resources include human, machine, material, job and other production factors. The interaction and association between resources form relatively independent service units for a specific operation. The resources of a logistics service unit include logistics equipment, code scanning equipment, assigned job and other resources. The resources of an inventory service unit involve storage device, handling equipment, assigned job and other resources. The resources of a production service unit consist of machine, buffer, assigned job and other resources. These service units work together to complete the production jobs in the workshop. Job scheduling and to-do-job instruction generation are in the Top-Down fashion while the operation parameters and real-time statuses of physical resources and units are fed back to the workshop from the bottom to top. The production planning and scheduling instructions of the workshop are distributed to the physical units and physical resources from the top to bottom. The instruction execution and real-time data feedback mechanism in the physical twin are shown in Fig. 5.

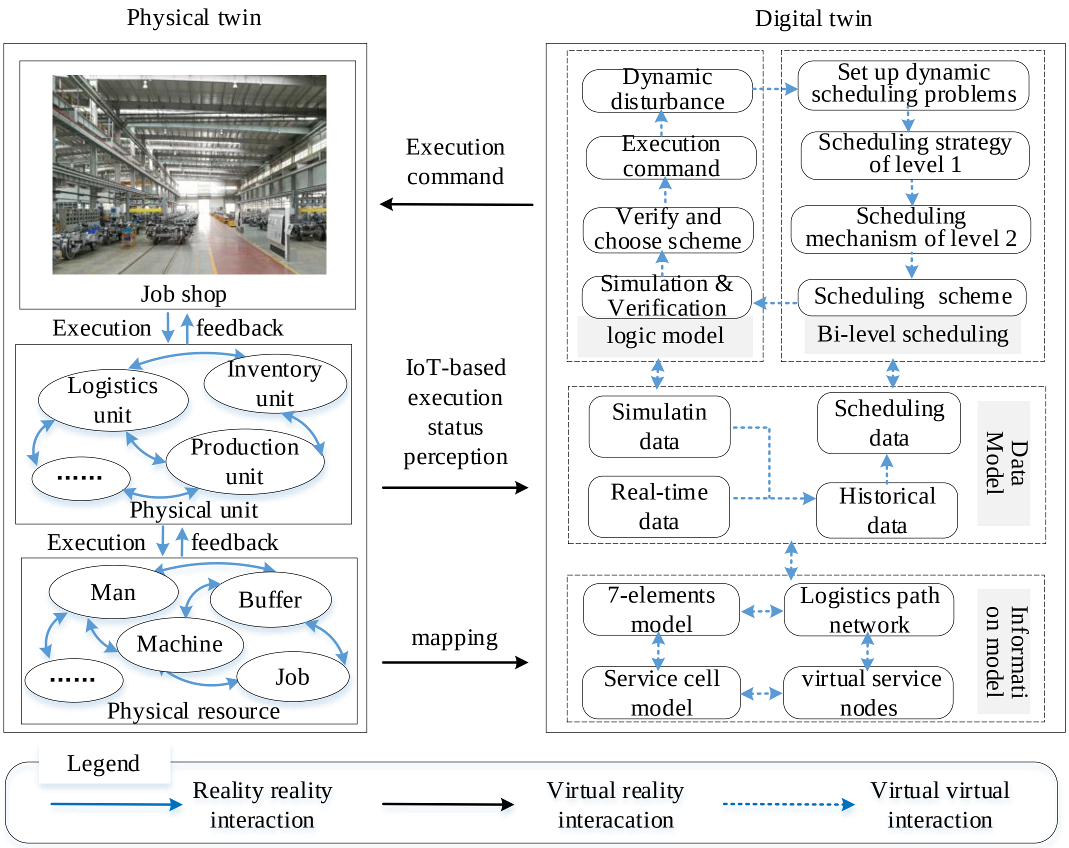

Fig.4 Information fusion and interaction

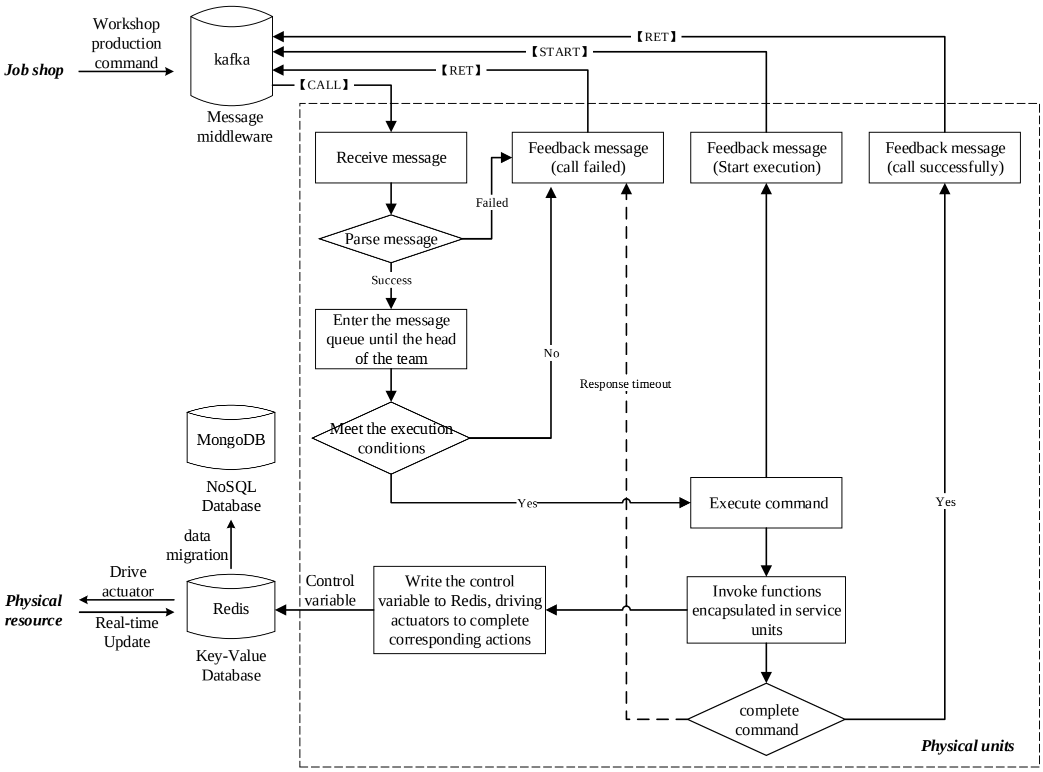

Fig. 5 Command execution and real-time data feedback mechanism in physical twin

In the information space, information model, data model, logic model and scheduling system interact and fuse with each other. The information model is the basis of other models, which can express and model physical resources in a unified way. The data model and logic model provide a foundation for complete scheduling. Based on the scheduling model, a bi-level scheduling mechanism based on multi agents is built to realize the scheduling with production disturbance factors (such as machine failure, resource shortage, order insertion, order cancellation, etc.). The logical model provides simulation and verification for the production planning and scheduling scheme generated by multi digital twin scheduling agents (system). The data model connects the information model, logical model and scheduling system.

Between the physical twin and the digital twin, the physical resources are mapped into the information model, combined with logic model, which can realize the true reflection and description of the physical workshop composition, workshop production layout, logistics path layout and production organization relationship. Based on the real-time perception and transmission of the data model with the industrial Internet of things (IIoT), operation parameters and real-time statuses of physical resources and units can be synchronized and updated to the information models and logic models. Based on the information model, logic model and data model, the multi-digital twin scheduling agent system generates the scheduling scheme and production instructions according to the real conditions, and then issues the scheduling instructions after the simulation verification. The workshop distributes the jobs to the physical unit, acting on the specific resources. Then the execution results and parameters would be fed back to the digital twin data model, thus forming a "command-execution-feedback-regulation" closed-loop.

In order to ensure the real-time collection of abnormal data sources, a data collection method is as shown in the Fig. 5. The real-time data of the workshop is collected by the real-time database, which is called Redis established by SCADA. SCADA transmits the real-time data to the scheduling system by the message queue of message middleware Kafka. The specific transmission process is that the multi-agent scheduling system sends a specific execution instruction to SCADA through Kafka, and the SCADA will pass the execution result synchronously to the scheduling system through Kafka message middleware. Then the data collection of the scheduling system is realized. According to the characteristics of the rapidity of storage and writing of real-time database Redis and synchronization of Kafka instructions, the real time of abnormal data collection is ensured.

The Fig. 5 also describes the instruction execution and parameter feedback mechanism in the physical twin. The workshop production instructions based on the multi-dimensional model and system fusion of digital twins are delivered to physical units through Kafka for analysis and execution. And then the control variables are written into the key-value database Redis, driving the corresponding resource executors to complete the corresponding actions. On the one hand, the changes of resource operation parameters and real-time statuses will be written to Redis to trigger new events in real time. On the other hand, the real-time data in Redis is migrated to MongoDB, a NoSQL database, at a certain frequency for long-term storage in the form of files.

