Robotics and Computer–Integrated Manufacturing 79 (2023) 102435

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Full length Article

Edge computing-based real-time scheduling for digital twin flexible job shop with variable time window

Jin Wang [a], Yang Liu [b,c,*], Shan Ren [a], Chuang Wang [a], Shuaiyin Ma [d]

[a] School of Modern Post, Xi'an University of Posts & Telecommunications, Xi'an, 710061, P. R. China
[b] Department of Management and Engineering, Linköping University, SE-581 83 Linköping, Sweden
[c] Industrial Engineering and Management, University of Oulu, 90570 Oulu, Finland
[d] School of Computer Science and Technology, Xi'an University of Posts & Telecommunications, Xi'an, 710061, P. R. China

ARTICLE INFO

Keywords:
Edge computing
Digital twin (DT)
Real-time scheduling (RS)
Improved Hungarian algorithm (IHA)
Flexible job shop

ABSTRACT

Production scheduling is the central link between enterprise production and operation management and is also the key to realising efficient, high-quality and sustainable production. However, in real-world manufacturing, the frequent occurrence of abnormal disturbance leads to the deviation of scheduling, which affects the accuracy and reliability of scheduling execution. The traditional dynamic scheduling methods (TDSMs) cannot solve this problem effectively. This paper presents a real-time digital twin flexible job shop scheduling (R-DTFJSS) method with edge computing to address the issue. Firstly, an overall framework of R-DTFJSS is proposed to realise real-time scheduling (RS) through real-time interaction between physical workshop (PW) and virtual workshop (VW). Secondly, the implementation process of R-DTFJSS is designed to realise real-time operation allocation. Then, to obtain the optimal RS result, an improved Hungarian algorithm (IHA) is adopted. Finally, a case simulation from an industrial case of a cooperative enterprise is described and analysed to verify the effectiveness of the proposed R-DTFJSS method. The results show that compared with the TDSMs, the R-DTFJSS method can effectively deal with unexpected and frequent abnormal disturbances in the production process.

1. Introduction

In the rapidly changing market economy environment, manufacturing enterprises face great challenges, such as the globalisation of market competition, the diversification and personalised demands of customers and markets, etc. [1]. In this context, manufacturing enterprises must constantly improve workshop productivity and flexibility to enhance their core competitiveness [2]. As a key link to enhancing the workshop production capacity, production scheduling plays an important role in realising high-quality and high-efficiency production [3]. Therefore, business and academic circles pay more and more attention to the production scheduling problem, especially the FJSS problem, which is suitable for the main production mode of manufacturing enterprises.

In the past few decades, there has been a lot of research on FJSS in a static environment. At the same time, many algorithms have been developed to solve static FJSS problems, such as ant colony algorithm [4], evolutionary algorithms [5], tabu search algorithm [6], genetic algorithm [7], particle swarm optimisation [8] and neighbourhood search algorithm [9]. In addition, some scholars also studied static FJSS problems under resource constraints [10], static FJSS problems considering preventive maintenance and priorities [11], and static distributed FJSS problems [12]. Workshop information is fully known in advance in the static environment and remains constant throughout the production process. Thus, the generated schedule is assumed to be not modified after its execution [13]. However, in the real workshop environment, frequent abnormal disturbances make the actual execution deviate from the production plan, making the original scheduling scheme no longer feasible [14]. Thus, it is necessary to study the FJSS problem by considering abnormal disturbances to ensure the production

Abbreviations: IHA, Improved Hungarian algorithm; R-DTFJSS, Real-time digital twin flexible job shop scheduling; FJSS, Flexible job shop scheduling; IIoT, Industrial Internet of things; PW, Physical workshop; VW, Virtual workshop; RS, Real-time scheduling; R-FJSS, Real-time flexible job shop scheduling; DT, Digital twin; PS, Pre-scheduling; PQM, Pre-processing queue of machines; MOs, Machinable operations; TOP, Temporary operation pool; AM, Available machine; ESSM(s), Existing static scheduling method(s); TDSM(s), Traditional dynamic scheduling method(s).

* Corresponding author.
E-mail addresses: yang.liu@liu.se (Y. Liu), renshan@xupt.edu.cn (S. Ren).

https://doi.org/10.1016/j.rcim.2022.102435
Received 30 April 2022; Received in revised form 23 June 2022; Accepted 25 July 2022
Available online 31 July 2022

process's stability, punctuality, and sustainability [15].

With the rapid development of intelligent technologies (e.g. IIoT, DT, cyber-physical systems, edge computing, artificial intelligence, 5G), more and more manufacturing enterprises are beginning to use these technologies to realise intelligent control and management of the workshop [16]. In this way, manufacturing data can be collected and processed in real-time in the production process, and manufacturing resources can be interconnected through different lifecycle phases, which provides a good opportunity to solve the FJSS problem under frequent abnormal disturbances [17]. Thus, some scholars have attempted to combine intelligent sensor technology, big data analysis technology and visual monitoring technology with production scheduling technology to study the real-time data-driven R-FJSS problem. For example, Ghaleb et al. proposed an RS method considering new task arrivals and random machine breakdowns to deal with the real-time information-driven R-FJSS problem [18]. Li et al. addressed the real-time data-based R-FJSS problem in the context of inadequate transportation resources [19]. Although many achievements have been made in the research of R-FJSS, there are still some problems and challenges in the intelligent manufacturing environment. They are listed as follows.

(1) Most existing research on R-FJSS is carried out in the IIoT environment [20,21]. Although real-time manufacturing data collection can be realised in this environment, it is difficult to widely apply in practice applications because a large amount of collected data is difficult to convert into useful information [22]. Moreover, manufacturing data is exploding in the IIoT environment, and existing cloud-centric architectures cannot meet massive real-time data storage and processing requirements in the workshop [23]. Thus, it is necessary to propose a new R-FJSS architecture to improve the RS system's information processing ability and real-time response-ability.
(2) Most existing research on R-FJSS adopts the predictive-reactive scheduling strategy to respond to abnormal disturbances in the production process [24,25]. The method usually generates an original schedule at the initial time and then performs the rescheduling at the appropriate time point to adapt to the manufacturing environment changes caused by abnormal disturbances [26]. However, in this case, each newly generated rescheduling may be different from the original schedule, resulting in an earlier or later processing time for unallocated operations in the original schedule. This has a negative impact on the associated production activities, thus reducing the stability of the production system. Thus, it is necessary to design a new real-time data-based RS strategy to improve the flexibility of the RS system and achieve sustainable production.
(3) Most existing research on R-FJSS uses heuristic algorithms or meta-heuristic algorithms to solve the RS results [27,28]. For heuristic algorithms, the dispatching rule is the most commonly used RS method, which can obtain RS results in a short time to ensure the rapidity and effectiveness of real-time decision-making [29]. However, no single dispatching rule can simultaneously satisfy the optimisation of multiple objectives in the workshop [30]. Meta-heuristic algorithms can get a better solution than the heuristic algorithm. However, due to the slow solving speed and low solving efficiency, the meta-heuristic algorithm cannot meet the real-time requirements of the RS system. Thus, it is necessary to develop a new real-time operation allocation method to improve solving efficiency, reduce computational complexity, and ensure the high efficiency and reliability of the RS system.

This paper presents an IHA-based R-DTFJSS method with edge computing, a novel idea for R-FJSS to address the above challenges. The proposed method has the following three major contributions.

(1) An overall framework of edge computing-based R-DTFJSS is established. Under this framework, the storage capacity, processing efficiency and transmission speed of real-time manufacturing data can be effectively improved to meet the real-time demand of the RS system. In addition, the real-time interaction between PW and VW can improve the real-time response-ability of the RS system to abnormal disturbances to ensure stable and reliable production operation.
(2) A RS strategy is designed based on the variable time window. Compared with the TDSMs, the proposed RS strategy can perform the real-time operation allocation at each time t according to the PS results in the corresponding time window and the real-time data of the PW. Meanwhile, the time window length can be adjusted dynamically according to the influence degree of abnormal disturbance on production. Therefore, it can quickly eliminate the adverse effects of abnormal disturbance to ensure the stability and flexibility of the RS system.
(3) An IHA-based real-time data-driven operation allocation method is proposed. Compared with the existing RS algorithm, the proposed real-time operation allocation method only assigns the MOs to the optimal machine at each time t by IHA. Therefore, it can reduce the computational complexity to realise the fast and reliable operation of the RS system.

Therefore, based on the above analysis, the edge computing-based R-DTFJSS method proposed in this paper is not simply a study of FJSS but provides a new paradigm for realising efficient, high-quality and sustainable production.

The structure of this paper is as follows. Section 2 summarises the existing research on RS and DT in manufacturing. Section 3 designs an overall framework of edge computing-based R-DTFJSS and analyses each component. The R-DTFJSS model is proposed in Section 4. An IHA-based solve method for R-DTFJSS is provided in Section 5. Section 6 uses an industrial case of a cooperative enterprise to illustrate the performance of the R-DTFJSS method. Section 7 gives conclusions and future works.

2. Literature review

This section reviews the research related to this paper, which is mainly divided into two parts: (1) RS; (2) DT in dynamic workshop scheduling. The literature analysis is presented at the end of this section.

2.1. Real-time scheduling

In manufacturing, the term "real-time" means immediate response to abnormal disturbances without interrupting production operations [31]. Therefore, in some literature, RS is also called rescheduling, online scheduling or dynamic scheduling [18]. The earliest literature on dynamic job shop scheduling was written by Chang et al. [32]. Subsequently, Karsiti et al. proposed a two-level scheduling technique, which uses heuristic rules to determine machine routing and sequencing of operations [33]. A dynamic scheduling algorithm based on fuzzy logic was proposed by Roy and Zhang [34]. They improved the heuristic method and got the approximate optimal solution. Bistline et al. proposed an RS decision support system that allowed planned and unplanned machine outages and customer order changes in the following years [35]. Kutanoglu and Sabuncuoglu studied the scheduling mechanism of the production system based on iterative simulation in a dynamic random environment [36]. It was not until 2003 that Vieira et al. summarised the strategies and methods of dynamic scheduling problems for the first time [37]. They define the rescheduling process, strategies and methods. Since then, more and more scholars have paid attention to studying dynamic scheduling [38–40].

In the existing literature, the methods for solving dynamic scheduling mainly include the following five categories: (1) dispatching rules

[41,42]; (2) simulation-based methods [40,43]; (3) artificial intelligence-based methods, such as particle swarm optimisation algorithm [15,44], genetic algorithm [45] [46] and ant colony optimisation algorithm [47,48]; (4) multi-agent methods [49,50]; (5) integrating methods [51,52]. However, because these TDSMs generally do not consider the real-time status information of manufacturing resources in the production process, they cannot respond quickly to abnormal disturbances.

The continuous integration of the new generation of information technology and the manufacturing industry provides an opportunity to build a workshop big data environment. Some papers began to study the real-time data-driven RS problem in this context. For example, Dang and Tang developed a new deep learning-based real-time data scheduling method and an improved fuzzy algorithm to deal with papermaking workshop scheduling problems [53]. Li et al. presented a hybrid deep Q network method based on real-time data to deal with the FJSS problem with insufficient transportation resources [19]. In addition, in recent years, the authors' research team has also researched FJSS problems based on real-time manufacturing information [2,21,54]. However, due to the lack of real-time interaction mechanisms, the scheduling scheme is different from the actual operation results, thus affecting production efficiency and causing production chaos.

2.2. Digital twin in dynamic workshop scheduling

The term "twin" was first proposed during NASA's Apollo program, which created two spacecraft, one flying into space and the other remaining on the ground as a twin. Subsequently, with the wide application of computer-aided technology and management information system in manufacturing, Grieves mentioned DT in his product life management course in 2003 [55]. Although the definition of DT was unclear at that time, it contained the main components of DT technology: virtual space, physical space and connected interfaces. DT entered a new phase of development in 2011 when the US Air Force Research Laboratory proposed the concept in a paper on predicting the life of aircraft structures [56]. In 2012, the US Air Force Research Laboratory proposed the airframe DT concept [57]. The same year, NASA came up with a widely known definition of DT [58]. Recently, with the development of Industry 4.0, DT technology is more and more widely used in the field of dynamic workshop scheduling. At present, the research on improving the real-time stability and robustness of dynamic workshop scheduling by DT technology is generally carried out in the following three types of workshops: flow shop, job shop and flexible job shop.

For the flow shop, Negri et al. proposed a DT-based proof-of-concept of a heuristics framework for robust scheduling applied to a flow shop scheduling problem [59]. Tliba et al. developed a DT-driven dynamic scheduling approach for a hybrid flow shop based on the combination of both optimisation and simulation [60]. For the job shop, Zhang et al. explored a DT-enhanced dynamic scheduling method to improve productivity [61]. To reduce the scheduling deviation. Fang et al. developed a DT-based job shop scheduling method [62]. For the flexible job shop, Yan et al. addressed the FJSS problem under finite transportation conditions for the DT workshop [63]. Liu et al. proposed a DT-driven adaptive scheduling method to deal with frequent abnormal events in the flexible job shop [64].

From the description of the above literature review, it can be seen that experts and scholars have carried out extensive studies on DT-based dynamic workshop scheduling problems in various types of workshops to improve production efficiency and product quality. However, most of these researches still use traditional scheduling strategies and algorithms to deal with abnormal disturbances in the production process. Under the traditional scheduling model, abnormal disturbance response may not be timely, and the decision cannot be optimised in the RS process. In addition, the existing research on DT-driven dynamic workshop scheduling considering edge computing is quite limited.

