Dynamic Scheduling Framework of the Flexible Mixed-Model Assembly Line Based on the Internet of Manufacturing Things

LEI SHI¹, AND GANG GUO²

¹School of Mechanical Engineering, Chongqing University, Chongqing 400044, China
²School of Automotive Engineering, Chongqing University, Chongqing 400044, China

Corresponding author: Gang Guo (guogangcqu@gmail.com)

This work was supported by the National Key Research and Development Program of China under Grant 2019YFB1706102.

ABSTRACT To satisfy the needs of market customization, the traditional manufacturing is gradually transforming and upgrading into the intelligent manufacturing. In mass customization (MC), the assembly operations of modular products tend to be organized as the form of flexible mixed-model assembly line (FMMAL). The dynamic scheduling problem of FMMAL is quite complex with three issues of the product sequencing, station allocation and material delivery. At the same time, the disruption events of station failure, product inserting and product reworking are also considered. To solve this problem, a comprehensive framework combining the architecture of Internet of Manufacturing Things (IoMT) with the dynamic scheduling algorithms is proposed. Firstly, the IoMT-based FMMAL are constructed via the multi-agent system (MAS) and ubiquitous environment. Secondly, a mathematical model of FMMAL are formulated with the decision variables, optimization objectives and constraint conditions. Thirdly, the IoMT-oriented algorithms for dynamic scheduling are proposed including the fuzzy analytic hierarchy process (FAHP) for normalization, weighted sum of properties-improved genetic algorithm (WSP-IGA) for prescheduling, priority weights search-simulated annealing (PWS-SA) for rescheduling. Lastly, a discrete event simulation of a numerical case is conducted to demonstrate the practicality and validity of proposed theories and algorithms. The results show that the proposed hyper-heuristics (WSP-IGA and PWS-SA) can respectively realize the prescheduling and rescheduling of FMMAL in four modes including the synthesized mode, time-efficient mode, just-in-time mode and energy-saving mode, which are superior to the four referenced meta-heuristics.

INDEX TERMS Internet of manufacturing things, dynamic scheduling, flexible mixed-model assembly line, hyper-heuristics.

LIST OF SYMBOLS

| Symbol | Description |
|--------|-------------|
| I, n_I | Quantity and set of planned products |
| J, n_J | Quantity and set of assembly operations or stages |
| K, n_K | Quantity and set of assembly stations |
| A, n_A | Quantity and set of AGVs for part delivery |
| l*, J* | Set of inserted products and set of reworked operations |
| K_j, R_j | Quantity and set of stations at stage j |
| x_(i,j,k) | 0-1 variable, whether the operation j of product i is executed in station k or not |
| y_(i',j,k) | 0-1 variable, whether the operation j of product i' in station k precedes that of product i or not |
| z_(a,t,j) | 0-1 variable, whether the target of delivery t of AGV a is stock j or not |
| S_(i,j), F_(i,j) | Starting time point and ending time point of operation j of product i |
| A_(T_(i,j,k)) | Assembly time of operation j of product i in station k |
| M_(T_(i,j,k)) | Maintenance time of operation j of product i in station k |
| W_(T_(i,j)) | Waiting time of operation j of product i in station k |
| L | A large enough number |
| p_(i,k) | Average assembly power of operation j of product i in station k |
| p_b | Average basic power of whole assembly line |
The automated guided vehicle (AGV) and modular products. The FMMAL is a kind of assembly line which can continuously and stably produce a kind of assembly line which can continuously and stably assemble various types of products. At the same time, these products need to be designed for a product family, namely that each product has the similar parts, assembly sequence and delivery sequence, which makes it possible to carry out the multi-variety assemblies according to the customized needs. The studies of MMALs generally focus on three problems including the balancing, scheduling and delivery for the MMALs. And these problems are usually studied together.

The flexible mixed-model assembly line (FMAL) [24], [25] refers to a kind of assembly line which can continuously and stably assemble various types of products. At the same time, these products need to be designed for a product family, namely that each product has the similar parts, assembly sequence and delivery sequence, which makes it possible to carry out the multi-variety assemblies according to the customized needs. The studies of MMALs generally focus on three problems including the balancing, scheduling and delivery for the MMALs. And these problems are usually studied together.