The specific steps of instruction execution and feedback in the physical unit are described as follows. Firstly, Kafka releases the workshop production order to the physical unit in the form of message (using call statement). If the production order can be correctly parsed, it will enter the message queue to wait for execution. Otherwise, it will feed back a failure message of the instruction parsing (using RET statement). Then, if the message at the head of the queue meets the execution conditions, the corresponding function or method encapsulated in the unit will be called and executed. The execution starting message (using the start statement) is fed back, and the control variables are written to Redis. The actuator is driven to execute the instruction action. Otherwise, the call failure message will be fed back. Finally, if the instruction is executed, the success message of the calling is fed back (using RET statement). If the response time is out, the failure message of the call is fed back.

4.2 step 2- how to do scheduling at Level 1 and coordinate Level 2 scheduling problem definition

The digital twin agent of the workshop I is composed of a workshop digital twin scheduling agent, a job digital twin agent and service unit digital twin agents. A multi-digital twin agent structure in the workshop level is shown in Figure 6.

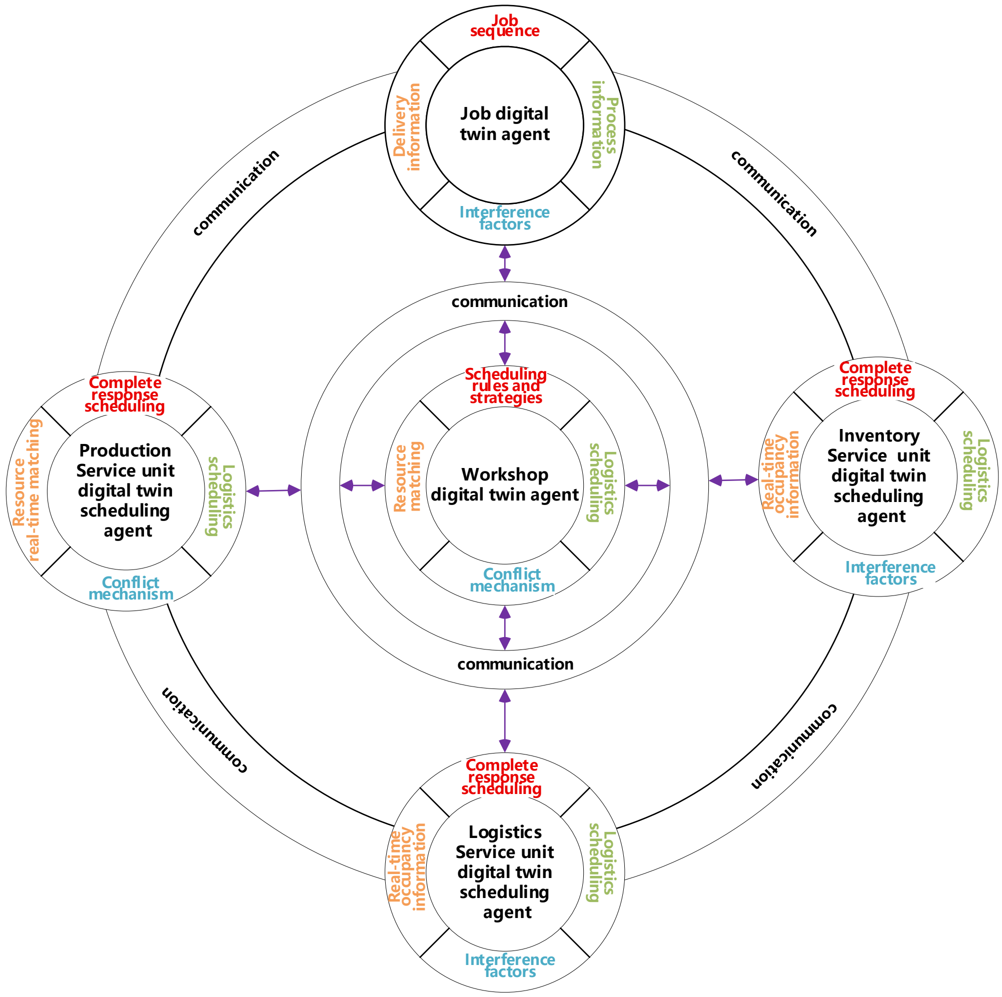

Fig.6 Digital twin agent structure of workshop layer

A service unit twin agent not only maps the production resources within the service unit, but also connects with JDTAs through communication mechanism, so as to obtain the information of the jobs and auxiliary production resources needed (such as tools, fixtures, worker's technical level, etc.). If the inventory is limited and has a great impact on production, it is necessary to establish an inventory unit digital twin scheduling agent which communicates with the workshop digital twin scheduling agent and the production unit digital twin scheduling agents to provide or store raw materials and semi-finished products. That is, the PUDTSA needs to communicate with the IUDTSA to form accurate schedulable information.

The workshop digital twin scheduling agent allocates resources and sub-jobs to each service unit according to schedulable information. For a job which can be processed in different service units, the service unit can be determined by rules or strategies. Service unit digital twins communicate with each other and logistics units organize logistics activities according to the process route to ensure the implementation of the scheduled scheme. Of course, according to certain rules and policies, the task/job also can be assigned to each service unit. The service unit scheduling agent obtains the relevant status information of real-time feedback, and then coordinates the internal scheduling of service unit digital twins. Generally, we assume that the inventory/cache is infinitely large, and the inventory/cache twin agent is not considered in the scheduling process. The auxiliary resources are enough to all service units and have been allocated to the service units by WDTSA in advance. When the logistics equipment is sufficient and the logistics path is smooth, or the processing time is much longer than the transportation time, the impact of logistics on processing is not considered. Then there are only WDTSA, PUDTSA and JTDA agents on the workshop level.

In actual production, due to the large number of machines, the similar machines could be divided into the same production service unit according to the process similarity. If the number of machines in the same service unit is still large, which is not easy to manage, the production service unit can be further divided according to the actual situation of machines managed by workers.

According to the reference [37], scheduling rules can be established in WDTSA. In this paper, WDTSA chooses the rule of Shortest Processing Time (SPT), First in First out (FIFO) and Shortest Processing Time-Work In Next Queue (SPT-WINQ) for the top-level scheduling respectively, due to the SPT rule with the best average waiting time and average flow time (the difference between task end time and task release time), FIFO rule with the best maximum flow time and maximum completion time, and SPT-WINQ rule with mixed effect. Firstly, the job twin agent JDTA is sorted according to the SPT rule, FIFO rule and SPT-WINQ rule individually. Then the job twin agent enters the service units with the minimum workload. Because there are three scheduling strategies, there will be three different scheduling schemes. By comparison, we can choose the best rule.

4.3 Step 3-how to do level 2—service unit dynamic scheduling

Each intelligent service unit is composed of different kinds of agents. In order to ensure that each agent cooperatively completes the internal and external logistics, processing, gauging and other functions of the intelligent service unit, it is necessary to analyze the production organization structure, logistics organization structure and agent function characteristics of the intelligent manufacturing system based on the combination and configuration rules of each element and component in the virtual intelligent service unit. In this paper, we classify the multi-agents under the cooperation of multi-interference factors in production dynamics into: service unit digital twin scheduling agent which solves job task conflict and coordinates resource scheduling strategy, resource agent that sends statuses to other agents (including workpiece, processing equipment, logistics equipment and auxiliary resource agent, etc.) and sub job agent to provide sub job information. Through the cooperation mechanism between agents, the sub-job allocation and sequence are adjusted. The multi-digital twin agent structure of a service unit layer is shown in Figure 7.