2.3. Literature analysis

To further explain the differences between this paper and the existing literature in a similar field, a comparison is made between our paper and the recent literature on dynamic workshop scheduling, shown in Table 1. At the same time, the comparison results are analysed from the application of information technology, rescheduling strategy and rescheduling algorithm.

2.3.1. Application of information technology

At present, most research on the application of information technology to dynamic workshop scheduling is carried out in the environment of IIoT [65,66]. In this environment, manufacturing data is growing exponentially. To solve the problem of manufacturing data explosion, a few experts and scholars began to use edge computing to study the data-driven dynamic workshop scheduling problem to improve manufacturing data storage efficiency and transmission speed [54,67]. However, due to the lack of a real-time interaction mechanism, the dynamic scheduling decision is not optimised or timely, and even the production plan cannot be implemented smoothly. Therefore, DT and edge computing need to be combined to improve workshop production's real-time, robustness, and accuracy. Through a rigorous literature search, only one paper combined DT and edge computing to promote the development of dynamic workshop scheduling [68]. In addition, no application of DT and edge computing in R-FJSS was found.

2.3.2. Rescheduling strategy

In the literature about dynamic workshop scheduling, the predictive-reactive scheduling strategy is generally adopted to deal with abnormal disturbances in the production process [13,52,62,69]. However, under this rescheduling strategy, the response to abnormal disturbance may be delayed or too frequent, resulting in uncontrollable and unstable production. Although some literature has proposed RS strategies to allocate operations in real-time, these strategies still have some defects. 1) they can only respond to individual abnormal disturbances [70–72]; 2) the degree of impact of abnormal disturbance on production is not considered [21].

2.3.3. Rescheduling algorithm

Most literature uses traditional intelligent algorithms to solve dynamic workshop scheduling problems [15,61]. However, these algorithms have a lot of computation in the solving process, and with the increase of machines and jobs, the computational complexity becomes higher. Although some scholars use dispatching rule or game theory to solve dynamic workshop scheduling problems, dispatching rule cannot optimise all objectives [19,72], and Nash equilibrium solutions of game theory sometimes do not exist [2,73].

Thus, based on the above analysis, it can be seen that there are still many shortcomings in the research of dynamic workshop scheduling. This paper proposes an edge computing-based R-DTFJSS method to improve the RS system's stability, rapidity, and accuracy to fill these deficiencies.

3. The overall framework of edge computing-based R-DTFJSS

At present, many experts have proposed the edge computing-based DT framework in their research fields. However, to the best of the authors' knowledge, there is no research on R-FJSS under the edge computing-based DT framework. Thus, different from existing research, edge computing-based DT technology is applied to flexible job shops in this paper. By placing smart sensors and edge equipment in the PW, real-time data in the production process can be collected and processed quickly and accurately. Based on this, real-time operation allocation is carried out at each time t by using IHA through real-time data interaction between PW and VW. Since the operation allocation is driven by real-time data and RS is only intended for machines which are available

Table 1
Summary of the literature on the dynamic workshop scheduling

No.	Reference	Workshop type	Application of information technology	Dynamic factors	Rescheduling strategy	Rescheduling algorithm
1	[2]	Flexible job shop	IIoT	Exception events	RS	Bargaining-game
2	[13]	Flexible job shop	None	Exception events	Predictive-reactive	Greedy randomised adaptive search
3	[15]	Job shop	None	New job arrival	Predictive-reactive	Improved PSO algorithm
4	[19]	Flexible job shop	IIoT	Machine breakdown; New job arrival	RS	Dispatching rule
5	[21]	Flexible job shop	IIoT	Exception events	RS	Infinitely repeated game
6	[52]	Flexible job shop	None	Machine breakdown; New job arrival; Job cancellation	Predictive-reactive	Ant colony optimisation
7	[54]	Flexible job shop	Edge computing	Exception events	RS	Evolutionary game
8	[61]	Job shop	DT	Exception events	Predictive-reactive	Heuristic algorithm
9	[62]	Job shop	DT	Exception events	Predictive-reactive	NSGA-II algorithm
10	[65]	Flexible job shop	IIoT	Machine breakdown	Predictive-reactive	Petri net and ant colony optimisation
11	[66]	Open shop	IIoT	New job arrival	RS	Dispatching rule
12	[67]	Flexible job shop	Edge computing	Machine breakdown	Predictive-reactive	Improved great deluge algorithm
13	[68]	Job shop	DT, Edge computing	Exception events	Completely reactive	Dispatching rule
14	[69]	Flexible job shop	None	Machine breakdown; New job arrival	Predictive-reactive	Hierarchical multi-agent proximal policy
15	[70]	Flexible job shop	None	Machine breakdown; New job arrival	Predictive-reactive	Mixed-integer programming
16	[71]	Flexible job shop	None	Machine breakdown; New job arrival; jobs cancellation and change	Predictive-reactive	Monte Carlo tree search algorithm
17	[72]	Job shop	IIoT	Exception events	RS	Dispatching rule
18	[73]	Flexible job shop	IIoT	Machine breakdown	Predictive-reactive	Game theory
19	[74]	Flexible job shop	None	New job arrival	Predictive-reactive	Evolutionary multitask algorithm
20	This paper	Flexible job shop	DT, Edge computing	Exception events	RS	IHA

for manufacturing, the deviation between planning and execution can be greatly eliminated.

Based on the above description, an edge computing-based R-DTFJSS framework is established, as shown in Fig. 1, which comprises PW, data management and VW. Through the connection of the data management module, the VW can monitor and control the PW in real-time, and the PW can update the data and information of the VW synchronously. Therefore, the intelligent decision-making of R-DTFJSS can be realised through real-time information interaction.

3.1. Physical workshop

The PW consists of the physical entities of a real-world workshop. Its function is to execute the scheduling scheme according to the optimisation results of the VW. At the same time, the real-time data of PW is sent to VW through the data management module, which lays the data foundation for real-time decision-making in the VW.

The PW is equipped with various data acquisition equipment such as smart sensors and radio frequency identification devices (RFID) to obtain all kinds of real-time data on manufacturing resources. For example, smart sensors can collect various real-time data generated by the machine, such as vibration, temperature, etc. RFID can automatically identify, track and manage workpieces, materials and people. However, since most of the collected real-time data may be inconsequential, there is no need to send all the data to the data management module. Therefore, the edge equipment and edge server are installed in the PW. The collected real-time data are pre-processed by edge equipment to remove redundant, misleading and inconsistent information, and the pre-processed data is forwarded to the edge server. The edge server controls the data transfer and summary of the whole workshop and interacts with the data management module. In this mode, only useful data is sent to the data management module, which reduces the amount of data transmitted through the network and improves the real-time data transmission performance. Therefore, the response speed of the R-DTFJSS system to abnormal disturbance can be enhanced. In addition, the data of the information management systems (e.g. ERP, MES, PDM, etc.) are also sent to the data management module for storage and analysis.

3.2. Data management

The function of the data management module is to connect PW and VW. It receives real-time data from PW and optimisation results from VW. The seamless integration, real-time interaction, and fusion of PW and VW can be realised through real-time data management. This part mainly contains three modules: data storage module, data processing module, and data visualisation module. The functions of these modules are as follows.

The data storage module is responsible for storing the data from the PW and VW. In general, the data in the manufacturing process of R-DTFJSS includes structured data (e.g. equipment state, personnel state, etc.) and unstructured data (e.g. audio, video, documents, etc.). The data processing module is responsible for discovering and mining useful information in the collected data to provide information support for subsequent RS. It mainly includes four processes, i.e., data reduction, data analysis, data mining and data mapping. The data visualisation module mainly expresses all kinds of data in the form of graphics. This module interprets data visually through various visualisation techniques (e.g. statement, chart, virtual reality, graph, etc.), displays real-time data, and realises visual interaction.

Through the above three modules, the PW can get the real-time decision-making information sent by the VW, and the VW can get the real-

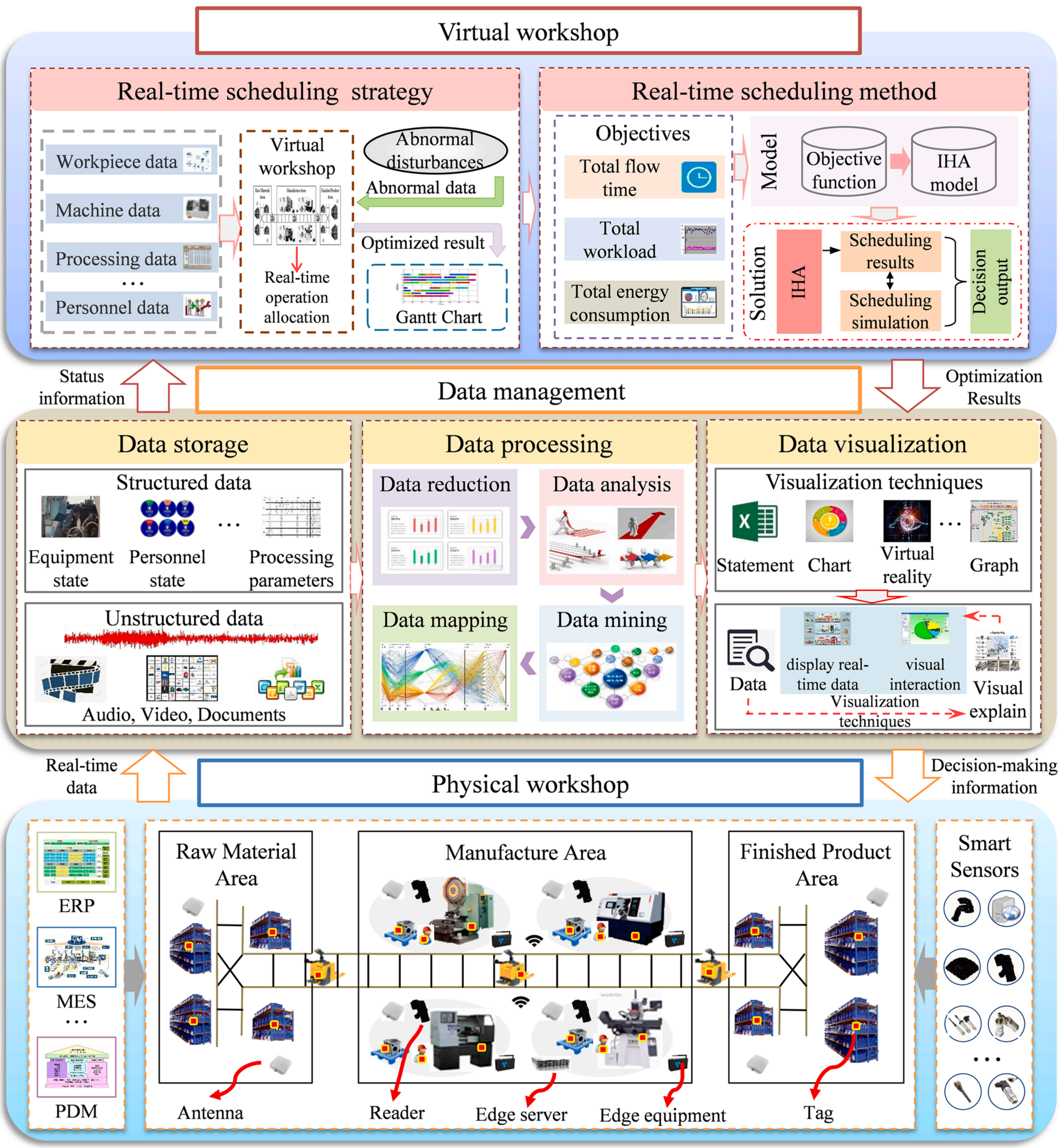

Fig. 1. The overall framework of edge computing-based R-DTFJSS

time status information of the PW.

3.3. Virtual workshop

The VW is the reconstruction and digital mapping of the PW. Its function is to make an RS based on the real-time data of the PW and feed it back to the PW for execution. The PW can simultaneously execute VW production's scheduling scheme in this mode. In addition, through real-time interaction with the PW, the VW can timely and quickly deal with the PW's abnormal disturbance and enhance the RS system's rapidity and stability. The RS strategy module and RS method module are embedded in the VW.

The RS strategy module gives the concrete thinking of RS, that is, according to the real-time status information (e.g. workpiece data, machine data, processing data, personnel data, etc.) of the PW, the real-time operation allocation is performed at each time *t*. When abnormal disturbances occur, the VW can respond to abnormal disturbance in time so as to quickly eliminate the adverse effects of abnormal disturbance.

The RS method module gives the basic flow of RS by IHA. First, the objective functions of total flow time, total workload and total energy consumption are established. Then, based on the idea of RS strategy, the mathematical model of IHA is established. Next, RS results can be obtained by solving the IHA, and the simulation process is carried out. Finally, the decision-making information is sent to the PW for execution.

4. The R-DTFJSS model

In the edge computing-based R-DTFJSS framework proposed above, according to the real-time data of PW, real-time operation allocation is carried out in VW. This section proposes the optimisation objectives that need to be met in operation allocation.

4.1. Problem formulation

In the R-DTFJSS, there are m machines and n jobs. Each job J_i contains n_i operations with sequential constraints to be machined. At least one operation can be processed on multiple machines in the workshop. The main purpose of R-DTFJSS is to generate RS in VW based on the real-time data of the PW to achieve the optimisation objectives of the R-DTFJSS. Based on the above analysis, we use the notation in Table 2 to define the parameters and variables needed for the mathematical model.