The flexible mixed-model assembly line (FMAL) [24], [25] refers to a kind of MMAL of which each stage or operation is introduced with the unrelated parallel assembly stations to eliminate the bottleneck and improve the flexibility. And the FMAL has become a common form of intelligent production line in mass customization (MC) to organize the assembly operations of modular products. The FMAL is similar to the hybrid flow shop (HFS) [26], but the FMAL scheduling problem is far more complex than the HFS scheduling problem [27] because the bill of materials (BOM) [28] and material delivery [29] need to be considered. Hence the scheduling problem of FMAL is also the NP-hard problem besides that of HFS. What’s more, there are a lot of uncertainties in the real world and the various disruption events will degrade the quality of scheduling solution, which increases the complexity of FMAL scheduling problem further [30]. So, this study on the dynamic scheduling of FMAL is very meaningful.

To our best knowledge, there is not any literature investigating on the dynamic scheduling of FMAL in MC because this problem combines three issues of the product sequencing, station allocation and material delivery. And the traditional architectures and methods of manufacturing systems fail to solve this problem because of disruption events. Therefore, it is necessary to introduce a comprehensive framework combining the architecture of IoMT with dynamic scheduling algorithms to intellectualize and reconstruct the FMAL, which makes the FMAL react and adjust to the disruption events in real time, so as to complete the tasks as well as possible. The goal of this paper is to solve the practical application of FMAL in MC at some fast and real-time computation occasions. In other words, the prescheduling and rescheduling solutions of FMAL are quasi-optimal but not exactly optimal. The contributions and innovations are as follows.
1) According to the IoMT system, a dynamic scheduling unit (DSU) of assembly operation is proposed to architect the FMMAL in MC. 
2) A mixed integer programming model (MIPM) for the dynamic scheduling of FMMAL is formulated.
3) The multiple objectives of FMMAL including the $C_{max}$, TET and TEC, are normalized via the fuzzy analytic hierarchy process (FAHP).
4) A weighted sum of properties-improved genetic algorithm (WSP-IGA) are proposed for prescheduling.
5) A priority weights search-simulated annealing (PWS-SA) are proposed for rescheduling.
6) A discrete event simulation of a numerical case is conducted.

The remainder of this paper is organized as follows. Section II reviews the dynamic scheduling algorithms and architectures of IoMT systems as well as the research gaps. Section III constructs the IoMT-based FMMAL via the MAS and ubiquitous environment. Section IV formulates a mathematical model of dynamic scheduling of FMMAL. Section V proposes the IoMT-oriented algorithms for dynamic scheduling. Section VI demonstrates the practicality and validity of proposed theories and algorithms via a discrete event simulation of a numerical case. Section VII concludes this paper.

II. RELATED WORK
Because of uncertainties, real-time reactions and multiple objectives, the dynamic scheduling algorithms have gradually developed from the exact algorithms to the intelligent search algorithms which exploit some dispatching rules and intelligent approaches to search for the quasi-optimal solutions within limited time [12]. To tackle the dynamic nature of FMS, Jerald et al. [13] designed different scheduling mechanisms including the genetic algorithm, simulated annealing algorithm, memetic algorithm and particle swarm algorithm, and considered two objectives of minimizing the idle time and minimizing the total penalty cost. Fattahi and Fallahi [14] developed a meta-heuristic algorithm based on the genetic algorithm to solve the dynamic scheduling in flexible job shop, considering two objectives of the efficiency and stability. Against a complex dynamic job shop problem, Pickardt et al. [15] proposed a two-stage hyper-heuristic for the generation of a set of work center-specific dispatching rules combining a genetic programming algorithm with an evolutionary algorithm. Nguyen et al. [16] developed a new approach to automatic programming via iterated local search for dynamic job shop scheduling to overcome the limitations of automatic design of dispatching rules. Tang et al. [17] adopted an improved particle swarm optimization for the Pareto optimal solution to address the dynamic flexible flow shop scheduling problem reducing the energy consumption and makespan. Liu et al. [18] proposed a proactive-reactive approach and an enhanced NSGA-II algorithm hybridization strategy to study a permutation flow shop scheduling problem considering the objective of total flow time. Zhang and Wang [19] employed the constraint programming and mixed-integer linear programming to formulate the flexible assembly job-shop scheduling problem in a dynamic manufacturing environment, and developed several dispatching rules with machine feedback mechanism. Zhou et al. [20] proposed three types of hyper-heuristic methods for coevolution of the machine assignment rules and job sequencing rules to solve the multi-objective dynamic flexible job shop scheduling problem, and found that the heuristics discovered by the evolved scheduling policies are more useful and competitive. The above studies are very inspiring, but they fail to design the hyper-heuristics on the basis of product properties, including the assembly time, due time and assembly power. At the same time, they fail to tackle three issues of the product sequencing, station allocation and material delivery together, and fail to consider three objectives of the $C_{max}$, TET and TEC together. All these are critical to the dynamic scheduling of FMMAL in MC. 