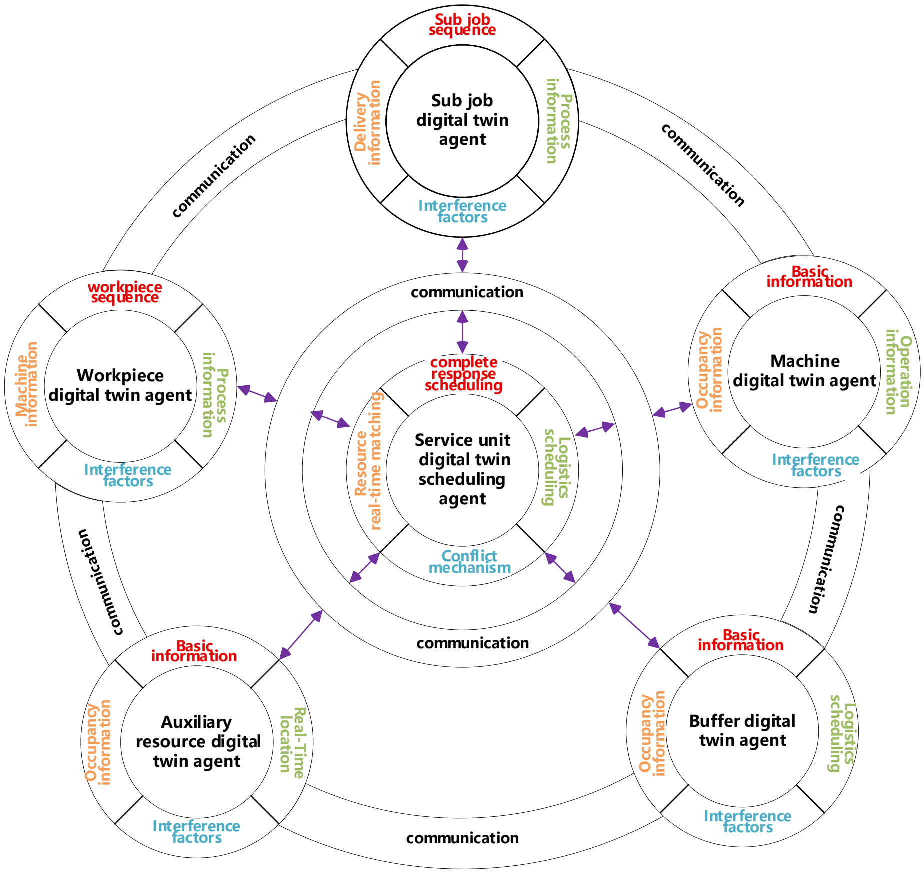

Fig.7 Multi digital twin agent structure on a service unit layer

Under different negotiation mechanisms, the quality of solution obtained by the agent scheduling system is different. In general, the negotiation mechanisms such as CNP[47], game theory[48] and humoral immunity[42] are widely used at present.

In this paper, a multi-agent scheduling method based on humoral immunity [42] is used to schedule a service unit. This paper only considers machine resources and ignores the other resources, such as manipulator, cutter, buffer, fixture, AGV and so on. When a multi-agent scheduling method simulates the humoral immune response mechanism in the service unit, the corresponding relationship is that the service unit twin agent corresponds to the peripheral lymphoid tissue, the decomposed job corresponds to the antigen at the process level, and the machine corresponds to the macrophage. The multi-agent scheduling response negotiation mechanism in the service unit based on humoral immunity is shown in Fig. 8. In order to make the method applicable to the negotiation within the service unit, we modify the negotiation mechanism and process as follows.

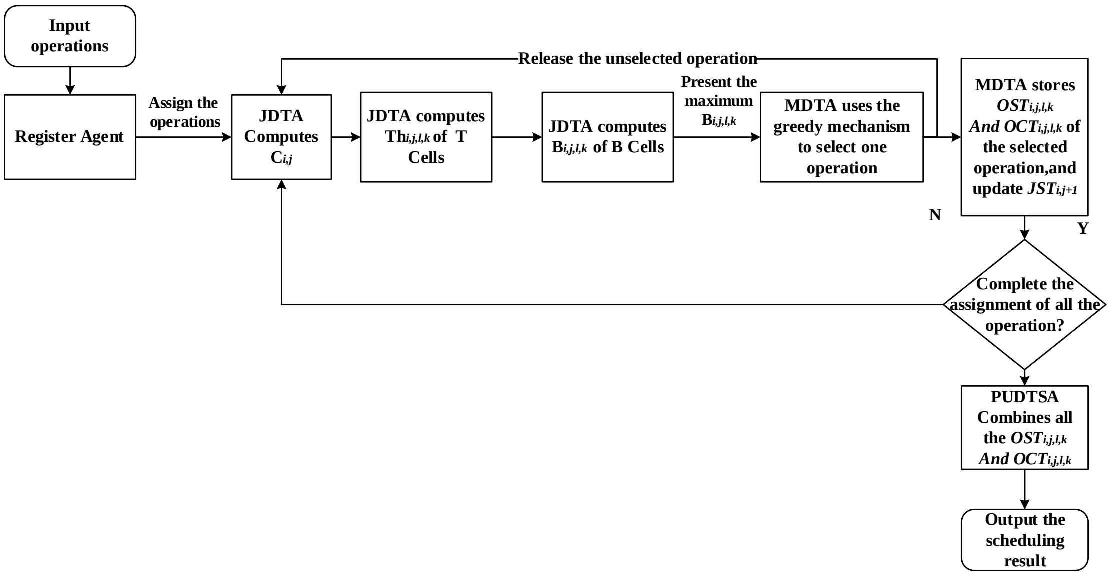

Fig. 8 Negotiation mechanism of humoral immune response within the service unit

In Fig. 8, the antigen concentration, i.e. the shortest processing time of the remaining process of the job:

$$C_{l,i,j} = \min_{\substack{1\le l\le u \\ 1\le k\le M_{ml}}} (t_{i,j,l,k}) + \min_{\substack{1\le l\le u \\ 1\le k\le M_{ml}}} (t_{i,j+1,l,k}) + \ldots + \min_{\substack{1\le l\le u \\ 1\le k\le M_{ml}}} (t_{i,n_i,l,k}) \tag{10}$$

Cell activation time：

$$Th_{i,j,l,k} = \max(JST_{l,i,j}, MST_{l,k}) + p_{i,j,l,k} = OST_{i,j,l,k} + p_{i,j,l,k} = OCT_{i,j,l,k} \tag{11}$$

B Cell activation value：

$$B_{i,j,l,k} = C_{l,i,j} \times \frac{1}{Th_{i,j,l,k}} \tag{12}$$

Among them, $C_{l,i,j}$ is the antigen concentration value of $O_{l,i,j}$, $JST_{l,i,j}$ is the earliest start time of $O_{l,i,j}$ (i.e., the completion time of $O_{l,i,j\text{-}1}$), $MST_{l,k}$ is the earliest start time of machine k in service unit l (i.e., the completion time of the last operation in the processing list of machine k in service unit l), $Th_{i,j,l,k}$ is the armed helper T cell activation time of $O_{l,i,j}$ on machine k in service unit l, $B_{i,j,l,k}$ is the B cell activation value of $O_{l,i,j}$ on machine k in service unit l, $OST_{i,j,l,k}$ is the real start time of $O_{l,i,j}$ on machine k in service unit l, and $OCT_{i,j,l,k}$ is the real completion time of $O_{l,i,j}$ on machine k in service unit l.

$MDTA_{lk}$ selects a processing procedure based on the submitted maximum B cell activation value. When there are multiple identical maximum B cell activation values, $MDTA_{lk}$ preferentially selects the process with high antigen concentration.

Specific process in Humoral Immune Negotiation Mechanism is described as follows.

Step 1: Register and network all JDTA's and MDTA's with the PUDTSA after entering all the processes for all the jobs to be processed.

Step 2: PUDTSA sends a list of antigens to each JDTA, JDTA releases antigens one by one according to the processing sequence of the process (ie, JDTA releases the next antigen in the antigen list after the released antigen is distributed), and calculates the antigen concentration $C_{l,i,j}$.

Step 3: The T cells in JDTA start to calculate their respective activation time $Th_{i,j,l,k}$ and B cell activation value $B_{i,j,l,k}$, and send a maximum $B_{i,j,l,k}$ to both corresponding MTAs.

Step 4: MDTA compares the presented $B_{i,j,l,k}$ using a greedy mechanism, selects the antigen with the largest $B_{i,j,l,k}$ as the process to be processed, extracts $Th_{i,j,l,k}$ information from it, and saves it along with $OST_{i,j,l,k}$ and $OCT_{i,j,l,k}$ in the MDTA's to-be-processed list.

Step 5: MDTA sends the pending list to PUDTSA, combines the MDTA pending lists into a scheduling result table, and outputs the scheduling result.

4.4 Step 4-how to deal with what if scenarios in a service unit and communicate with the Level 1

A scheduled scheme generated by the dynamic scheduling system can be simulated in the virtual workshop with its simulation model, which can be generated for each scheme as shown in Fig. 1. On this basis, we can evaluate the production process. In order to evaluate different scheduling schemes reasonably according to the working conditions, a comprehensive evaluation method based on combination weighting can be adopted. After simulation, the evaluation index value $V=\{v_{sr}|1\le s\le S,1\le r\le R\}$ can be obtained, where v_{sr} is the rth index of sth scheduling scheme, S is the number of scheduling schemes, and R is the number of indexes used to evaluate system performance under different scheduling schemes, e.g. the average utilization rate of machine/buffering equipment and the makespan. Taking V as input, the comprehensive score of the sth scheme T_s could be calculated according to Eq. (13) and (14), the scheme with the highest score is thus selected.

$$T_s=\sum_{r=1}^{R}w_r v_{sr} \tag{13}$$

$$w_r=\varepsilon * w_r^u+(1-\varepsilon)*w_r^o \tag{14}$$

Where, w_r , w_r^u and w_r^o represents the combined weight, subjective weight and objective weight of rth index, respectively; and ε is the distribution coefficient of each weight.