In the process of R-DTFJSS, some assumptions are formulated as follows:

(1) All machines are available and idle at the initial time.
(2) In the process of RS, the transportation time of the job is not considered.

Table 2
Notations

Notations	Description
$M = \{M_1, M_2, \cdots, M_k, M_m\}$	Set of machines
$J = \{J_1, J_2, \cdots, J_i, J_n\}$	Set of jobs
$J_i = \{O_{i1}, O_{i2}, \cdots, O_{in_i}, \}$	Set of operations of job J_i
C_{sum}	The total flow time of all the assigned operations
t^p_{ijk}	The preparation time of O_{ij} operated on M_k
t^c_{ijk}	The cutting time of O_{ij} operated on M_k
t^{idle}_k	The idle time of M_k
x_{ijk}	1, if the O_{ij} is processed on M_k and 0 otherwise
WL^{all}	The total workload of all machines
E^{all}	The total energy consumption
P^p_k	The preparation power of M_k
p^{idle}_k	The idle power of M_k
P^c_{ijk}	The cutting power of O_{ij} operated on M_k
C_{ijk}	The completion time of O_{ij} operated on M_k
C^t_{pijk}	The completion time of the assigned operation on M_k at time t during PS
C^t_{Rijk}	The completion time of the assigned operation on M_k at time t during RS
g^t	The number of machines in which C^t_{pijk} and C^t_{Rijk} are not equal at time t
T_n	The n^{th} time window
S^{tl}_n	The beginning time of the n^{th} time window
ξ	The average deviation between PS and RS
ΔT	The time span of the time window
β	The consistency between PS and RS
L_{sh}	The cost that the s^{th} operation is assigned to the h^{th} machine
x_{sh}	1, if the s^{th} operation is assigned to the h^{th} machine and 0 otherwise
t^p_{sh}	The preparation time of the s^{th} operation on the h^{th} machine
t^{idle}_{sh}	The idle time between the h^{th} machine finishing the last assigned operation and the starting of the s^{th} operation
WL^s_h	The total workload of the h^{th} machine before the s^{th} operation is assigned to the h^{th} machine
C^s_h	The total flow time of operations on the h^{th} machine before the s^{th} operation is assigned to the h^{th} machine
t^c_{sh}	The cutting time of the s^{th} operation on the h^{th} machine
p^{idle}_h	The idle power of the h^{th} machine
p^p_h	The preparation power of the h^{th} machine
E^s_h	The total energy consumption of the h^{th} machine before the s^{th} operation is assigned to the h^{th} machine
ε	A large enough number
P^c_{sh}	The cutting power of the s^{th} operation on the h^{th} machine

(3) During the manufacturing execution stage, if an operation is stopped due to machine breakdown, this operation will be reassigned at the next time t.
(4) During RS, abnormal disturbances may occur at any time.
(5) The recovery time of machine breakdown is known.
(6) The number of historical average deviations in the workshop server is greater than r.

4.2. Objective functions

(1) Minimising the total flow time of all the assigned operations (C_{sum}):

$$Minf_1 = C_{sum} = \sum_{k=1}^{m} t^{idle}_k + \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n_i}\left(t^p_{ijk} + t^c_{ijk}\right)\cdot x_{ijk} \tag{1}$$

(2) Minimising the total workload of all machines (WL^{all}):

$$Minf_2 = WL^{all} = \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n_i}\left(t^p_{ijk} + t^c_{ijk}\right)\cdot x_{ijk} \tag{2}$$

(3) Minimising the total energy consumption (E^{all}).

$$\begin{aligned} Minf_3 &= E^{all} \\ &= \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n_i} t^p_{ijk}\cdot P^p_k\cdot x_{ijk} + \sum_{k=1}^{m} t^{idle}_k\cdot p^{idle}_k + \sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n_i} t^c_{ijk}\cdot P^c_{ijk}\cdot x_{ijk} \end{aligned} \tag{3}$$

Subject to:

$$C_{ijk} - C_{i,j-1,k} \geq t^p_{ijk} + t^c_{ijk} i = 1, 2, \cdots, n; j = 2, 3, \cdots, n_i \tag{4}$$

$$\sum_{k=1}^{m} x_{ijk} = 1, \forall i, j \tag{5}$$

Eqs. (1)-(3) give the objective functions. Inequity (4) ensures sequence constraints on the operations. Eq. (5) guarantees that an operation is finally assigned to a machine.

5. IHA-based solve method for R-DTFJSS

The R-DTFJSS method based on the IHA is developed in this section to increase production efficiency and reduce energy consumption in the manufacturing process. By using the IHA, the operation can be assigned in real-time.

5.1. The implementation process of R-DTFJSS

The R-DTFJSS consists of two layers: the PS layer and the RS layer. The PS layer divides the production horizon into multiple time windows based on the time span ΔT, and generates a PS at the initial time of each time window to provide a basis for the RS. The RS layer carries out RS at each time t within the corresponding T_n through real-time interaction between PW and VW. In this paper, the time unit of ΔT is defined by eq. (6).

$$\Delta T = \left\lceil \frac{\max C_i}{w} \right\rceil i = 1, 2, \cdots, n \tag{6}$$

The C_i represents the completion time of J_i when all operations are assigned to the appropriate machine using the IHA of the PS layer at the initial time. The w denotes the parameter, which is used to adjust ΔT. If w is larger, it means that ΔT is smaller, and the frequency of PS generation is higher. Therefore, the RS system requires higher rapidity and stability. On the other hand, if w is smaller, it means ΔT is larger, and the frequency of PS generation is lower. Therefore, the RS system requires lower rapidity and stability. Fig. 2 shows the implementation process of R-DTFJSS. The specific implementation steps are shown as follows:

Step 1: At the initial time T_n, the IHA is used to assign all operations

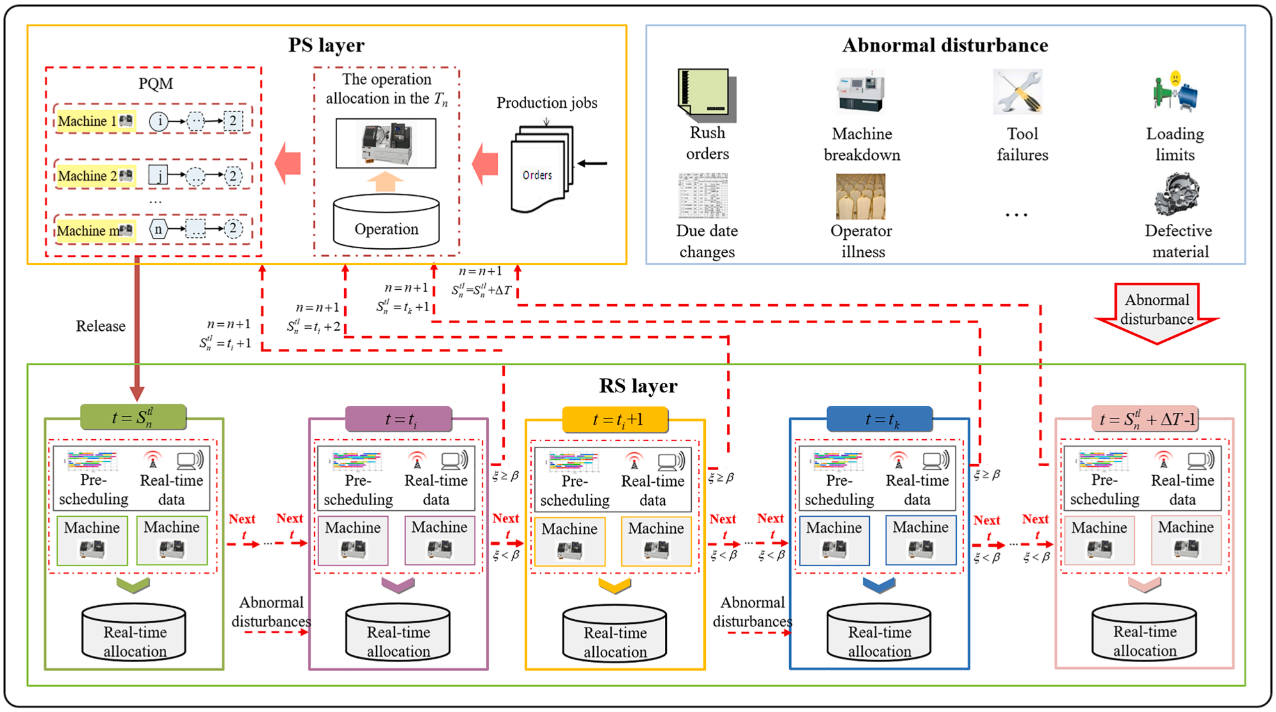

Fig. 2. The implementation process of R-DTFJSS

that can start processing within T_n to the appropriate machine. These operations are added to the corresponding PQM. Here, $T_n = [S_n^{tl}, S_n^{tl} + \Delta T)$. Thus, if the operation allocation is carried out in the T_1, then n=1, $S_1^{tl} = 0$ and$T_1 = [0, \Delta T)$. With the above method, the PS in T_n can be generated.

Step 2: In the RS within T_n, the IHA is used to assign the MOs to optimal machines at each time t according to the PS result within T_n and real-time data of the PW.

Step 2.1: If no abnormal disturbance occurs during the RS, go to Step 1 after the RS at time t ($t = S_n^{tl} + \Delta T - 1$) is complete. In this case, n=n+1, $S_n^{tl} = S_n^{tl} + \Delta T$.

Step 2.2: If some abnormal disturbances occur during the RS, the PS evaluation will be triggered when the RS is finished at each time t after the occurrence of the first abnormal disturbance. The PS evaluation can be implemented in the RS simulation process of the VW to determine whether the current PS result is still available. The detailed PS evaluation process is as follows.

The average deviation between PS and RS is proposed to evaluate the impact of abnormal disturbances on PS. The average deviation is set as ξ^t, which is shown as follows:

$$\xi = \frac{\sum_{k=1}^{m} \left| C^t_{pijk} - C^t_{Rijk} \right| \Big/ C^t_{pijk}}{g^t} \quad i \in [1, n] j \in [1, n_i] \tag{7}$$

According to the historical data of the workshop, β is set as the consistency between PS and RS. Therefore, r historical average deviations are randomly selected in the workshop server at the initial time, and the average value of these deviations is taken as the value of β. Because the historical average deviation is never zero, the value of β is never zero. In addition, the closer the value of β is to zero, the higher the consistency between PS and RS. If $\xi \geq \beta$, it indicates that the abnormal disturbance has a great impact on the PS. If $\xi < \beta$, it indicates that the abnormal disturbance has a small impact on the PS.

Therefore, based on the above analysis, after the PS evaluation at time t, the following method can be used to judge whether the RS can continue in the current time window.

(1) If $\xi \geq \beta$, go to Step 1. In this case, n=n+1, $S_n^{tl} = t + 1$.
(2) If $\xi < \beta$, the RS will continue at the next time t. Here, t=t+1. when $t = S_n^{tl} + \Delta T - 1$, the RS within T_n complete and then go to Step 1. In this case, n=n+1, $S_n^{tl} = S_n^{tl} + \Delta T$.

Step 3: Repeat the above steps until the R-DTFJSS ends.

5.2. Mathematical model of the IHA

According to the analysis of the R-DTFJSS implementation process, it can be seen that the R-DTFJSS problem is a multi-step operation allocation problem. To build the mathematical model of the IHA, we assume that p operations need to be assigned to q machines at a time. Therefore, the mathematical model of the IHA is established as follows:

$$\min z = \sum_{s=1}^{p} \sum_{h=1}^{q} L_{sh} x_{sh} = (L \cdot X), s \in J, h \in M \tag{8}$$

Subject to:

$$\sum_{s=1}^{p} x_{sh} = 1 h = 1, 2, \cdots, q \tag{9}$$

$$\sum_{h=1}^{q} x_{sh} = 1 s = 1, 2, \cdots, p \tag{10}$$

$$L = \begin{bmatrix} L_{11} & \cdots & L_{1q} \\ \cdots & L_{sh} & \cdots \\ L_{p1} & \cdots & L_{pq} \end{bmatrix} \tag{11}$$

Eq. (8) minimises the total cost z. Eq. (9) indicates that an operation is finally assigned to a machine. Eq. (10) shows that a machine processes only one operation at a time. Eq. (11) gives the cost matrix L ($L = (L_{sh})_{p \times q}$).

In the IHA, as long as the cost matrix L is established, the optimal solution and the minimum total cost can be obtained by solving the IHA. Thus, how to set the cost matrix L is an issue that should be considered. To get the cost matrix L, we have to know the element L_{sh} of the cost matrix L. Since the total cost in this paper is related to three optimisation objectives, L_{sh} should be composed of three components.

To minimise the f_1 (C_{sum}), let

$$L_{sh}^1 = t_{sh}^p + t_{sh}^c + t_{sh}^{idle} + C_h^s \tag{12}$$

To minimise the f_2 (WL^{all}), let

$$L_{sh}^2 = t_{sh}^p + t_{sh}^c + WL_h^s \tag{13}$$

To minimise the f_3 (E^{all}), let

$$L_{sh}^3 = t_{sh}^p \cdot p_h^p + t_{sh}^c \cdot p_{sh}^c + t_{sh}^{idle} \cdot p_h^{idle} + E_h^s \tag{14}$$

Here, if the h^{th} machine cannot process the s^{th} operation, set the t_{sh}^p and t_{sh}^c to a sufficiently large number ε. Because the dimensions of L_{sh}^1, L_{sh}^2 and L_{sh}^3 are inconsistent, normalisation processing is needed. The normalisation process adopts Eq. (15).

$$X' = \frac{9.9 \times (X - \min X)}{\max X - \min X} + 0.1 \tag{15}$$

In Eq. (15), the X represents the original data and the X' represents the normalised value. In this paper, the X stands for L_{sh}^1, L_{sh}^2 and L_{sh}^3, and the X' stands for L_{sh}^{nor1}, L_{sh}^{nor2} and L_{sh}^{nor3}. Therefore, the normalised values L_{sh}^{nor1}, L_{sh}^{nor2} and L_{sh}^{nor3} of data L_{sh}^1, L_{sh}^2 and L_{sh}^3 can be obtained, respectively.