The IoMT system has the characteristics of real-time perception, monitoring and controlling, which is an important guarantee to realize the dynamic scheduling algorithms. Especially the MAS and ubiquitous environment employed in the IoMT system facilitate the production data collection and distributed-artificial-intelligence implementation [31]. Xiang and Lee [32] build an efficient agent-based dynamic scheduling for real-world manufacturing systems with various products, processes, and disturbances, combining the ant colony intelligence with the local agent coordination to obtain efficient global performance. Erol et al. [33] proposed a multi-agent-based scheduling approach for automated guided vehicles and machines within a manufacturing system. And this approach can work under a real-time environment and generate feasible schedules using negotiation/bidding mechanisms between agents. Jiang et al. [34] focused on distributed optimal scheduling based on multi-agent systems including the goals and constraints of the system, a two-layer decision model and the required indicators, the roles and functions of different agents, the dynamic decision cycle and the multistage negotiation mechanism, and a rescheduling algorithm. Tao and Qi [35] proposed a framework-New IT driven service-oriented smart manufacturing, combining the Internet of Things, cloud computing, big data, mobile Internet and cyber-physical systems, which facilitated the visions of smart manufacturing. Wang et al. [36] proposed a new multiagent-based real-time scheduling architecture for an Internet of Things-enabled flexible job shop, including a real-time task assigning dynamic scheduling strategy and a bargaining-game-based negotiation mechanism. Feng et al. [37] proposed a dynamic scheduling system based on the cyber-physical energy monitoring system to improve the energy efficiency of the job shop, including a novel process of scheduling-monitoring-updating optimizing, the tool aging condition and geometry information of the workpiece, and a modified genetic algorithm with multi-layer coding. Fang et al. [38] proposed a new job shop scheduling method based on
digital twin to reduce the scheduling deviation, including the architecture and working principle of the new job shop scheduling mode, scheduling resource parameter updating methods and dynamic interactive scheduling strategies. The above studies present the development trend of IoMT system, but they fail to construct the IoMT on the basis of enterprise information system in MC. And they fail to build the MAS and ubiquitous environment according the characteristics of FMMAL. At the same time, they fail to propose the specialized agent negotiation for dynamic scheduling of FMMAL. All these are critical to the construction of IoMT-based FMMAL in MC.

Against the gaps of above studies, this paper proposes a comprehensive framework combining the architecture of IoMT with the dynamic scheduling algorithms to construct the IoMT-based FMMAL which can react and adjust in real time according to the production conditions, device states and disruption events, so as to achieve the objectives of minimizing the $C_{\text{max}}$, $TET$ and $TEC$.

III. SYSTEM CONSTRUCTION

A. OVERALL ARCHITECTURE

The IoMT system is the foundation of dynamic scheduling of intelligent production line in MC. It employs the technologies of modern informatics, advanced manufacturing and automatic production to integrate the enterprise information system (EIS) and underlying physical system to meet the various, customized and dynamic needs in MC. The proposed overall architecture of IoMT system is shown in Figure 1. It is based on the idea of service-oriented architecture (SOA) [39] and considers the MC production as the core to link the isolated information subsystems and underlying shop floor. From top to bottom, the proposed IoMT system is comprised of the enterprise LAN, factory LAN and fieldbus network.