In the physical workshop, based on verified scheduling scheme and simulation details, production orders could be generated, including ex-inventory orders, logistics orders, operation orders and in-inventory orders. These orders are sent to production service unit for execution through a message middleware, e.g. Kafka [59]. Based on the real-time statuses and operating parameters of equipment obtained via a memory database, e.g. Redis, production service units drive the equipment to execute these orders and feedbacks execution results to workshop, i.e. execution started, execution succeeded, execution failed, etc. If all jobs in the scheduling scheme are completed, the execution ends; otherwise, the execution continues, while real-time monitoring whether the execution process deviates from the scheduling scheme.

If the status of any internal production resource in the service unit is abnormal, resulting in large deviation of job/task execution and forming new tasks, new jobs/tasks need to be allocated and sequenced. First, the new jobs/tasks are negotiated and scheduled in the service unit. On the basis of minimizing the makespan, the real-time online scheduling in the service unit also needs to be stable and robust to a certain extent when compared with the original scheduling scheme. If it still fails to meet the requirements after negotiation and optimization, the information should be fed back to the top-level workshop digit twin scheduling agent. The workshop digit twin scheduling agent will distribute some jobs/tasks to other service units or reallocates resources to the original service unit.

4.5 Step 5—how to redefine the problem in Level 1 to start the loop again

The redefinition of a dynamic scheduling problem can be divided into two cases: 1) it comes from the new orders obtained from the ERP system or direct external input; 2) it comes from the major abnormal execution of a scheduled production that cannot be handled by the current service unit according to the real-time feedback status information. For example, the equipment state changes from available state to fault state, for which the production is affected seriously. After the feedback of a job/task execution, compared with the expected delivery time, it could be found that there was serious delay.

For the first case, it should be started from the top level. The workshop digital twin scheduling agent of the first level communicates with each service unit digital twin agent, and assigns the jobs/tasks according to their current situation. The service unit digital twin scheduling agent on the second level schedules these new jobs/tasks and those ones that have not started through the negotiation mechanism.

The second case will lead to the failure of internal negotiation mechanism of digital twin scheduling agent in a service unit generally, that is, no matter how to optimize, it will not get satisfactory results because of insufficient resources and production capacity. Therefore, it is necessary to generate new tasks and send requests to the workshop digital twin scheduling agent in the first level to assign tasks to service units with relatively fewer jobs/tasks. Or the top-level digital twin scheduling agent allocate more resources to the current service unit, such as logistics equipment, man and other production resources, in order to expand the capacity of the service unit.

Both of the new jobs/tasks will dynamically redefine scheduling problem and start the loop again.

5 Experimental study

5.1 Verification example

In this paper, Kacem8×8 benchmark example[42] is selected as the verification example, and the job examples are shown in the table 1. It supposes that the jobs in Table 1 are the new arrivals at the time of E_1=0.

Table 1 Kacem 8x8 (Time unit: hours)

		M1	M2	M3	M4	M5	M6	M7	M8
J1	$O_{1,1}$	5	3	5	3	3	–	10	9
	$O_{1,2}$	10	–	5	8	3	9	9	6
	$O_{1,3}$	–	10	–	5	6	2	4	5
J2	$O_{2,1}$	5	7	3	9	8	–	9	–
	$O_{2,2}$	–	8	5	2	6	7	10	9
	$O_{2,3}$	–	10	–	5	6	4	1	7
	$O_{2,4}$	10	8	9	6	4	7	–	–
J3	$O_{3,1}$	10	–	–	7	6	5	2	4
	$O_{3,2}$	–	10	6	4	8	9	10	–
	$O_{3,3}$	1	4	5	6	–	10	–	7
J4	$O_{4,1}$	3	1	6	5	9	7	8	4
	$O_{4,2}$	12	11	7	8	10	5	6	9
	$O_{4,3}$	4	6	2	10	3	9	5	7
J5	$O_{5,1}$	3	6	7	8	9	–	10	–
	$O_{5,2}$	10	–	7	4	9	8	6	–
	$O_{5,3}$	–	9	8	7	4	2	7	–
	$O_{5,4}$	11	9	–	6	7	5	3	6
J6	$O_{6,1}$	6	7	1	4	6	9	–	10
	$O_{6,2}$	11	–	9	9	9	7	6	4
	$O_{6,3}$	10	5	9	10	11	–	10	–
J7	$O_{7,1}$	5	4	2	6	7	–	10	–
	$O_{7,2}$	–	9	–	9	11	9	10	5
	$O_{7,3}$	–	8	9	3	8	6	–	10
J8	$O_{8,1}$	2	8	5	9	–	4	–	10
	$O_{8,2}$	7	4	7	8	9	–	10	–
	$O_{8,3}$	9	9	–	8	5	6	7	1
	$O_{8,4}$	9	–	3	7	1	5	8	–

Generally, the service units are divided according to the process similarity. However, the processing functions of all the eight machine tools in this example are similar (only the processing time is different probably). In order to reflect the advantages of the bi-level distributed scheduling, this example divides the eight machine tools into two service units. M1, M2, M3 and M4 are in the first service unit, and M5, M6, M7 and M8 are in the second service unit.

5.2 Experimental results and analysis

In this experiment, JADE is used as the development environment of a multi-agent system. Java language and multithreading technology are used for developing codes. Each thread represents an agent. The whole multi-agent system runs on a computer with 2.8 GHz CPU and 8 GB memory. The benchmark example of the test system uses kacem8*8 to verify the algorithm. It is supposed that a batch of jobs from ERP arrive at E_1=0 time (see Table 1). According to SPT, FIFO and SPT-WINQ rule, the scheduling is carried out respectively. If the system does not perceive other dynamic interference factors, the scheduling results are shown in figure 9, 10, and 11.