Based on the above analysis, L_{sh} can be obtained by the weighted sum of L_{sh}^{nor1}, L_{sh}^{nor2} and L_{sh}^{nor3}, as shown as follows:

$$L_{sh} = \omega_1 \cdot L_{sh}^{nor1} + \omega_2 \cdot L_{sh}^{nor2} + \omega_3 \cdot L_{sh}^{nor3} \tag{16}$$

Subject to:

$$\omega_1 + \omega_2 + \omega_3 = 1,\ 0 \le \omega_1, \omega_2, \omega_3 \le 1 \tag{17}$$

In Eq. (16), ω_1,ω_2 and ω_3 are weight coefficients for L_{sh}^{nor1}, L_{sh}^{nor2} and L_{sh}^{nor3} respectively. Therefore, the cost matrix L of operation allocation can be obtained through the above method.

5.3. IHA-based R-DTFJSS method

Based on the implementation process of R-DTFJSS and the mathematical model of the IHA, an IHA-based R-DTFJSS method is designed. The detailed implementation of the method is shown as follows.

5.3.1. PS layer

At the initial time of T_n, the PS layer generates a PS within T_n to guide the execution of RS. Fig. 3 shows the detailed implementation process of the PS within T_n. The specific implementation steps are shown as follows.

Step 1: According to the RS results of T_{n-1}, the first unassigned operation of all jobs is put into the PS operation pool. All operations in the PS operation pool are numbered according to the size of the job number. Therefore, the sequence number of each operation in the PS operation pool can be known.

Step 2: Since the operations in the PS operation pool need to be assigned to the machines, let the number of operations in the PS operation pool be equal to p and the number of machines in the workshop be equal to q.

Step 3: Based on Step 1 and Step 2, the cost matrix L of operation allocation is established according to the method in Section 5.2.

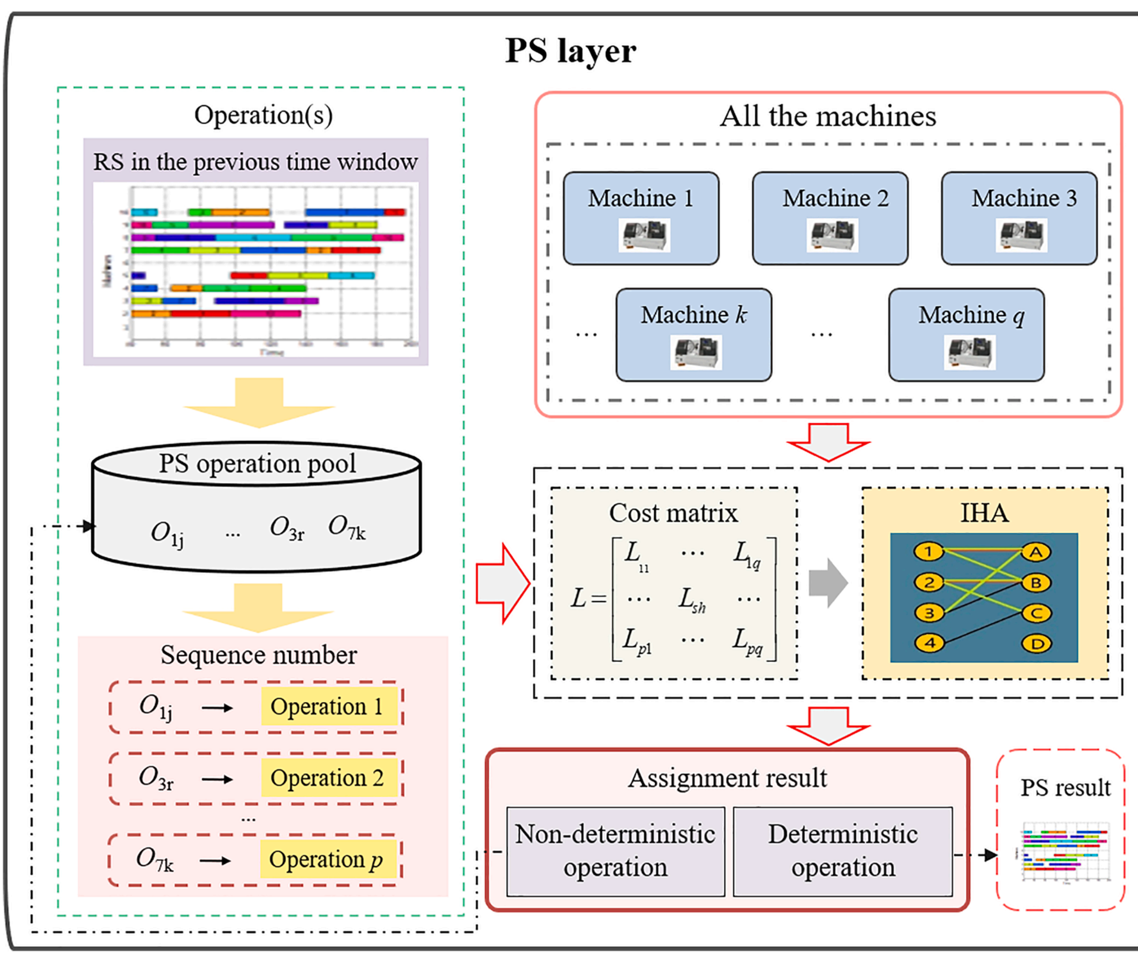

Fig. 3. The operation allocation process in the PS layer

Step 4: The cost matrix L is solved by the method in Section 5.4, and the optimal solution is obtained.

Step 5: Based on the optimal solution, some operations can be assigned to the machines. Because these allocation results may not be optimal, only one operation is eventually added to the PQM, called the deterministic operation. The remaining operations that are not added to the PQM are called non-deterministic operations and are put back into the PS operation pool. The selection rule for deterministic operation is shown in Fig. 4.

Step 6: Repeat the above process until all the operations that can start processing within T_n have been allocated.

5.3.2. RS layer

In the PS layer, the PS result of T_n is obtained. Therefore, in the RS layer, RS is carried out at each time t in the T_n according to the PS results and the real-time manufacturing data feedback by the PW. Fig. 5 describes in detail the RS process of each time t. The specific implementation steps are shown as follows.

Step 1: Based on the PS results and real-time data of the PW, the AM is added to the RS machine pool. Then these machines are numbered. Thus, the sequence number of each AM in the RS machine pool can be obtained.

Step 2: The MOs are added to the RS operation pool based on the PQM and real-time manufacturing information. Then these operations are numbered. The numbering method is the same as Step 1 in Section 5.3.1. Here, the selection of MOs should meet two requirements:

(1) The sequential constraints of the operation in the PQM, i.e., the latter operation can be considered as the MOs only after the previous operation is finished. The special case here is that if an operation is assigned to M_a in the PS and M_b in RS, the operation is considered finished even if the operation is not finished in RS.
(2) The sequence constraints of the operation in a job.

Step 3: Since the operations in the RS operation pool need to be assigned to the machines in the RS machine pool, let the number of operations in the RS operation pool be equal to p and the number of machines in the RS machine pool be equal to q.

Step 4: According to the method in Section 5.2, the cost matrix L of operation allocation is established.

Step 5: The cost matrix L is solved by the method in Section 5.4, and the optimal solution is obtained.

Step 6: The MOs are assigned to the optimal machines based on the optimal solution. There is a special case here: when an abnormal disturbance occurs, an operation on M_c in PS is allocated to other machines in RS. If the completion time of the operation on another machine is greater than or equal to the completion time of the operation on M_c, the operation will not be allocated. Return to Step 2 and take the operation out of the RS operation pool.

5.4. IHA solution

The Hungarian algorithm is an effective optimisation method to solve assignment problems. In applying the Hungarian algorithm, the cost matrix is required to have the same number of rows and columns. However, for the R-DTFJSS problem proposed in this paper, the cost matrix L usually does not meet this requirement. Therefore, it is necessary to develop an IHA, that is, to improve the cost matrix L and obtain a new cost matrix with an equal number of rows and columns. The specific improvement method of cost matrix L is as follows.

(1) In each operation allocation process, if $p<q$, the virtual operations are added to the allocation process. The number of virtual operations added is equal to (q-p). The cutting power, preparation time and cutting time of the virtual operation on each machine should be a large enough number ε. Therefore, the cost matrix L is improved to a q-order square matrix.
(2) In each operation allocation process, if $q<p$, virtual machines are added to the allocation process. The number of virtual machines added is equal to (p-q). The cutting power, preparation time and cutting time of each operation on the virtual machine should be a large enough number ε. Therefore, the cost matrix L is improved to a p -order square matrix.

```
//The selection rule for deterministic operation
Input: Optimal solution of operation allocation.
Start
  For operations that are ready to be assigned to machines
  If there are more than one operation
     {Select the operation with the minimum processing time (preparation time plus cutting time)
      on its optional machine and place these operations into the TOP
     If there are more than one operations in the TOP
        {The operation with the most remaining operations is left in the TOP and the other operations
         are taken out
        If there are still more than one operations in the TOP
           {The operation with earliest completion is left in the TOP and the other operations are
            taken out
           If there are still more than one operations in the TOP
              {The operation with the minimum workload on its machine is left in the TOP and the
               other operations are taken out
              If there are still more than one operations in the TOP
                 {Select one operation randomly to left in the TOP and the operation in the TOP is
                  a deterministic operation}
              Else the operation in the TOP is a deterministic operation}
           Else the operation in the TOP is a deterministic operation}
        Else the operation in the TOP is a deterministic operation}
     Else the operation in the TOP is a deterministic operation}
  Else the operation assigned is a deterministic operation
End
Output: A deterministic operation.
```

Fig. 4. The selection rule for deterministic operation

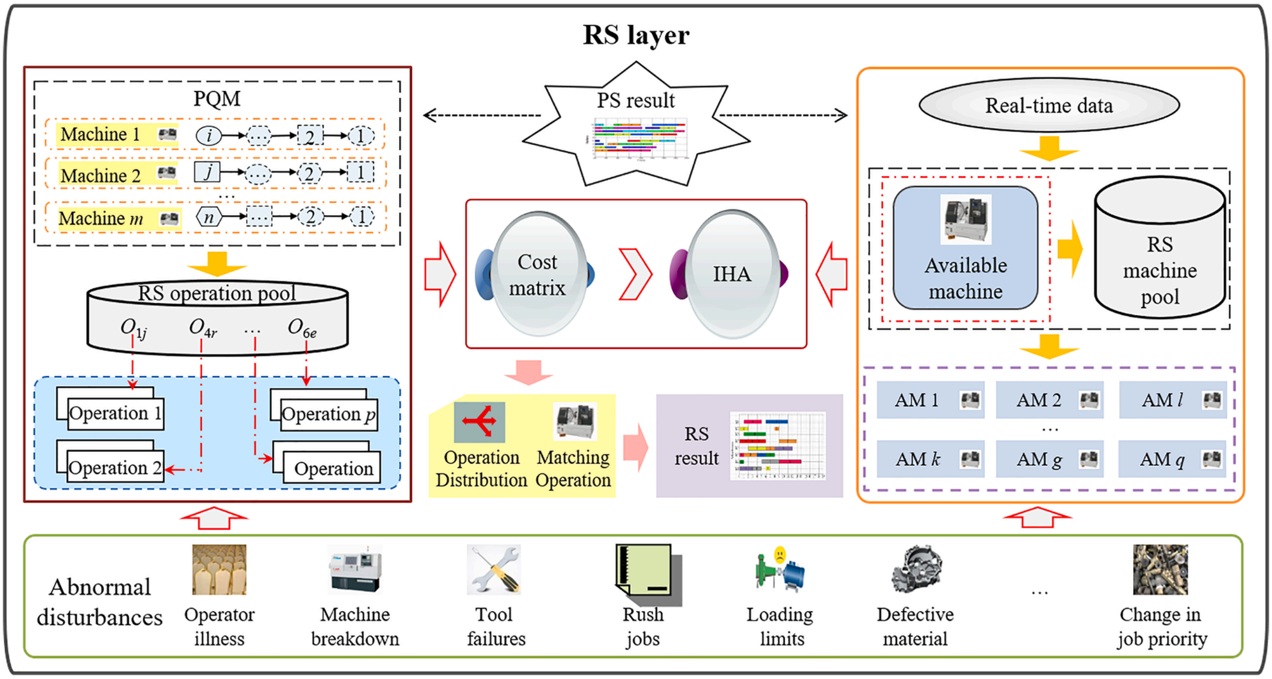

Fig. 5. The RS process in the RS layer

Thus, through the above method, the cost matrix L can be improved to a k-order square matrix L_1, where $k = \max(p,q)$. To obtain the optimal results, square matrix L_1 needs to be transformed, and the basic principle used is described as follows: for each element in any row and column of the square matrix L_1, add or subtract the same constant number to get a new square matrix, whose new square matrix has the same optimal solution as the original cost matrix L_1 [75]. Based on this basic principle, this paper summarises the solution of the IHA, as shown in Fig. 6.