The enterprise LAN is comprised of the enterprise resource planning (ERP), product data management (PDM), computer-aided process planning (CAPP), computer-aided technologies (CAX), data center and firewall. The factory LAN is comprised of the manufacturing execution system (MES), advanced planning and scheduling (APS) and database server. And the proposed MAS in this paper runs on the APS. The fieldbus network of discrete manufacturing is comprised of the fieldbus protocol, industrial Wi-Fi, execution devices and perception devices. The WebService+XML connects the function modules of different network applications via the defined interfaces and protocols, so that these function modules can mutually interact in a unified way. The OPC unified architecture (OPC UA) is a cross-platform standard for communication, aiming to send messages between the clients and servers on the heterogeneous networks, so as to realize the communication between the underlying devices and network applications.

B. FMMAL ARCHITECTURE
Referring to the DSU of HFS [40], a DSU of FMMAL is proposed here. There are multiple unrelated assembly stations undertaking the same assembly operation at each stage of FMMAL and the scheduling relationship between the adjacent stages is relatively independent. Hence each stage containing the parallel stations can be considered a DSU of assembly operation to construct the FMMAL. As shown in Figure 2(a), from top to bottom, the proposed DSU is accordingly comprised of three levels, namely the EIS, MAS and ubiquitous shop floor.

The EIS is comprised of the ERP, MES, PDM, CAPP and CAX, and it serves as a manufacturing knowledge database which manages the order tasks, process data and completion feedbacks. The MAS is comprised of the scheduling agent, dispatch agent, station agent, inspection agent and AGV agent, and it is a middleware running on the APS to provide the real-time decision support via the communication, cooperation and competition between the agents. The ubiquitous shop floor is comprised of various execution devices and perception devices, and it performs the control commands from the MAS and perceives the production data. The execution devices include the source, insertion, dispatch buffer, stations, inspection buffer, drain, stock buffer and AGVs. The perception devices include the RFID readers, RFID tags, multi-source sensors, flaw detectors, Wi-Fi and gateways. In the DSU, the indicators of the assembly time, punctuality and energy consumption are considered.

The FMMAL consists of several stages containing the parallel stations. As shown in Figure 2(b), each stage is considered a DSU in this paper, so the proposed FMMAL can be constructed with a series connection of several DSUs. The source and insertion are only set at the beginning, while the drain is only set at the end. And the whole line shares a common AGV delivery system. The products can be reworked at each stage and the reworked products are returned to the dispatch buffer of this stage. The dispatch agent, station agent, inspection agent and stock agent are local to their own DSU, while the scheduling agent and AGV agent are global to all the DSUs. The considered disruption events of FMMAL are the station failure, product inserting and product reworking. The considered optimization objectives of FMMAL are minimizing the $C_{\text{max}}$, $TET$ and $TEC$.

![Diagrammatic Sketch of the FMMAL Architecture](image-url)

**FIGURE 2.** Diagrammatic Sketch of the FMMAL Architecture: (a) DSU of the FMMAL and (b) FMMAL constructed with the DSUs.
C. MAS OF THE FMMAL
Since the MAS is a significant part of the DSU of FMMAL, this paper needs to build and elucidate the MAS of FMMAL further here, as shown in Figure 3.

1) Scheduling agent. The scheduling agent is global to all the DSUs and the whole FMMAL has only one public scheduling agent. As shown in Figure 3(a), the scheduling agent is responsible for receiving the order tasks in real time, initializing the production scheduling conditions, generating the optimal prescheduling and rescheduling solutions dynamically and evaluating the performance of production scheduling results.

2) Dispatch agent. The dispatch agent is local to its own DSU and each DSU has only one private dispatch agent. As shown in Figure 3(b), the dispatch agent is responsible for identifying the planned products, inserted products and reworked products in a certain dispatch buffer in real time and then assigning the corresponding priorities to the products according to their process properties.