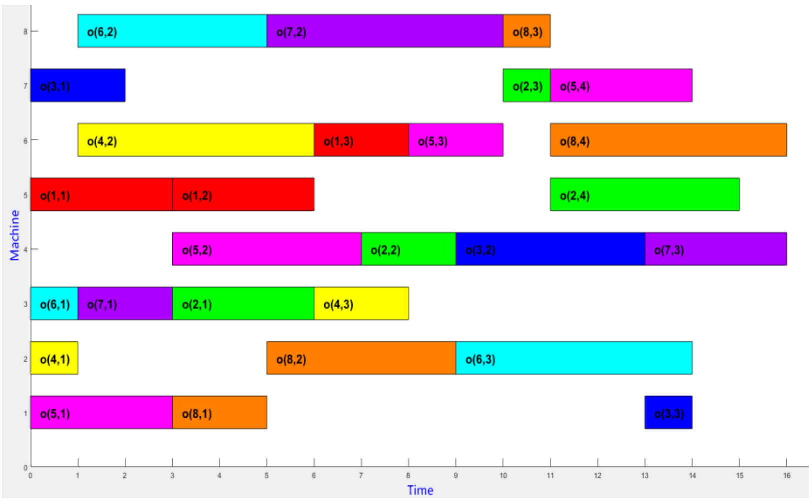

Fig. 9 The scheduling results of SPT rule with new arrivals at E_1=0

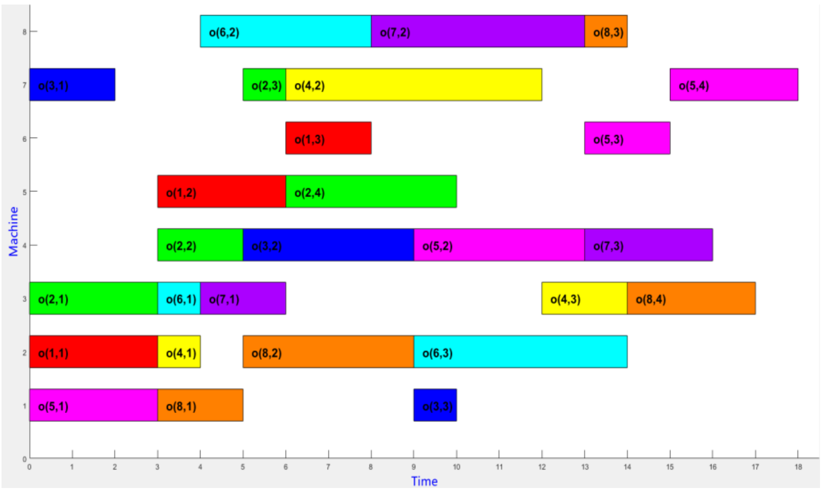

Fig. 10 The scheduling results of FIFO rule with new arrivals at E_1=0

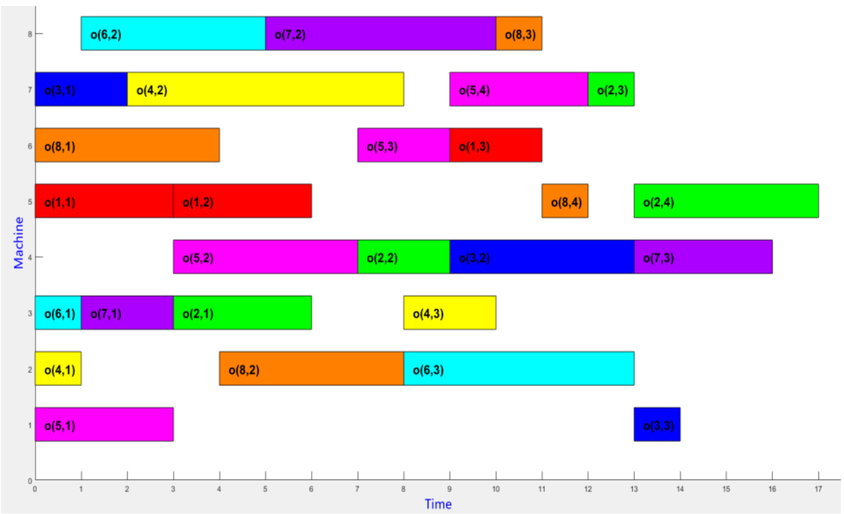

Fig. 11 The scheduling results of SPT-WINQ rule with new arrivals at E_1=0

In this paper, based on the bi-Level dynamic MAS scheduling architecture, the rule scheduling and humoral immune algorithm can be combined by different levels using different methods. From the result in figure 9, we can see that the optimal solution of scheduling result based on STP is 16. Although the optimal solution 14 in [42] is not reached, it is a reasonably feasible solution. And with the increase of the problem scale, the real time is more significant. For the bi-level distributed multi-agent system, when the dynamic changes occur, the coordination within the service unit twin agents should be considered at first to ensure the stability of the system.

Combined with the previous digital twin framework, on the one hand, in the process of scheduling optimization, a twin agent can perceive actual situations of the physical twins in real time, and connect the data into the scheduling model. It can ensure the effective real-time communication of various scheduling agents in the virtual space. Therefore, once there is an exception that needs to be handled, the online scheduling is carried out immediately. On the other hand, the scheduling optimization results are transmitted to the physical unit by the twin agent to control the execution in the physical space, and then the purpose of the fusion of virtual space and real space can be achieved.

In order to verify the feasibility and effectiveness of the bi-level distributed scheduling to dynamically dealing with new job arrivals and resource changes, another new arrival job experiment and a machine failure experiment are added in succession.

After that, at the time of E_2, the 2th hour, the multi-agent scheduling system perceives two new job arrivals. If there is no other dynamic interference factor, the operation table of the two new jobs J9 and J10 is shown in table 2, and the scheduling results with SPT, FIFO and SPT-WINQ are shown in the figures 12 to 14.

Table 2 the operations of the new arrival jobs at E_2=2

		M1	M2	M3	M4	M5	M6	M7	M8
J9	$O_{9,1}$	6	2	5	-	4	6	7	-
	$O_{9,2}$	3	5	3	4	7	5	8	6
J10	$O_{10,1}$	4	5	-	1	6	5	3	4
	$O_{10,2}$	5	3	5	4	5	6	2	7
	$O_{10,3}$	8	5	2	6	3	4	-	5

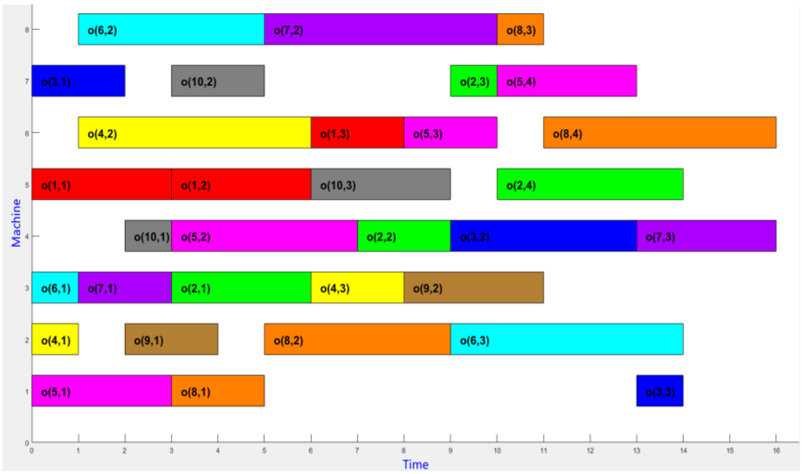

Fig. 12 The scheduling results of SPT rule at E_2=2

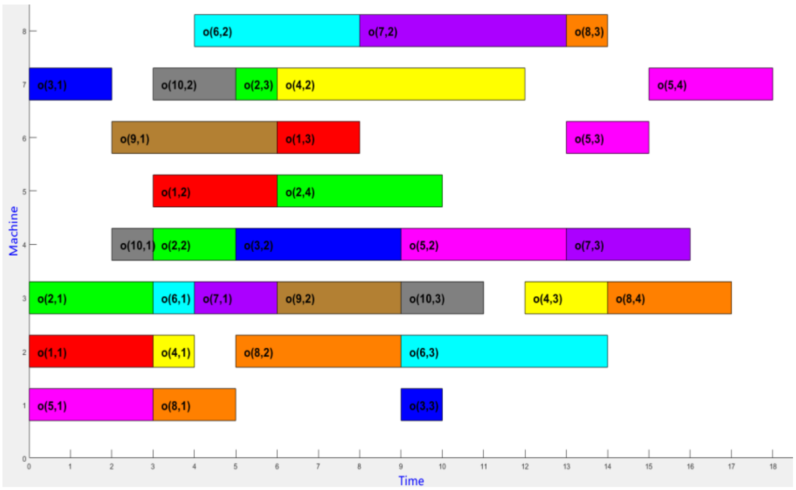

Fig. 13 The scheduling results of FIFO rule at E_2=2

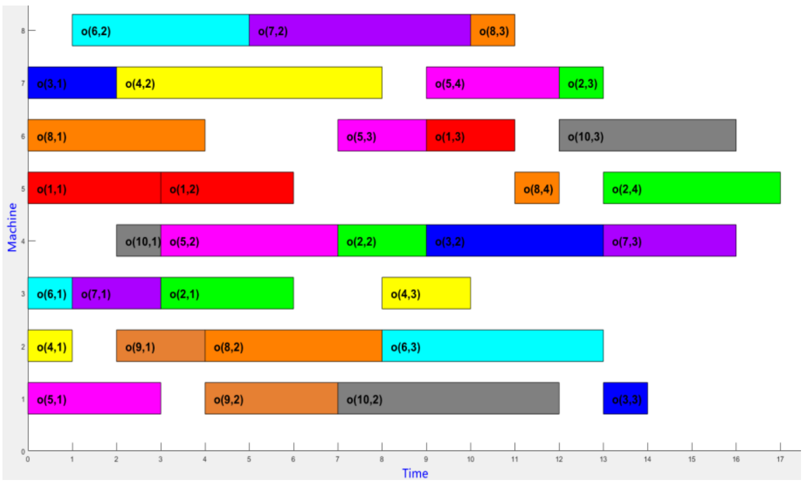

Fig. 14 The scheduling results of SPT-WINQ rule at E_2=2

And then, at the time of E_3, the 6th hour, the service unit 1 twin agent perceives that there is a fault in machine tool 3 (M3). According to the method used in this paper, a real-time scheduling is needed only in the service unit 1. If there is no other dynamic interference factor, the result of scheduling with three different rules is shown in Figure 15 to 17.