6. Case simulation

This section presents an industrial application scenario to illustrate the advantages of the proposed R-DTFJSS method compared with the traditional scheduling method.

6.1. Case description

Here, the performance of the proposed R-DTFJSS method is illustrated through an industrial case of a cooperative enterprise. The cooperation enterprise is located in Xi'an, Shaanxi Province, China, which is a typical discrete manufacturing enterprise that mainly produces various pump bodies. Through our one-month observation, although the enterprise has collected the data of manufacturing resources in the manufacturing process, the collected data have not been fully interconnected. Therefore, real-time manufacturing data cannot be used accurately in the manufacturing process. In addition, the enterprise adopts the TDSM, namely an event-driven rescheduling strategy. In this dynamic scheduling strategy, the frequent response to abnormal disturbances leads to poor system stability. Therefore, it is urgent for enterprises to adopt the proposed R-DTFJSS method to manage and control the manufacturing process to ensure smooth production.

To prove the superiority of the R-DTFJSS method, the DT workshop is established in cooperative enterprises using IIoT, edge computing and DT technology. As shown in Fig. 7, the lower part is each component of the PW, and the upper part is the schematic interface of the VW. The PW mainly consists of a manufacturing area containing various intelligent

//Algorithm for IHA
Input: a k-order square matrix L_1.
Start
Step 1. Each row of square matrix L_1 minus the minimum of that row and get a new square matrix L_2.
Step 2. Each column of square matrix L_2 minus the minimum of that column and get a new square matrix L_3.
Step 3. Cover all the zero elements in square matrix L_3 with least straight lines. If all zero elements are covered by a straight line, go to *Step* 5; otherwise, go to *Step* 4.
Step 4. Find the minimum of the elements that are not covered by lines. All the elements that are not covered by lines minus this minimum and the elements that are covered by two lines plus this minimum. Thus, square matrix L_4 can be obtained. Set L_1=L_4 and return to *Step* 1.
Step 5. According to the distribution of zero element in square matrix L_3, the optimal result can be obtained.
END
Output: The optimal solution.

Fig. 6. Algorithm for IHA

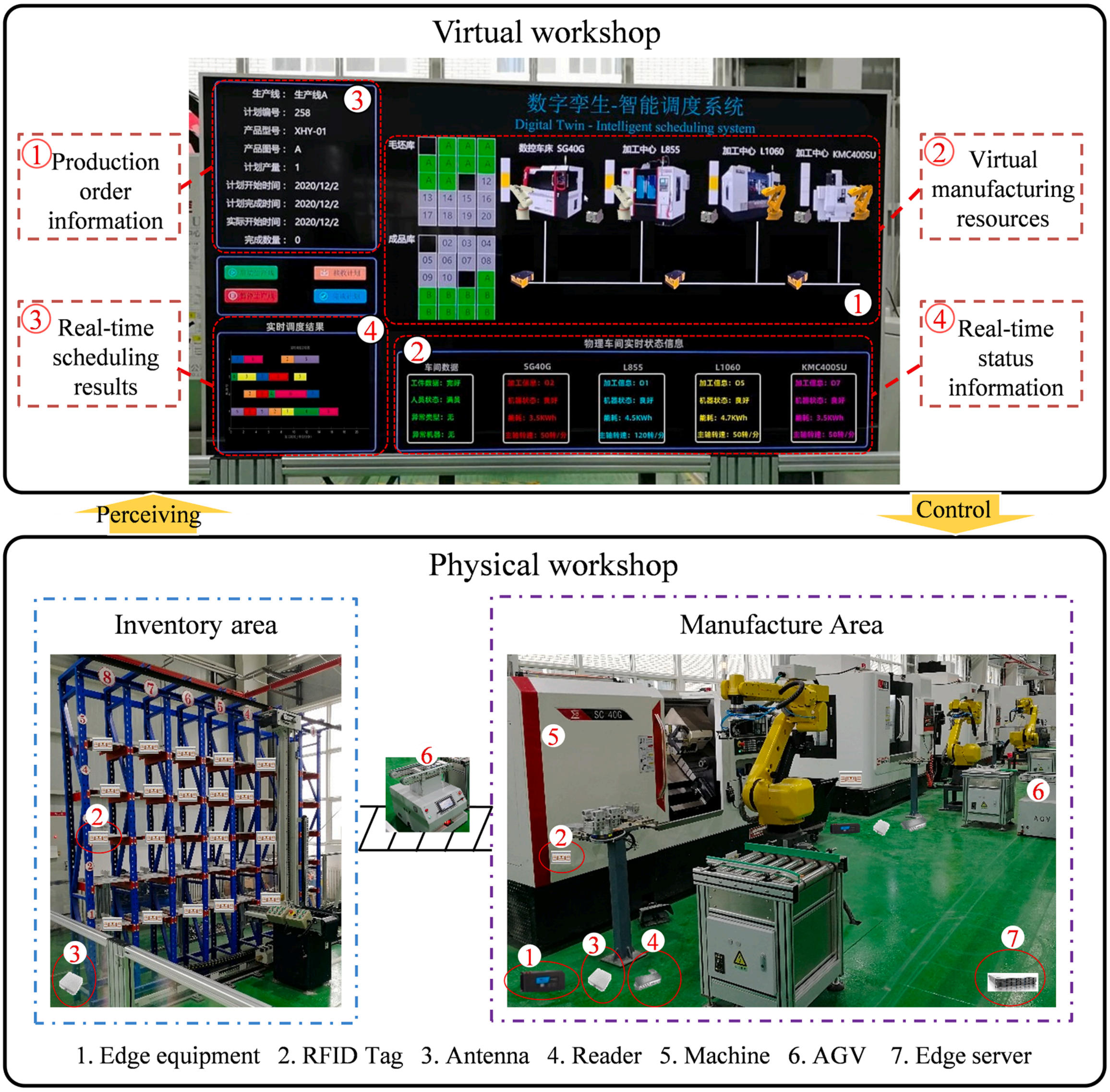

Fig. 7. Schematic diagram of DT workshop

manufacturing resources and an inventory area containing various shelves. In these two areas, intelligent machines, automatic guided vehicles, RFID, edge equipment and smart sensors are arranged to realise real-time manufacturing data collection and processing. The VW comprises production order information, virtual manufacturing resources, RS results, and real-time status information. Through the real-time interaction between PW and VW, the VW simulates the manufacturing process in real-time and sends the RS results to the PW for execution.

6.2. Simulation experiment

Based on the description of the DT workshop, we use a simulation experiment to illustrate the advantages of the R-DTFJSS method. At present, it is pointed out in some literature that the complexity and difficulty of dynamic scheduling can be illustrated if there are more than 6 machines in the workshop [76]. Thus, we use the data of the simulation experiment in Kacem's paper [77] as the original data, which is a static scheduling problem with 8 jobs multiply 8 machines. Since Kacem's paper only gives the processing time of the operation on the machine, to be consistent with the variables in the R-DTFJSS model, the processing time is divided into preparation time and cutting time. In addition, because the R-DTFJSS method needs to be compared with TDSMs in a dynamic environment, J_9 is taken as a rush order. Table 3 presents detailed simulation experimental data. Each set of data ($C/V/B$) below row O_{ij} and column M_k means that if O_{ij} is assigned to M_k, the preparation time is C, the cutting time is V and the cutting power is B. The idle power and preparation power of M_k are shown in Table 4. These data are derived from the actual machine power in production. The units of time and power are defined as an hour (h) and a kW respectively.

In the simulation experiment, R-DTFJSS is completed through the following four steps.

(1) The parameters of the R-DTFJSS method are set. Here, w, ε and β are 4, 99, and 0.15, respectively. The weight coefficients ω_1, ω_2 and ω_3 are set to 1/3.
(2) At the beginning of each T_n, the IHA is used to assign all operations that can start processing within corresponding T_n to the appropriate machine (refer to Section 5.1 and Section 5.3.1). Therefore, the PS results within T_n are generated in a static environment. These results determine which operations can be assigned during RS within corresponding T_n.
(3) According to the PS results within T_n, based on the real-time status information of the PW, the MOs are assigned to the

Table 3
The instance of the R-DTFJSS

Jobs	Operations	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
J_1	O_{11}	1/4/2.5	1/2/3.8	2/3/3.8	1/2/1.8	1/2/1.7	-	2/8/1.8	1/8/2.0
	O_{12}	4/6/2.2	-	1/4/3.1	4/4/2.1	1/2/1.9	4/5/1.7	1/8/2.1	3/3/2.1
	O_{13}	-	2/8/2.7	-	2/3/1.9	3/3/2.1	1/1/2.3	2/2/1.9	2/3/2.4
J_2	O_{21}	2/3/2.2	3/4/3.2	1/2/3.3	4/5/2.2	3/5/1.9	-	2/7/2.3	-
	O_{22}	-	4/4/3.5	2/3/3.2	1/1/1.7	2/4/2.1	2/5/1.9	5/5/2.1	4/5/2.8
	O_{23}	-	4/6/3.1	-	2/3/2.5	3/3/2.3	1/3/2.3	0.3/0.7/1.8	3/4/3.0
	O_{24}	5/5/2.4	4/4/3.3	2/7/3.4	3/3/1.8	2/2/1.8	3/4/2.2	-	-
J_3	O_{31}	3/7/2.6	-	-	1/6/2.2	2/4/1.7	2/3/2.6	1/1/1.9	2/2/2.2
	O_{32}	-	4/6/3.3	3/3/3.1	2/2/1.9	3/5/2.2	3/6/2.7	2/8/2.1	-
	O_{33}	0.3/0.7/2.9	2/2/2.7	2/3/3.8	2/4/2.2	-	2/8/2.4	-	2/5/2.4
J_4	O_{41}	2/1/2.1	0.2/0.8/2.9	3/3/3.1	2/3/1.7	4/5/2.3	2/5/2.3	4/4/2.2	2/2/2.3
	O_{42}	5/7/2.7	4/7/2.8	4/3/2.9	4/4/2.3	5/5/1.7	1/4/1.7	3/3/1.9	1/8/2.0
	O_{43}	2/2/2.2	3/3/2.7	1/1/3.1	5/5/2.1	1/2/2.2	1/8/2.7	2/3/2.2	1/6/2.8
J_5	O_{51}	1/2/2.2	1/5/3.2	2/5/3.3	3/5/1.6	4/5/2.5	-	4/6/2.3	-
	O_{52}	5/5/2.3	-	2/5/3.2	2/2/2.7	2/7/2.1	2/6/2.3	2/4/2.1	-
	O_{53}	-	5/4/3.6	4/4/3.6	3/4/2.8	1/3/2.6	1/1/1.8	3/4/2.4	-
	O_{54}	5/6/2.8	3/6/3.5	-	3/3/2.2	2/5/2.2	2/3/2.2	1/2/1.8	1/5/2.3
J_6	O_{61}	3/3/2.9	3/4/3.5	0.2/0.8/3.2	1/3/1.9	3/3/1.9	3/6/2.1	-	5/5/2.5
	O_{62}	4/7/3.2	-	4/5/3.2	3/6/2.4	2/7/2.4	3/4/2.3	2/4/2.0	2/2/2.3
	O_{63}	2/8/4.6	1/4/2.7	2/7/3.1	5/5/1.7	3/8/2.2	-	3/7/1.8	-
J_7	O_{71}	1/4/2.4	1/3/3.6	1/1/3.7	2/4/2.2	2/5/1.8	-	5/5/2.1	-
	O_{72}	-	5/4/3.1	-	4/5/2.3	3/8/2.3	3/6/2.1	1/9/2.4	1/4/2.9
	O_{73}	-	4/4/2.6	5/4/3.1	1/2/1.6	4/4/2.6	3/3/2.0	-	5/5/2.2
J_8	O_{81}	1/1/2.2	1/7/2.9	2/3/3.2	2/7/2.5	-	2/2/2.7	-	4/6/2.3
	O_{82}	3/4/2.9	2/2/2.7	2/5/3.5	2/6/2.3	4/5/2.9	-	3/7/1.9	-
	O_{83}	5/4/3.2	2/7/3.9	-	3/5/3.1	2/3/2.1	1/5/1.9	3/4/1.8	0.4/0.6/2.8
	O_{84}	3/6/2.6	-	1/2/3.9	3/4/1.9	0.5/0.5/2.7	2/3/1.8	4/4/2.2	-
J_9	O_{91}	2/7/3.2	1/4/3.5	2/3/3.8	2/2/2.1	2/7/1.9	1/4/1.7	2/2/2.1	1/2/2.3
	O_{92}	-	2/3/4.5	3/8/3.9	-	1/4/2.2	2/4/2.1	1/2/1.7	2/4/2.2
	O_{93}	2/3/2.7	3/5/2.8	2/6/3.4	4/5/2.3	1/1/1.8	2/2/2.2	2/3/2.3	-

Table 4
Idle power and preparation power

M_k	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
Idle power [kW]	1.3	1.8	2.1	0.9	0.7	0.9	0.8	1.1
preparation power [kW]	1.9	2.5	2.9	1.4	1.6	1.5	1.6	1.9

optimal machine at each time t ($t \in T_n$) in the VW by IHA (refer to Section 5.1 and Section 5.3.2).

(4) By repeating the above process, all operations can be allocated in real-time. When abnormal disturbances occur in the manufacturing process, the adverse effects can be eliminated through the real-time interaction between PW and VW.