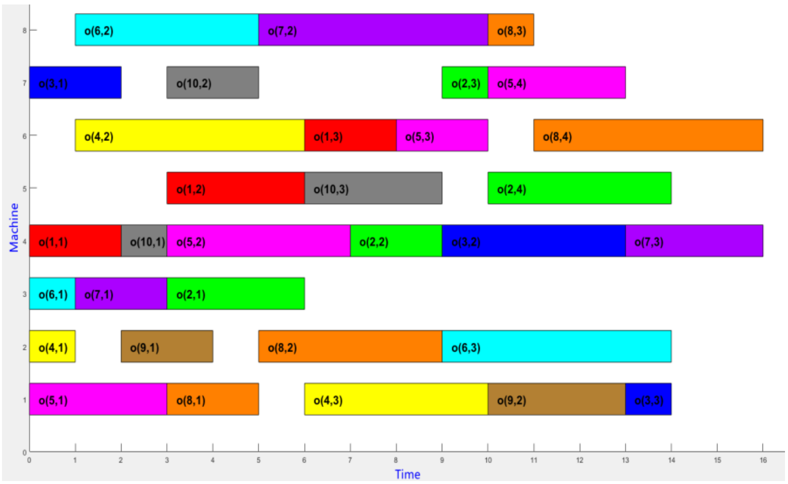

Fig. 15 The scheduling results of SPT rule at E_3=6

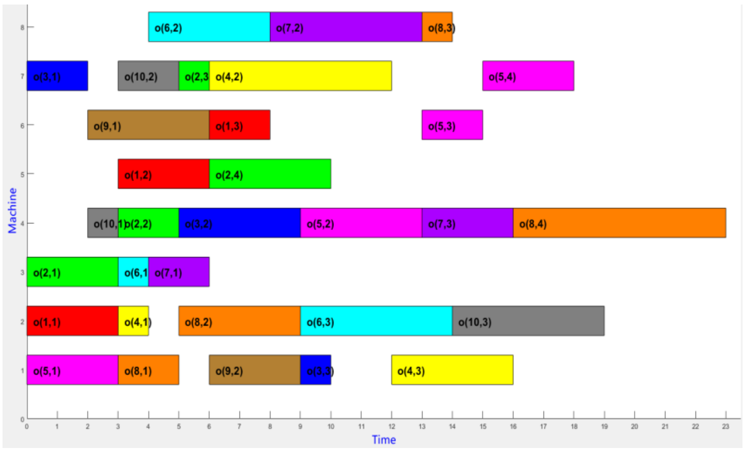

Fig. 16 The scheduling results of FIFO rule at E_3=6

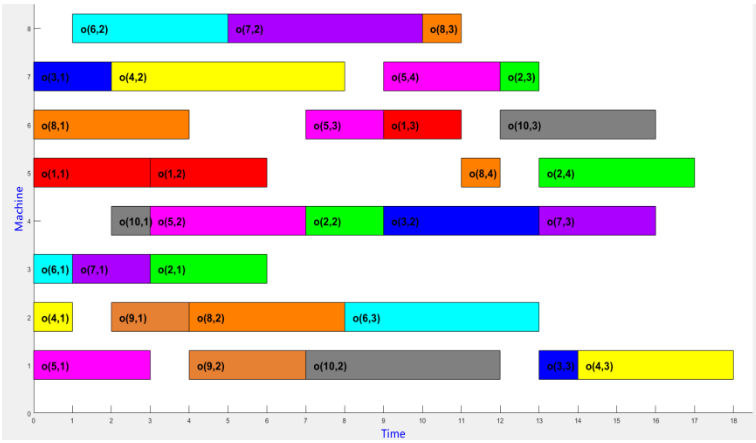

Fig. 17 The scheduling results of SPT-WINQ rule at E_3=6

In order to verify the effectiveness and robustness of the bi-level multi-agent dynamic scheduling architecture proposed in this paper, which combines the rule scheduling and humoral immune negotiation mechanism scheduling, the solution of this method is compared with that of a single-level multi-agent dynamic scheduling architecture, including the single-level architecture based on STP, FIFO, SPT-WINQ and humoral immune negotiation mechanism scheduling individually. Based on the three different conditions, the comparison of calculation results is shown in Table 3 to Table 6. In these tables, condition 1 means the condition with only new arrival jobs at E_1=0, condition 2 means the condition with new arrival jobs at E_1=0 and E_2=2, and condition 3 means the condition with new arrival jobs at E_1=0 and E_2=2, and machine tool fault at E_3=6. RM is the index of robustness, the change degree of the maximum completion time [60].

Table 3 Dynamic scheduling results based on SPT rule

Dynamic condition	Single-level rule architecture (SPT rule)			Bi-level MAS architecture (top-level: SPT)		
	makespan (h)	computing time (s)	robustness RM	makespan (h)	computing time (s)	robustness RM
Condition 1	19	0.142	-	16	0.198	-
Condition 2	21	0.158	0.1053	16	0.251	0
Condition 3	23	0.169	0.0952	16	0.312	0

Table 4 Dynamic scheduling results based on FIFO rule

Dynamic condition	Single-level rule architecture (FIFO rule)			Bi-level MAS architecture (top-level: FIFO)		
	makespan (h)	computing time (s)	robustness RM	makespan (h)	computing time (s)	robustness RM
Condition 1	21	0.138	-	18	0.206	-
Condition 2	24	0.186	0.1429	20	0.251	0.1111
Condition 3	26	0.213	0.0833	21	0.312	0.05

Table 5 Dynamic scheduling results based on SPT-WINQ rule

Dynamic condition	Single-level rule architecture (SPT-WINQ rule)			Bi-level MAS architecture (top-level: SPT-WINQ)		
	makespan (h)	computing time (s)	robustness RM	makespan (h)	computing time (s)	robustness RM
Condition 1	19	0.151	-	17	0.215	-
Condition 2	22	0.196	0.1579	17	0.268	0
Condition 3	24	0.223	0.0909	18	0.319	0.0588

As can be seen from Table 3 to Table 5, although the computing time with the proposed bi-level hybrid dynamic scheduling architecture are longer than those of the general rule scheduling method, the quality of the solution from the bi-level hybrid dynamic scheduling architecture is obviously better than that of the single-level rule scheduling method, and the average solution quality is improved by 25.4%, which is mainly due to the immune negotiation mechanism used in the second level to further optimize the top-level rule scheduling results. The robust performance is also significantly improved, which is mainly due to the bi-level architecture. It is proved that the bi-level multi-agent system method is feasible and effective to solve a dynamic flexible job shop scheduling problem.

Table 6 Dynamic scheduling results under different MAS architectures

Dynamic condition	single-level multi-agent architecture based on humoral immune negotiation mechanism			bi-level multi-agent architecture based on SPT & humoral immune negotiation mechanism		
	makespan (h)	computing time (s)	robustness RM	makespan (h)	computing time (s)	robustness RM
Condition 1	14	0.193	-	16	0.218	-
Condition 2	16	0.235	0.1429	16	0.251	0
Condition 3	17	0.311	0.0625	16	0.312	0

As can be seen from Table 6, at the beginning, the accuracy of the bi-level multi-agent architecture is slightly lower than that of the single-level multi-agent architecture, and the computing time is faster than that of the single-level architecture. However, the objective of the bi-level multi-agent architecture is to make stable scheduling changes when a machine tool fault occurs only within the service unit, without involving the whole work shop rescheduling. Moreover, with the increase of dynamic factors, the gap is decreasing, and even the accuracy is better than that of single-level agent structure.

In order to reduce the difference between the scheduling model and the actual physical model, its simulation evaluation model is needed to evaluate these scheduling schemes quickly. Therefore, this paper uses the workshop simulation system developed by the our team to build the workshop production line simulation model, as shown in Figure 18. Then, the simulation results for the different scheduling rules and dynamic conditions are obtained as shown in table 7. In order to show and evaluate the simulation results, the simulation is timed in seconds instead of hours.

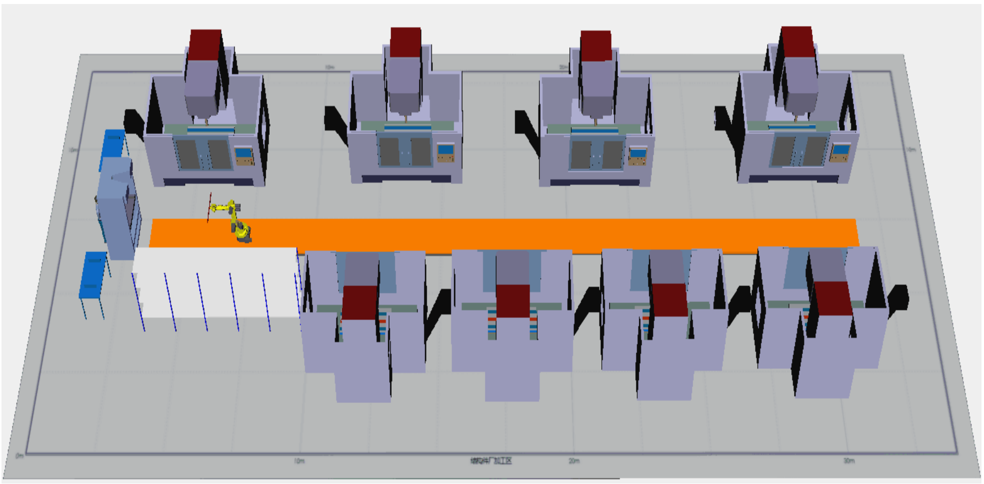

Figure 18 The production line simulation model

Table 7 the simulation results with three scheduling rules and dynamic conditions

rule	condition	Average utilization of machine tool (%)	Average blocking rate of machine tool (%)	Utilization rate of logistics equipment (%)	Simulation completion time (second)
SPT	Condition 1	57.775	1.15	18.5	61506
	Condition 2	64.7625	1.0875	22.7	61854
	Condition 3	64.9125	1.1	22.1	62406
FIFO	Condition 1	50.15	0.95	18.2	68196
	Condition 2	57.2	0.837	21.6	69234
	Condition 3	49.7375	0.7625	16.7	87738
SPT-WINQ	Condition 1	53.175	0.8	17.8	64338
	Condition 2	62	0.9	20.7	66048
	Condition 3	59.5	0.85	19.4	70308

It can be seen from table 7 that when two batches of new jobs arrive one after another, and then a machine tool fault occurs, the performance of SPT top-level strategy is the best in terms of machine tool utilization rate, logistics equipment utilization rate and maximum completion time. This is consistent with the conclusion of references [61] and [62], that is, SPT is the best rule to minimize the average working time and the number of delayed jobs. SPT- WINQ is the best rule for the average blocking rate of processing equipment. However, the blocking rate of the equipment in this example is very small, which can be ignored. Therefore, we select the most reasonable scheduling scheme under STP.