6.3. Performance analysis of the R-DTFJSS method

To verify the performance of the R-DTFJSS method, we compare it with the ESSMs and TDSMs.

6.3.1. Comparison with the ESSMs

To illustrate the advantages of the R-DTFJSS method in a static environment, we compare the R-DTFJSS method with the ESSMs. The ESSMs include PSO+SA proposed by Xia and Wu [78], P-DABC employed by Li et al. [79], TL-HGAPSO developed by Huang et al. [80], and HLO-PSO presented by Ding and Gu [81]. Although these ESSMs use the simulation data proposed by Kacem's paper [77] for FJSS, these methods only take makespan and workload as optimisation objectives and do not consider the optimisation of total energy consumption and total flow time. Thus, we calculate the energy consumption and the total flow time according to the Gantt charts obtained by the ESSMs. Table 5 shows the optimisation results of the R-DTFJSS method and the ESSMs.

According to the results shown in Table 5, the R-DTFJSS method is not always superior to the ESSMs in optimising f_1. The reason is that in a static environment, the R-DTFJSS method always preferentially assigns an operation to its optional machine with the minimum processing time

Table 5
The optimisation results of the R-DTFJSS method and ESSMs

Objectives	PSO+SA	P-DABC	TL-HGAPSO	HLO-PSO	R-DTFJSS
f_1 [h]	99	99	100	102	101
f_2 [h]	75	77	78	80	73
f_3 [kWh]	193.29	195.44	206.64	205.24	184.79

(preparation time plus cutting time), which tends to reduce the total workload of machines. But on the other hand, it may lead to long waiting queues in specific machines, which results in a long flow time for all the operations. However, the R-DTFJSS method can get a better value of f_1 compared with the HLO-PSO method. For the optimisation of f_2 and f_3, the R-DTFJSS method completely outperforms the ESSMs. For example, the value f_2 obtained by the R-DTFJSS method is 73h, while the minimum value f_2 obtained by the ESSMs is 75h. Therefore, it can be concluded that the R-DTFJSS method can improve the value f_2 by 2.6%. In addition, the value f_3 obtained by the R-DTFJSS method is 184.79kWh, which can still achieve a maximum of 10.6% and a minimum of 4.4% improvement compared with the ESSMs. Therefore, it can be seen from the above analysis that the performance of the R-DTFJSS method is better than ESSMs in a static environment.

6.3.2. Comparison with the TDSMs

To further illustrate the advantages of the R-DTFJSS method in a dynamic environment, we compare the R-DTFJSS method with the TDSMs, including the right-shift rescheduling method, periodic rescheduling method, event-driven rescheduling method and heuristic algorithm. In the right-shift rescheduling method, the static scheduling result obtained by Fang and Xi [82] is used as the initial scheduling. In the periodic rescheduling method, we use the NSGA-II algorithm to reschedule every 5h [83]. In the event-driven rescheduling method, we still use the NSGA-II algorithm for rescheduling when abnormal disturbances occur [83]. In this NSGA-II algorithm, the size of the population, the maximum number of iterations, mutation probability and crossover probability are set as 100, 500, 0.1 and 0.8, respectively. The heuristic algorithm uses shortest processing time (SPT) and random assignment method (RAM) to assign the operation to the corresponding machine. In addition, at each dynamic scheduling point, the least work remaining (LWR), most work remaining (MWR) and random sorting method (RSM) are used to sort the operations on the machine. Based on the above heuristic algorithm, six dynamic scheduling methods are proposed, which are SPT+LWR, SPT+MWR, SPT+RSM, RAM+LWR, RAM +MWR and RAM +RSM.

Three test scenarios are given to compare the R-DTFJSS method with the TDSMs in a dynamic environment. Test scenario 1 mainly considers the occurrence of machine breakdowns. That is, M_2 and M_5 are faulty at time t_1 ($t_1 = 3$) and t_2 ($t_2 = 8$), and repaired at time t_3 ($t_3 = 7$) and t_4 ($t_4 = 13$), respectively. Test scenario 2 focuses on inserting a rush order, which J_9 is added to the manufacturing process at time t_5 ($t_5 = 6$). Test scenario 3 considers both the occurrence of machine breakdowns in test scenario 1 and the insertion of a rush order in test scenario 2. Figs. 8, 9, 10 show the optimisation results of various scheduling methods in three test scenarios, respectively. Compared with the TDSMs, the improvement of the R-DTFJSS method in the three scenarios is shown in Tables 6-8, respectively.

According to the data shown in Fig. 8 and Table 6, in test scenario 1, the optimisation value of the R-DTFJSS method for f_1 is 107h. In the TDSMs, the minimum value of f_1 is 109h. Therefore, compared with the TDSMs, the R-DTFJSS method can achieve a minimum 1.8% improvement. Although the value of f_2 obtained by the R-DTFJSS method is larger than that obtained by SPT+LWR, SPT+MWR and SPT+RSM, it can still achieve a maximum improvement of 6.0% compared with other TDSMs. In addition, compared with the TDSMs, the R-DTFJSS method can also get a better value of f_3, achieving a minimum and maximum improvement of 0.2% and 17.4%, respectively. Therefore, the R-DTFJSS method is generally superior to the TDSMs in test scenario 1.

According to the data shown in Fig. 9 and Table 7, in test scenario 2, the optimisation value of the R-DTFJSS method for f_1 is 107h. Compared with the best result of f_1 obtained by the TDSMs, the R-DTFJSS method can achieve a 4.5% improvement. Although the value of f_2 obtained by the R-DTFJSS method is not optimal compared with the heuristic algorithm based on SPT, the R-DTFJSS method can still achieve a maximum improvement of 10.9% compared with other TDSMs. For the optimisation of f_3, the value obtained by the R-DTFJSS method is 198.19 kWh, which is also completely superior to the results obtained by the TDSMs. Therefore, it can be said that the R-DTFJSS method is superior to the TDSMs in test scenario 2.

According to the data shown in Fig. 10 and Table 8, the R-DTFJSS method also outperforms the TDSMs in test scenario 3. For the optimisation of f_1, the R-DTFJSS method can achieve the maximum and minimum improvement of 21.3% and 5.9%, respectively. Compared with the TDSMs, although the optimisation of f_2 by the R-DTFJSS method is not always optimal, the optimisation of f_3 can still achieve maximum and minimum improvement of 21.2% and 3.9%, respectively.

From the simulation experiment of the above three test scenarios, it can be seen that although the value of f_2 obtained by the R-DTFJSS method is not superior to the value obtained by the heuristic algorithm based on SPT, the R-DTFJSS method can achieve better results than other TDSMs. Meanwhile, in the three test scenarios, the values of f_1 and f_3 obtained by the R-DTFJSS method are completely superior to those obtained by the TDSMs. Therefore, the R-DTFJSS method performs better than the TDSMs in a dynamic environment with abnormal disturbance.

6.3.3. *Comparison with the CPU time*

For real-world factories, CPU time is important in RS. If the CPU time is too long, the RS system cannot respond to abnormal disturbances timely. Table 9 shows the average CPU time of the R-DTFJSS method and the TDSMs under three test scenarios. According to Table 9, the average CPU time of the R-DTFJSS method is much shorter than the TDSMs in the running process of dynamic scheduling. Thus, it can be seen from the above analysis that compared with the TDSMs, the R-DTFJSS method has advantages not only in RS results but also in average CPU time.

The main differences between the R-DTFJSS method and the TDSMs are reflected in the following aspects.

1) The R-DTFJSS method uses DT, edge computing and IIoT technologies to conduct RS through real-time interaction between PW and VW. In this scheduling mode, based on the real-time data of the PW, RS results are generated in the VW, and the assignment result is feedback to the PW for execution. For the TDSMs, only the initial manufacturing information of the workshop is considered for operation allocation. Therefore, abnormal disturbances in the manufacturing process lead to frequent interruptions of the production system.

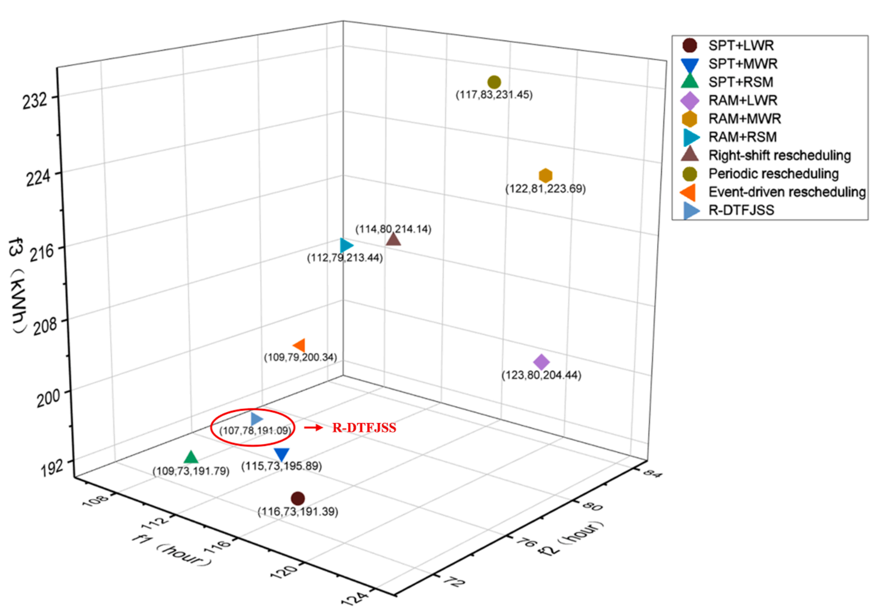

Fig. 8. The optimisation results of various scheduling methods in test scenario 1

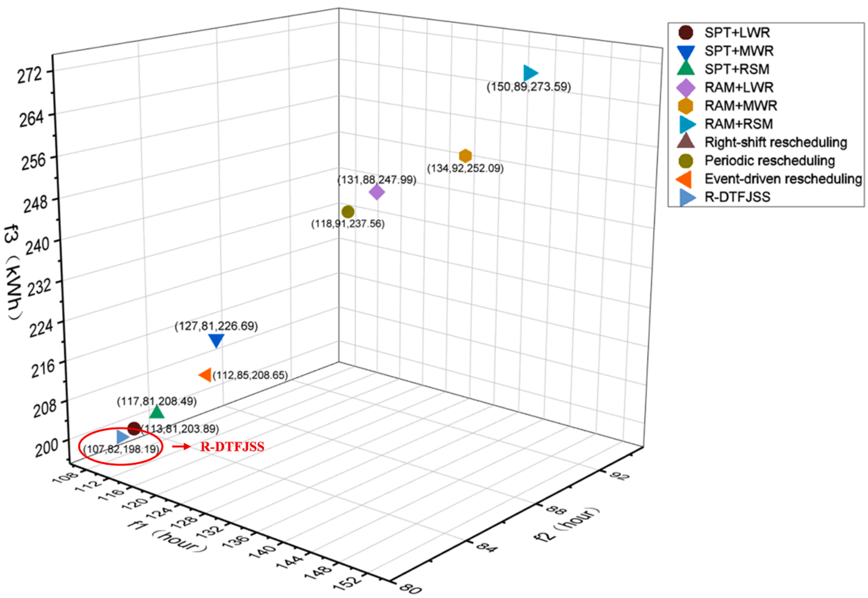

Fig. 9. The optimisation results of various scheduling methods in test scenario 2

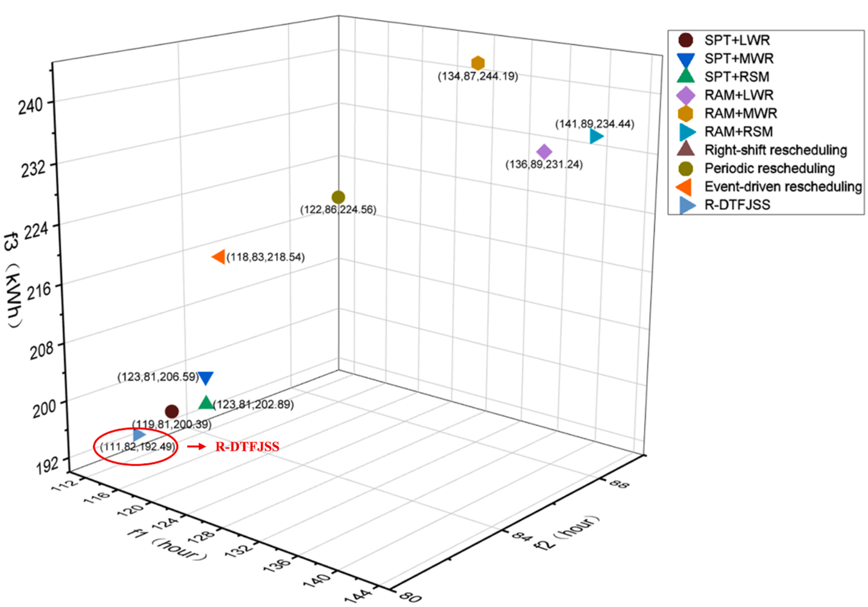

Fig. 10. The optimisation results of various scheduling methods in test scenario 3

Table 6
The improvement of R-DTFJSS method in test scenario 1

Scheduling methods	f_1 [h]	f_2 [h]	f_3 [kWh]
SPT+LWR	7.8%	-6.8%	0.2%
SPT+MWR	7.0%	-6.8%	2.5%
SPT+RSM	1.8%	-6.8%	0.4%
RAM+LWR	13.0%	2.5%	6.5%
RAM+MWR	12.3%	3.7%	14.6%
RAM+RSM	4.5%	1.3%	10.5%
Right-shift rescheduling	6.1%	2.5%	10.8%
Periodic rescheduling	8.5%	6.0%	17.4%
Event-driven rescheduling	1.8%	1.3%	4.6%

Table 7
The improvement of R-DTFJSS method in test scenario 2

Scheduling methods	f_1 [h]	f_2 [h]	f_3 [kWh]
SPT+LWR	5.3%	-1.2%	2.8%
SPT+MWR	15.7%	-1.2%	12.6%
SPT+RSM	8.5%	-1.2%	4.9%
RAM+LWR	18.3%	6.8%	20.1%
RAM+MWR	20.1%	10.9%	21.4%
RAM+RSM	28.7%	7.9%	27.6%
Right-shift rescheduling	-	-	-
Periodic rescheduling	9.3%	9.9%	16.6%
Event-driven rescheduling	4.5%	3.5%	5.0%

2) The R-DTFJSS method first generates the PS in the corresponding time window and then performs RS at each time t in this time window. For the TDSMs, a periodic rescheduling strategy or event-driven rescheduling strategy is generally used. Under these two rescheduling strategies, the response to abnormal disturbance may be too slow or frequent, resulting in unsustainable production.

Table 8
The improvement of R-DTFJSS method in test scenario 3

Scheduling methods	f_1 [h]	f_2 [h]	f_3 [kWh]
SPT+LWR	6.7%	-1.2%	3.9%
SPT+MWR	9.8%	-1.2%	6.8%
SPT+RSM	9.8%	-1.2%	5.1%
RAM+LWR	18.4%	7.9%	16.8%
RAM+MWR	17.2%	5.7%	21.2%
RAM+RSM	21.3%	7.9%	17.9%
Right-shift rescheduling	-	-	-
Periodic rescheduling	9.0%	4.7%	14.3%
Event-driven rescheduling	5.9%	1.2%	11.9%

Table 9
The average CPU time

Scheduling methods	average CPU time (s)
SPT+LWR	3.123
SPT+MWR	2.341
SPT+RSM	3.712
RAM+LWR	4.431
RAM+MWR	5.214
RAM+RSM	3.412
Right-shift rescheduling	10.231
Periodic rescheduling	20.412
Event-driven rescheduling	27.412
R-DTFJSS	0.184

7. Conclusions and future works

In intelligent manufacturing, abnormal disturbance of workshops seriously affects the smooth execution of production scheduling. Therefore, in the context of the rapid development of the new generation of information technology, it is necessary to develop a new RS mode to ensure efficient and sustainable production. This paper presents an IHA-based R-DTFJSS method with edge computing. Compared with the TDSMs, the R-DTFJSS method uses IHA to assign operations in real-time through real-time interaction between PW and VW, which reduces the scheduling computation complexity and improves the response-ability to abnormal disturbances.

The contribution of the paper mainly includes four aspects. Firstly, the overall framework of the R-DTFJSS based on edge computing is established, and the working mechanism of the R-DTFJSS is analysed. The proposed R-DTFJSS framework is the basis for solving the RS problem under abnormal disturbance. In this framework, the RS system can quickly detect and deal with the abnormal disturbance in the production process through real-time interaction between PW and VW to ensure stable and reliable production operation. Secondly, an RS strategy based on a variable time window is introduced to solve the R-DTFJSS considering frequent abnormal disturbances in the production process. In this strategy, according to the PS results and the real-time data of the PW, real-time operation allocation is carried out at each time t. In addition, the time window length can be dynamically adjusted according to the impact of abnormal disturbance on production. Thirdly, a real-time operation allocation method based on the IHA is proposed, i.e., at each time t, the operation is allocated to the optimal machine. In this method, the computational complexity can be significantly reduced. Meanwhile, computation time and complexity do not increase with the number of jobs and machines. Fourthly, the analysis of an industrial case proves that the proposed R-DTFJSS method is effective in a static environment and better than the TDSMs in a dynamic environment.

The limitation of this paper is that the proposed R-DTFJSS method does not consider logistics, machine maintenance and tool damage in production. In addition, the proposed framework and model lack a detailed description of DT and the relationship between edge computing and DT.

Future work can be studied in the following three aspects. The first aspect is to establish an RS model that is more consistent with the real-world manufacturing system and further promotes the application of the R-DTFJSS method in practice. The second aspect is to predict the abnormal disturbances in the DT workshop to guide the implementation of RS. The third aspect is to consider the logistics situation of raw materials or semi-finished products in the RS.

CRediT authorship contribution statement

Jin Wang: Writing – original draft, Methodology. **Yang Liu:** Supervision, Validation, Writing – review & editing. **Shan Ren:** Supervision, Validation, Writing – review & editing. **Chuang Wang:** Writing – review & editing. **Shuaiyin Ma:** Validation, Software.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The authors do not have permission to share data.

Acknowledgements

The work was supported by the Key Research and Development Program of Shaanxi (No. 2021GY-069) and the National Natural Science Foundation of China (No. 52005408).

References

[1] C. Qian, Y. Zhang, C. Jiang, S. Pan, Y. Rong, A real-time data-driven collaborative mechanism in fixed-position assembly systems for smart manufacturing, Robot. Comput. Integr. Manuf. 61 (2020), 101841, https://doi.org/10.1016/j.rcim.2019.101841.

[2] J. Wang, Y. Zhang, Y. Liu, N. Wu, Multiagent and bargaining-game-based real-time scheduling for internet of things-enabled flexible job shop, IEEE Internet Things J 6 (2019) 2518–2531, https://doi.org/10.1109/JIOT.2018.2871346.

[3] Y. Li, W. Huang, R. Wu, K. Guo, An improved artificial bee colony algorithm for solving multi-objective low-carbon flexible job shop scheduling problem, Appl. Soft Comput. J. (2020) 95, https://doi.org/10.1016/j.asoc.2020.106544.

[4] L. Li, K. Wang, An improved ant colony algorithm for multi-objective flexible job shop scheduling problem, in: Proc. 2009 IEEE Int. Conf. Autom. Logist. ICAL 2009, 2009, pp. 697–702, https://doi.org/10.1109/ICAL.2009.5262833.

[5] XY Li, L. Gao, A collaborative evolutionary algorithm for multi-objective flexible job shop scheduling problem, in: Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern, 2011, pp. 997–1002, https://doi.org/10.1109/ICSMC.2011.6083799.

[6] M. Saidi-Mehrabad, P. Fattahi, Flexible job shop scheduling with tabu search algorithms, Int. J. Adv. Manuf. Technol. 32 (2007) 563–570, https://doi.org/10.1007/s00170-005-0375-4.

[7] J.-C. CAI, L. WANG, Y.-P. XING, Research on flexible job shop scheduling problem based on improved genetic algorithm, destech trans, Eng. Technol. Res. (2017), https://doi.org/10.12783/dtetr/mdm2016/4877.

[8] G. Zhang, X. Shao, P. Li, L. Gao, An effective hybrid particle swarm optimisation algorithm for multi-objective flexible job-shop scheduling problem, Comput. Ind. Eng. 56 (2009) 1309–1318, https://doi.org/10.1016/j.cie.2008.07.021.

[9] M. Amiri, M. Zandieh, M. Yazdani, A. Bagheri, A variable neighbourhood search algorithm for the flexible job-shop scheduling problem, Int. J. Prod. Res. 48 (2010) 5671–5689, https://doi.org/10.1080/00207540903055743.

[10] FTS. Chan, T.C. Wong, L.Y. Chan, Flexible job-shop scheduling problem under resource constraints, Int. J. Prod. Res. 44 (2006) 2071–2089, https://doi.org/10.1080/00207540500386012.

[11] A. Farahani, H. Tohidi, M. Khalaj, A. Shoja, Partial flexible job shop scheduling considering preventive maintenance and priorities, WPOM-Working Pap, Oper. Manag. 11 (2020) 27, https://doi.org/10.4995/wpom.v11i2.14187.

[12] M. Ziaee, A heuristic algorithm for the distributed and flexible job-shop scheduling problem, J. Supercomput. 67 (2014) 69–83, https://doi.org/10.1007/s11227-013-0986-8.

[13] A. Baykasoğlu, F.S. Madenoğlu, A. Hamzadayı, Greedy randomised adaptive search for dynamic flexible job-shop scheduling, J. Manuf. Syst. 56 (2020) 425–451, https://doi.org/10.1016/j.jmsy.2020.06.005.

[14] F. Zhang, Y. Mei, S. Nguyen, M. Zhang, Evolving scheduling heuristics via genetic programming with feature selection in dynamic flexible job-shop scheduling, IEEE