Based on the digital twin, the newly scheduled results can be transmitted to the physical workshop through SCADA, so as to realize the virtual-real reactive scheduling. In addition, the data of reactive scheduling in the process of continuous arrival of new jobs and dynamic interference factors can be excavated into knowledge to form solutions under different working conditions, which is convenient for agent learning and training. The scheduling method can be better and faster when solving such problems in the future.

5.3 Discussions

In this paper, a dynamic scheduling method with bi-level multi-agents driven by digital twin service units is proposed. Compared with the existing multi-agent scheduling system, digital twin service units solve the problem of real-time communication and make real-time dynamic scheduling possible. In reference [42], although a multi-agent scheduling system is built, it is still verified by the specific examples of the static scheduling problem, and there is no verification of dynamic scheduling. The algorithm in this paper realizes the dynamic scheduling of new task arrival and machine failure. At the same time, a powerful fast simulation tool in the digital twin system is used to evaluate and compare the scheme, so as to obtain a more reasonable real-time scheduling solution.

Reference [49] adopts a pre-reaction scheduling model based on machine fault prediction. This method has excellent robustness in the case of high prediction accuracy. However, whether the predicted results of machine failures and new tasks are true or not is still only a certain probability event. This paper proposes a real-time pure reactive scheduling method, which does not need to predict future events and is simpler to operate.

Compared with the meta heuristic scheduling [56], which is also driven by digital twin, there is double benefits with the bi-level multi-agent scheduling architecture. Firstly, it greatly reduces the amount of calculation and effectively improves the computational efficiency. Therefore, it ensures the realization of real-time scheduling. According to the data provided by reference [42], compared with an improved genetic algorithm (eGA), the computational efficiency of the multi-agent system itself is generally increased by more than 30%, and compared with an improved particle swarm optimization algorithm (hDPOS), the computational efficiency can be increased by more than 20 times. Although the scheduling method proposed by this paper needs bi-level agent scheduling, it takes a little longer in a certain scale. But the first level uses policies or rules for scheduling, which is the fastest of all algorithms. The second level employs multi digital twin agents in a service unit for negotiation, the scale is far less than the single-level scheduling scale, and the negotiation speed is greatly improved. Therefore, with the increase of the scale, the time gap between a single-level and the bi-level multi-agent scheduling is gradually reduced in theory and practice, and even after reaching a certain scale, the computing speed and solution accuracy of the bi-level scheduling is likely to be better than that of a single-level scheduling.

The architecture proposed in this paper, when there is a dynamic disturbance in the system, first solves it in a digital twin service unit. Due to the lack of resources, task delay and other reasons, when it cannot be solved, the workshop digital twin scheduling agent is used to coordinate between different service units. That is to say, the local adjustment of the service unit can ensure the overall stability of the system. For example, when E_3=6, the third machine tool broke down. The third machine belongs to the first service unit, so only the internal adjustment of the first service unit is considered. Tasks need to be assigned to other machines in the first service unit. It can be seen that after the adjustment, the production of other service units will not be affected, and there is no need to transport materials between service units. Generally, the buffer is shared in the same service unit, and the internal logistics changes little. This is of great significance for practical production.

6 Conclusion

In this paper, a new bi-level distributed dynamic scheduling architecture is proposed, which is based on its service unit digital twin scheduling agents. Firstly, a service unit of the workshop is packaged according to seven elements to become a digital twin agent. Secondly, the real-time communication between the service unit twin scheduling agent and the physical service unit can complete the virtual scheduling. Then, the workshop is divided into two levels, namely, the service unit digital twin as the second level and the workshop digital twin as the first level. Each level is composed of different agents in the virtual space. At the workshop level, the resources and jobs are assigned to each service unit agent with the rules according to the delivery date/time, standard working hours and current production status. At the service unit level, the multi-agent scheduling negotiation mechanism within the unit is used to allocate and sort jobs to each machine/storage device/logistics equipment in order to achieve dynamic scheduling. Finally, an example is given to illustrate the usability, feasibility and real-time performance of the architecture.

Based on the breakthrough of the existing advanced digital twin technology, this paper constructs a new scheduling architecture. However, it is only a framework at present, and there are many problems which should be studied and explored in the future. We think it should be solved from the following three points. Firstly, optimizing the multi-agent negotiation algorithm to obtain better solution is undoubtedly one of the key technologies to solve the problem. The second key technology lies in the agents self-learning, self-adaptive and self-adjusting technology so that scheduling experience, rules and knowledge can be learnt and a better dynamic scheduling optimization scheme is enabled. The third key technology is to apply cloud and edge side computing and storage capabilities to further improve the computational efficiency and ensure the real-time operation of production lines and workshops.

Acknowledgements

This work is supported by National Major Research and Development Project of China (Grant number 2020YFB1708000).

Reference

[1] Leng J., Yan D., Liu Q., Xu K., Zhao J. L., Shi R., Chen X. ManuChain: Combining permissioned blockchain with a holistic optimization model as bi-level intelligence for smart manufacturing. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, 50(1), 182-192.

[2] Wang H., Wang J., Xu H., Zhao S. Distributed stochastic model predictive control for systems with stochastic multiplicative uncertainty and chance constraints. ISA Transactions. 2021. DOI: https://doi.org/10.1016/j.isatra.2021.03.038.

[3] Chen R, Yang B, Li S, et al. A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem[J]. Computers & Industrial Engineering, 2020, 149: 106778.

[4] Wang Z. J., Zhan Z. H., Kwong S., Jin H., Zhang J. Adaptive granularity learning distributed particle swarm optimization for large-scale optimization. IEEE transactions on cybernetics., 2021, 51(3), 1175-1188, doi: 10.1109/TCYB.2020.2977956.

[5] Jiang H, Qin S, Fu J, Zhang J, Ding G. How to model and implement connections

between physical and virtual models for digital twin application. J Manuf Syst 2020. https://doi.org/10.1016/j.jmsy.2020.05.012.
[6] Tao F, Liu W, Zhang M, Hu TL, Qi QL, Zhang H, et al. Digital twin five dimensional model and its application in ten fields. Comput Integr Manuf syst (Chinese), 2019;1:1-18
[7] Grieves M, Vickers J. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. Transdiscipl. Perspect. complex Syst., Springer; 2017, p. 85–113
[8] Rosen R, Von Wichert G, Lo G, Bettenhausen KD. About the importance of autonomy and digital twins for the future of manufacturing. IFAC-PapersOnLine 2015;48:567–72.
[9] Negri E, Fumagalli L, Macchi M. A review of the roles of digital twin in cps-based production systems. Procedia Manuf 2017;11:939–48.
[10] Kritzinger W, Karner M, Traar G, Henjes J, Sihn W. Digital Twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine 2018;51:1016–22.
[11] Zheng Y, Yang S, Cheng H. An application framework of digital twin and its case study. J Ambient Intell Humaniz Comput 2019;10:1141–53.
[12] Qi Q, Tao F. Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. IEEE Access 2018;6:3585–93.
[13] Tao F, Qi Q, Wang L, Nee AYC. Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: correlation and comparison. Engineering 2019;5:653–61.
[14] Boschert S, Rosen R. Digital twin—the simulation aspect. Mechatron. Futur., Springer; 2016, p. 59–74. https://doi.org/10.1007/978-3-319-32156-1_5
[15] Tao F, Zhang M. Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing. IEEE Access 2017;5:20418–27.
[16] Zhang H, Zhang G, Yan Q. Digital twin-driven cyber-physical production system towards smart shop-floor. J Ambient Intell Humaniz Comput 2019;10:4439–53.
[17] Tao F, Zhang H, Liu A, Nee AYC. Digital twin in industry: State-of-the-art. IEEE Trans Ind Informatics 2018;15:2405–15.
[18] Liu Q, Zhang H, Leng J, Chen X. Digital twin-driven rapid individualised designing of automated flow-shop manufacturing system. Int J Prod Res 2019;57:3903–19.
[19] Zhuang C, Liu J, Xiong H. Digital twin-based smart production management and control framework for the complex product assembly shop-floor. Int J Adv Manuf Technol 2018;96:1149–63.
[20] Sun X, Bao J, Li J, Zhang Y, Liu S, Zhou B. A digital twin-driven approach for the assembly-commissioning of high precision products. Robot Comput Integr Manuf 2020;61:101839.
[21] Liu, Q; Leng, Jiewu; Yan, Douxi; Zhang, Ding; Wei, Lijun; Yu, Ailin; Zhao, Rongli; Zhang, Hao; Chen, Xin (2020). Digital twin-based designing of the configuration, motion, control, and optimization model of a flow-type smart manufacturing system. J Manuf Syst 2020. https://doi.org/10.1016/j.jmsy.2020.04.012
[22] Gao KZ, Suganthan PN, Chua TJ, Chong CS, Cai TX, Pan QK. A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion. Expert Syst Appl 2015;42:7652–63.
[23] Wang Z, Zhang J, Yang S. An improved particle swarm optimization algorithm for dynamic job shop scheduling problems with random job arrivals. Swarm Evol Comput 2019;51:100594.