Trans. Cybern. 51 (2021) 1797–1811, https://doi.org/10.1109/TCYB.2020.3024849.
[15] Z. Wang, J. Zhang, S. Yang, An improved particle swarm optimisation algorithm for dynamic job shop scheduling problems with random job arrivals, Swarm Evol. Comput. (2019) 51, https://doi.org/10.1016/j.swevo.2019.100594.
[16] Y. Liu, Y. Zhang, S. Ren, M. Yang, Y. Wang, D. Huisingh, How can smart technologies contribute to sustainable product lifecycle management? J. Clean. Prod. (2020) 249, https://doi.org/10.1016/j.jclepro.2019.119423.
[17] Z. Guo, Y. Zhang, X. Zhao, X. Song, CPS-based self-adaptive collaborative control for smart production-logistics systems, IEEE Trans. Cybern. 51 (2021) 188–198, https://doi.org/10.1109/TCYB.2020.2964301.
[18] M. Ghaleb, H. Zolfagharinia, S. Taghipour, Real-time production scheduling in the Industry-4.0 context: Addressing uncertainties in job arrivals and machine breakdowns, Comput. Oper. Res. (2020) 123, https://doi.org/10.1016/j.cor.2020.105031.
[19] Y. Li, W. Gu, M. Yuan, Y. Tang, Real-time data-driven dynamic scheduling for flexible job shop with insufficient transportation resources using hybrid deep Q network, Robot. Comput. Integr. Manuf. 74 (2022), 102283, https://doi.org/10.1016/j.rcim.2021.102283.
[20] S. Tian, T. Wang, L. Zhang, X. Wu, An energy-efficient scheduling approach for flexible job shop problem in an internet of manufacturing things environment, IEEE Access 7 (2019) 62695–62704, https://doi.org/10.1109/ACCESS.2019.2915948.
[21] J. Wang, J. Yang, Y. Zhang, S. Ren, Y. Liu, Infinitely repeated game based real-time scheduling for low-carbon flexible job shop considering multi-time periods, J. Clean. Prod. (2020) 247, https://doi.org/10.1016/j.jclepro.2019.119093.
[22] Y. Zhang, G. Zhang, J. Wang, S. Sun, S. Si, T. Yang, Real-time information capturing and integration framework of the internet of manufacturing things, Int. J. Comput. Integr. Manuf. 28 (2015) 811–822, https://doi.org/10.1080/0951192X.2014.900874.
[23] J. Wen, C. Ren, A.K. Sangaiah, Energy-efficient device-to-device edge computing network: an approach offloading both traffic and computation, IEEE Commun. Mag. 56 (2018) 96–102, https://doi.org/10.1109/MCOM.2018.17010543.
[24] Z. Yahouni, N. Mebarki, Z. Sari, Evaluation of a new decision-aid parameter for job shop scheduling under uncertainties, RAIRO - Oper. Res. 53 (2019) 593–608, https://doi.org/10.1051/ro/2017073.
[25] S. Luo, L. Zhang, Y. Fan, Dynamic multi-objective scheduling for flexible job shop by deep reinforcement learning, Comput. Ind. Eng. 159 (2021), 107489, https://doi.org/10.1016/j.cie.2021.107489.
[26] X.P. Yang, X.L. Gao, Optimisation of dynamic and multi-objective flexible job-shop scheduling based on parallel hybrid algorithm, Int. J. Simul. Model. 17 (2018) 724–733, https://doi.org/10.2507/IJSIMM17(4)CO19.
[27] M. Souier, A. Hassam, Z. Sari, Meta-heuristics for real-time routing selection in flexible manufacturing systems, in: 2010: pp. 221–248. https://doi.org/10.1007/978-1-84996-119-6_8.
[28] M. Ham, Y.H. Lee, S.H. Kim, Real-time scheduling of multi-stage flexible job shop floor, Int. J. Prod. Res. 49 (2011) 3715–3730, https://doi.org/10.1080/00207543.2010.492797.
[29] H. Zhu, M. Chen, Z. Zhang, D. Tang, An adaptive real-time scheduling method for flexible job shop scheduling problem with combined processing constraint, IEEE Access 7 (2019) 125113–125121, https://doi.org/10.1109/ACCESS.2019.2938548.
[30] L. Nie, L. Gao, P. Li, X. Li, A GEP-based reactive scheduling policies constructing approach for dynamic flexible job shop scheduling problem with job release dates, J. Intell. Manuf. 24 (2013) 763–774, https://doi.org/10.1007/s10845-012-0626-9.
[31] C.M. Harmonosky, S.F. Robohn, Real-time scheduling in computer integrated manufacturing: A review of recent research, Int. J. Comput. Integr. Manuf. 4 (1991) 331–340, https://doi.org/10.1080/09511929108944511.
[32] Y.L. Chang, R.S. Sullivan, Schedule generation in a dynamic job shop, Int. J. Prod. Res. 28 (1990) 65–74, https://doi.org/10.1080/00207549008942684.
[33] M.N. Karsiti, J.B. Cruz, J.H. Mulligan, Simulation studies of multilevel dynamic job shop scheduling using heuristic dispatching rules, J. Manuf. Syst. 11 (1992) 346–358, https://doi.org/10.1016/0278-6125(92)90063-L.
[34] U. Roy, X. Zhang, A heuristic approach to n/m job shop scheduling: Fuzzy dynamic scheduling algorithms, Prod. Plan. Control. 7 (1996) 299–311, https://doi.org/10.1080/09537289608930356.
[35] W.G. Bistline, S. Banerjee, A. Banerjee, RTSS: An interactive decision support system for solving real time scheduling problems considering customer and job priorities with schedule interruptions, Comput. Oper. Res. 25 (1998) 981–995, https://doi.org/10.1016/S0305-0548(97)00092-0.
[36] E. Kutanoglu, I. Sabuncuoglu, Experimental investigation of iterative simulation-based scheduling in a dynamic and stochastic job shop, J. Manuf. Syst. 20 (2001) 264–279, https://doi.org/10.1016/s0278-6125(01)80046-7.
[37] G.E. Vieira, J.W. Herrmann, E. Lin, Rescheduling manufacturing systems: A framework of strategies, policies, and methods, J. Sched. (2003) 39–62, https://doi.org/10.1023/A:1022235519958.
[38] H.C. Hwang, B.K. Choi, Workflow-based dynamic scheduling of job shop operations, Int. J. Comput. Integr. Manuf. 20 (2007) 557–566, https://doi.org/10.1080/09511920601024179.
[39] M. Zandieh, MA Adibi, Dynamic job shop scheduling using variable neighbourhood search, Int. J. Prod. Res. 48 (2010) 2449–2458, https://doi.org/10.1080/00207540802662896.
[40] H. Xiong, H. Fan, G. Li, G. Jiang, Research on steady-state simulation in dynamic job shop scheduling problem, Adv. Mech. Eng. 7 (2015) 1–11, https://doi.org/10.1177/1687814015604546.
[41] M.S. Lu, R. Romanowski, Multicontextual dispatching rules for job shops with dynamic job arrival, Int. J. Adv. Manuf. Technol. 67 (2013) 19–33, https://doi.org/10.1007/s00170-013-4765-8.
[42] A. Teymourifar, G. Ozturk, Z.K. Ozturk, O. Bahadir, Extracting new dispatching rules for multi-objective dynamic flexible job shop scheduling with limited buffer spaces, Cognit. Comput 12 (2020) 195–205, https://doi.org/10.1007/s12559-018-9595-4.
[43] V. Vinod, R. Sridharan, Simulation-based metamodels for scheduling a dynamic job shop with sequence-dependent setup times, Int. J. Prod. Res. 47 (2009) 1425–1447, https://doi.org/10.1080/00207540701486082.
[44] D. Tang, M. Dai, M.A. Salido, A. Giret, Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimisation, Comput. Ind. 81 (2016) 82–95, https://doi.org/10.1016/j.compind.2015.10.001.
[45] H. Zhang, Y.Q. Zhang, A discrete job-shop scheduling algorithm based on improved genetic algorithm, Int. J. Simul. Model. 19 (2020) 517–528, https://doi.org/10.2507/IJSIMM19-3-CO14.
[46] A. Rezaeipanah, M. Mojarad, Modeling the Scheduling Problem in Cellular Manufacturing Systems Using Genetic Algorithm as an Efficient Meta-Heuristic Approach, J. Artif. Intell. Technol. (2021), https://doi.org/10.37965/jait.2021.0018.
[47] R. Zhou, A.Y.C. Nee, H.P. Lee, Performance of an ant colony optimisation algorithm in dynamic job shop scheduling problems, Int. J. Prod. Res. 47 (2009) 2903–2920, https://doi.org/10.1080/00207540701644219.
[48] M.S. Lu, R. Romanowski, Multi-contextual ant colony optimisation of intermediate dynamic job shop problems, Int. J. Adv. Manuf. Technol. (2012) 667–681, https://doi.org/10.1007/s00170-011-3634-6.
[49] P. Lou, Q. Liu, Z. Zhou, H. Wang, S.X. Sun, Multi-agent-based proactive-reactive scheduling for a job shop, Int. J. Adv. Manuf. Technol. 59 (2012) 311–324, https://doi.org/10.1007/s00170-011-3482-4.
[50] L. Shi, G. Guo, X. Song, Multi-agent based dynamic scheduling optimisation of the sustainable hybrid flow shop in a ubiquitous environment, Int. J. Prod. Res. 59 (2021) 576–597, https://doi.org/10.1080/00207543.2019.1699671.
[51] L. Zhang, L. Gao, X. Li, A hybrid genetic algorithm and tabu search for a multi-objective dynamic job shop scheduling problem, Int. J. Prod. Res. (2013), https://doi.org/10.1080/00207543.2012.751509.
[52] S. Zhang, T.N. Wong, Flexible job-shop scheduling/rescheduling in dynamic environment: a hybrid MAS/ACO approach, Int. J. Prod. Res. 55 (2017) 3173–3196, https://doi.org/10.1080/00207543.2016.1267414.
[53] S. Dang, W. Tang, Real-Time Data Scheduling of Flexible Job in Papermaking Workshop Based on Deep Learning and Improved Fuzzy Algorithm, Mob. Inf. Syst (2021) 2021, https://doi.org/10.1155/2021/9930740.
[54] J. Wang, Y. Liu, S. Ren, C. Wang, W. Wang, Evolutionary game based real-time scheduling for energy-efficient distributed and flexible job shop, J. Clean. Prod. (2021) 293, https://doi.org/10.1016/j.jclepro.2021.126093.
[55] V. Bavane, M. Studies, R.V. Marode, Digital Twin, Manufacturing Excellence Through Virtual Factory Digital Twin : Manufacturing Excellence Through Virtual, White Pap (2018).
[56] E.J. Tuegel, A.R. Ingraffea, T.G. Eason, S.M. Spottswood, Reengineering aircraft structural life prediction using a digital twin, Int. J. Aerosp. Eng. (2011), https://doi.org/10.1155/2011/154798.
[57] C. Zhuang, T. Miao, J. Liu, H. Xiong, The connotation of digital twin, and the construction and application method of shop-floor digital twin, Robot. Comput. Integr. Manuf. (2021) 68, https://doi.org/10.1016/j.rcim.2020.102075.
[58] Z. Zhu, X. Xi, X. Xu, Y. Cai, Digital Twin-driven machining process for thin-walled part manufacturing, J. Manuf. Syst. 59 (2021) 453–466, https://doi.org/10.1016/j.jmsy.2021.03.015.
[59] E. Negri, V. Pandhare, L. Cattaneo, J. Singh, M. Macchi, J. Lee, Field-synchronized digital twin framework for production scheduling with uncertainty, J. Intell. Manuf. 32 (2021) 1207–1228, https://doi.org/10.1007/s10845-020-01685-9.
[60] K. Tliba, T.M.L. Diallo, O. Penas, R. Ben Khalifa, N. Ben Yahia, J.Y. Choley, Digital twin-driven dynamic scheduling of a hybrid flow shop, J. Intell. Manuf. (2022), https://doi.org/10.1007/s10845-022-01922-3.
[61] M. Zhang, F. Tao, A.Y.C. Nee, Digital Twin Enhanced Dynamic Job-Shop Scheduling, J. Manuf. Syst. 58 (2021) 146–156, https://doi.org/10.1016/j.jmsy.2020.04.008.
[62] Y. Fang, C. Peng, P. Lou, Z. Zhou, J. Hu, J. Yan, Digital-Twin-Based Job Shop Scheduling Toward Smart Manufacturing, IEEE Trans. Ind. Informatics. 15 (2019) 6425–6435, https://doi.org/10.1109/TII.2019.2938572.
[63] J. Yan, Z. Liu, C. Zhang, T. Zhang, Y. Zhang, C. Yang, Research on flexible job shop scheduling under finite transportation conditions for digital twin workshop, Robot. Comput. Integr. Manuf. (2021), https://doi.org/10.1016/j.rcim.2021.102198.
[64] L. Liu, K. Guo, Z. Gao, J. Li, J. Sun, Digital Twin-Driven Adaptive Scheduling for Flexible Job Shops, SUSTAINABILITY, 2022, https://doi.org/10.3390/su14095340.
[65] X. Wu, S. Tian, L. Zhang, The Internet of Things Enabled Shop Floor Scheduling and Process Control Method Based on Petri Nets, IEEE Access 7 (2019) 27432–27442, https://doi.org/10.1109/ACCESS.2019.2900117.
[66] Z. Zhuang, Y. Li, Y. Sun, W. Qin, Z.H. Sun, Network-based dynamic dispatching rule generation mechanism for real-time production scheduling problems with dynamic job arrivals, Robot. Comput. Integr. Manuf. (2022) 73, https://doi.org/10.1016/j.rcim.2021.102261.
[67] Y. Feng, Z. Hong, Z. Li, H. Zheng, J. Tan, Integrated intelligent green scheduling of sustainable flexible workshop with edge computing considering uncertain machine state, J. Clean. Prod. (2020), https://doi.org/10.1016/j.jclepro.2019.119070.

[68] LZ. Xu, Q.S. Xie, Dynamic production scheduling of digital twin job-shop based on edge computing, J. Inf. Sci. Eng. 37 (2021) 93–105, https://doi.org/10.6688/JISE.202101_37(1).0007.
[69] S. Luo, L. Zhang, Y. Fan, Real-Time Scheduling for Dynamic Partial-No-Wait Multiobjective Flexible Job Shop by Deep Reinforcement Learning, IEEE Trans. Autom. Sci. Eng. (2021), https://doi.org/10.1109/TASE.2021.3104716.
[70] Y. Lv, C. Li, Y. Tang, Y. Kou, Toward energy-efficient rescheduling decision mechanisms for flexible job shop with dynamic events and alternative process plans, IEEE Trans. Autom. Sci. Eng. (2021), https://doi.org/10.1109/TASE.2021.3115821.
[71] K. Li, Q. Deng, L. Zhang, Q. Fan, G. Gong, S. Ding, An effective MCTS-based algorithm for minimising makespan in dynamic flexible job shop scheduling problem, Comput. Ind. Eng. (2021) 155, https://doi.org/10.1016/j.cie.2021.107211.
[72] A.K. Turker, A. Aktepe, A.F. Inal, O.O. Ersoz, G.S. Das, B. Birgoren, A decision support system for dynamic job-shop scheduling using real-time data with simulation, Mathematics 7 (2019), https://doi.org/10.3390/math7030278.
[73] S. Tian, T. Wang, L. Zhang, X. Wu, The Internet of Things enabled manufacturing enterprise information system design and shop floor dynamic scheduling optimisation, Enterp. Inf. Syst. 14 (2020) 1238–1263, https://doi.org/10.1080/17517575.2019.1609703.
[74] F. Zhang, Y. Mei, S. Nguyen, M. Zhang, K.C. Tan, Surrogate-Assisted Evolutionary Multitask Genetic Programming for Dynamic Flexible Job Shop Scheduling, IEEE Trans. Evol. Comput. 25 (2021) 651–665, https://doi.org/10.1109/TEVC.2021.3065707.
[75] L.H. Wu, P.Y. Mok, J. Zhang, An adaptive multi-parameter based dispatching strategy for single-loop interbay material handling systems, Comput. Ind. 62 (2011) 175–186, https://doi.org/10.1016/j.compind.2010.10.010.
[76] MA Adibi, M. Zandieh, M. Amiri, Multi-objective scheduling of dynamic job shop using variable neighborhood search, Expert Syst. Appl. 37 (2010) 282–287, https://doi.org/10.1016/j.eswa.2009.05.001.
[77] I. Kacem, S. Hammadi, P. Borne, Approach by localisation and multiobjective evolutionary optimisation for flexible job-shop scheduling problems, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 32 (2002) 1–13, https://doi.org/10.1109/TSMCC.2002.1009117.
[78] W. Xia, Z. Wu, An effective hybrid optimisation approach for multi-objective flexible job-shop scheduling problems, Comput. Ind. Eng. 48 (2005) 409–425, https://doi.org/10.1016/j.cie.2005.01.018.
[79] JQ. Li, QK. Pan, K.Z. Gao, Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems, Int. J. Adv. Manuf. Technol. 55 (2011) 1159–1169, https://doi.org/10.1007/s00170-010-3140-2.
[80] X. Huang, Z. Guan, L. Yang, An effective hybrid algorithm for multi-objective flexible job-shop scheduling problem, Adv. Mech. Eng. (2018) 10, https://doi.org/10.1177/1687814018801442.
[81] H. Ding, X. Gu, Hybrid of human learning optimisation algorithm and particle swarm optimisation algorithm with scheduling strategies for the flexible job-shop scheduling problem, Neurocomputing 414 (2020) 313–332, https://doi.org/10.1016/j.neucom.2020.07.004.
[82] J. Fang, Y. Xi, A rolling horizon job shop rescheduling strategy in the dynamic environment, Int. J. Adv. Manuf. Technol. 13 (1997) 227–232, https://doi.org/10.1007/BF01305874.
[83] W. Wang, H. Yang, Y. Zhang, J. Xu, IoT-enabled real-time energy efficiency optimisation method for energy-intensive manufacturing enterprises, Int. J. Comput. Integr. Manuf. 31 (2018) 362–379, https://doi.org/10.1080/0951192X.2017.1337929.