[24] Ahmadi E, Zandieh M, Farrokh M, Emami SM. A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms. Comput Oper Res 2016;73:56–66.
[25] Jamrus T, Chien C-F, Gen M, Sethanan K. Hybrid particle swarm optimization combined with genetic operators for flexible job-shop scheduling under uncertain processing time for semiconductor manufacturing. IEEE Trans Semicond Manuf 2017;31:32–41.
[26] O'Donovan R, Uzsoy R, McKay KN. Predictable scheduling of a single machine with breakdowns and sensitive jobs. Int J Prod Res 1999;37:4217–33.
[27] Jamili A. Robust job shop scheduling problem: Mathematical models, exact and heuristic algorithms. Expert Syst Appl 2016;55:341–50.
[28] Wang B, Xie H, Xia X, Zhang X. A NSGA-II algorithm hybridizing local simulated-annealing operators for a bi-criteria robust job-shop scheduling problem under scenarios. IEEE Trans Fuzzy Syst 2018;27:1075–84.
[29] Shen X-N, Han Y, Fu J-Z. Robustness measures and robust scheduling for multi-objective stochastic flexible job shop scheduling problems. Soft Comput 2017;21:6531–54.
[30] Ovacikt IM, Uzsoy R. Rolling horizon algorithms for a single-machine dynamic scheduling problem with sequence-dependent setup times. Int J Prod Res 1994;32:1243–63.
[31] Adil B, Fatma S. M, Alper H. Greedy randomized adaptive search for dynamic flexible job-shop scheduling. J Manuf Syst 2020. https://doi.org/10.1016/j.jmsy.2020.06.005
[32] Sabuncuoglu I, Bayız M. Analysis of reactive scheduling problems in a job shop environment. Eur J Oper Res 2000;126:567–86.
[33] Qin W, Zhang J, Song D. An improved ant colony algorithm for dynamic hybrid flow shop scheduling with uncertain processing time. J Intell Manuf 2018;29:891–904.
[34] Li M, Yang P, Yuan Y, Wang X, Yi D. Dynamic Schedule Method Based on Rolling Horizon Procedure for a Discrete Manufacturing Shop with Uncertain Processing Time. IOP Conf. Ser. Mater. Sci. Eng., vol. 688, IOP Publishing; 2019, p. 55058.
[35] Church LK, Uzsoy R. Analysis of periodic and event-driven rescheduling policies in dynamic shops. Int J Comput Integr Manuf 1992;5:153–63.
[36] Gupta D, Maravelias CT, Wassick JM. From rescheduling to online scheduling. Chem Eng Res Des 2016;116:83–97.
[37] Sels V, Gheysen N, Vanhoucke M. A comparison of priority rules for the job shop scheduling problem under different flow time-and tardiness-related objective functions. Int J Prod Res 2012;50:4255–70.
[38] Ziaee M. A heuristic algorithm for solving flexible job shop scheduling problem. Int J Adv Manuf Technol 2014;71:519–28.
[39] Pérez MAF, Raupp FMP. A Newton-based heuristic algorithm for multi-objective flexible job-shop scheduling problem. J Intell Manuf 2016;27:409–16.
[40] Chou Y, Cao H, Cheng H. A bio-inspired mobile agent-based integrated system for flexible autonomic job shop scheduling[J]. J Manuf Syst 2013, 32(4):752-763.
[41] Sahin C, Demirtas M, Erol R, Baykasoğlu A, Kaplanoğlu V. A multi-agent based approach to dynamic scheduling with flexible processing capabilities. J Intell Manuf 2017;28:1827–45.
[42] Xiong W, Fu D. A new immune multi-agent system for the flexible job shop scheduling

problem. J Intell Manuf 2018;29:857–73.
[43] Lane DM, Mcfadzean AG. Distributed problem solving and real-time mechanisms in robot architectures. Eng Appl Artif Intell 1994;7:105–17.
[44] Zhou Y, Yang J-J, Zheng L-Y. Multi-agent based hyper-heuristics for multi-objective flexible job shop scheduling: A case study in an aero-engine blade manufacturing plant. IEEE Access 2019;7:21147–76.
[45] Zhang Y, Wang J, Liu Y. Game theory based real-time multi-objective flexible job shop scheduling considering environmental impact. J Clean Prod 2017;167:665–79.
[46] Xiong W. Research on Flexible Job Shop Scheduling Problem Based on immune multi-agent system, Doctoral Dissertation of Beijing University of science and technology, https://cdmd.cnki.com.cn/Article/CDMD-10008-1018215774.htm
[47] Zhang S, Wong TN. Flexible job-shop scheduling/rescheduling in dynamic environment: a hybrid MAS/ACO approach. Int J Prod Res 2017;55:3173–96.
[48] Wang J, Zhang Y, Liu Y, Wu N. Multiagent and bargaining-game-based real-time scheduling for internet of things-enabled flexible job shop. IEEE Internet Things J 2018; 6:2518–31.
[49] Alves F, Rocha AMAC, Pereira AI, Leitao P. Distributed Scheduling Based on Multi-agent Systems and Optimization Methods. Int. Conf. Pract. Appl. Agents Multi-Agent Syst., Springer; 2019, p. 313–7.
[50] Cupek R, Ziebinski A, Huczala L, Erdogan H. Agent-based manufacturing execution systems for short-series production scheduling. Comput Ind 2016;82:245–58.
[51] Guizzi G, Vespoli S, Santini S. On The Architecture Scheduling Problem Of Industry 4.0. CIISE, 2017, p. 94–100.
[52] Du H, Liu D, Wang C. An Integrated Scheduling Strategy in Dynamic Scheduling of Manufacturing Execution System. 2018 IEEE Int. Conf. Ind. Eng. Eng. Manag., IEEE; 2018, p. 1889–94.
[53] Zhang M, Tao F, Nee AYC. Digital Twin Enhanced Dynamic Job-Shop Scheduling. J Manuf Syst 2020. https://doi.org/10.1016/j.jmsy.2020.04.008.
[54] Zhang J, Ding G, Zou Y, Qin S, Fu J. Review of job shop scheduling research and its new perspectives under Industry 4.0. J Intell Manuf 2019;30.
[55] Zhou G, Zhang C, Li Z, Ding K, Wang C. Knowledge-driven digital twin manufacturing cell towards intelligent manufacturing. Int J Prod Res 2020;58:1034–51.
[56] Fang Y, Peng C, Lou P, Zhou Z, Hu J, Yan J. Digital-Twin-Based Job Shop Scheduling Toward Smart Manufacturing. IEEE Trans Ind Informatics 2019;15:6425–35.
[57] Leng J, Yan D, Liu Q, Zhang H, Zhao G, Wei, L, Chen X. Digital twin-driven joint optimisation of packing and storage assignment in large-scale automated high-rise warehouse product-service system. International Journal of Computer Integrated Manufacturing, 2019: 1-18. https://doi.org/10.1080/0951192X.2019.1667032.
[58] Jiang H, Ding G, Zhang J. Evolution and Operation Mechanism of Digital Twin Shopfloors. Chinese Mechanical Engineering, 2020; 031(007): 824-32, 41. http://www.cmemo.org.cn/CN/Y2020/V31/I07/824
[59] Apache Software Foundation. Kafka. 2017 (accessed 3 Nov 2020). https://kafka.apache.org/
[60] Zhang G, Wu L, Nie L, Wang Y. Robust Flexible Job Shop Scheduling Method with

Machine Breakdowns[J]. Journal of System Simulation, 2016, 28(04):867-873.
[61] Ho NB, Tay JC. Evolving dispatching rules for solving the flexible job-shop problem. 2005 IEEE Congr. Evol. Comput., vol. 3, IEEE; 2005, p. 2848–55.
[62] Safarzadeh S, Shadrokh S, Salehian A. A heuristic scheduling method for the pipe-spool fabrication process. J Ambient Intell Humaniz Comput 2018;9:1901–18.