3) Station agent. The station agent is local to its own DSU and each DSU may have several private station agents. As shown in Figure 3(c), the station agent is responsible for monitoring the information of station states and assembly operations, and selecting one of the products in the dispatch buffer to assemble according to their priorities. Once a station failure occurs, the station agent will call for maintenance.

4) Inspection agent. The inspection agent is local to its own DSU and each DSU has only one private inspection agent. As shown in Figure 3(d), the inspection agent is responsible for acquiring the data of assembly qualities in a certain inspection buffer. The unqualified products are sent into the specified dispatch buffer for reworking, while the qualified products are sent into the drain or the next stage with the finish quantity recorded.

5) Stock agent. The stock agent is local to its own DSU and each DSU has only one private stock agent. As shown in Figure 3(e), the stock agent is responsible for recording the quantity change of parts in the corresponding stock buffer in real time and controlling the stock to receive the delivered parts by an AGV and feed the needed parts for the assembly operation at this stage.

6) AGV agent. The AGV agent is global to all the DSUs and the whole FMMAL may have several public AGV agents. As shown in Figure 3(f), the AGV agent is responsible for selecting one of the stock buffers as a target according to their priorities and controlling the corresponding AGV to carry the needed parts from the warehouse to the targeted stock buffer. When a delivery is finished, the AGV will return to the warehouse.

In the DSU, when a product can select one of parallel stations for assembly, the dispatch agent will publish this assembly task to the station agents and they will bid for the task. Only if a station is available and has the minimum priority value, the corresponding station agent will win the bid of the task. When a finished product leaves its station and there are still multiple products in the dispatch buffer, the station agent will select the product with the minimum priority value.
into the corresponding station to assemble. When an AGV can select one of multiple stock buffers for delivery, the AGV agent will publish this delivery service to the stock agents and they will bid for the service. Only if a stock buffer is available and has the minimum priority value, the corresponding stock agent will win the bid of service.

IV. MATHEMATICAL MODEL

The MIPM [41] for the dynamic scheduling of FMMAL is formulated here, which includes the decision variables, optimization objectives and constraint conditions.

A. DECISION VARIABLES

The FMMAL dynamic scheduling has three kinds of decision variables, namely \( x_{ijk} \), \( y_{i'jk} \) and \( z_{atj} \), as shown in Equation (1)–(3). \( x_{ijk} \) is used to determine the station allocation. \( y_{i'jk} \) is used to determine the product sequencing. And \( z_{atj} \) is used to determine the material delivery.

\[
x_{ijk} = \begin{cases} 1, & \text{if } (i, j, k) \text{ is true; } \\ 0, & \text{otherwise.} \\
\end{cases} \tag{1}
\]

\[
y_{i'jk} = \begin{cases} 1, & \text{if } (i', j, k) \text{ preceding } (i, j, k) \text{ is true; } \\ 0, & \text{otherwise.} \\
\end{cases} \tag{2}
\]

\[
z_{atj} = \begin{cases} 1, & \text{if } (a, t, j) \text{ is true; } \\ 0, & \text{otherwise.} \\
\end{cases} \tag{3}
\]

B. OPTIMIZATION OBJECTIVES

In the assembly process of modular products in MC, there are various evaluation criteria and optimization objectives. This paper here selects the most important three objectives, namely minimizing the \( C_{\text{max}} \), \( TET \) and \( TEC \) which represent the time-efficient, just-in-time and energy-saving performances of FMMAL, respectively. The optimization objectives of FMMAL is shown in Equation (4)–(6).

\[
f_1 = \min C_{\text{max}} = \min C_{i=last} \tag{4}
\]

\[
f_2 = \min TET = \min \sum_{i\in\mathbb{U}_t} |C_i - d_i| \tag{5}
\]

\[
f_3 = \min TEC = \min \left[ p_0 \cdot C_{\text{max}} + \sum_{i\in\mathbb{U}_t} \sum_{j\in\mathbb{U}_j} \sum_{k\in\mathbb{K}_j} x_{ijk} \cdot AT_{ijk} \cdot \tilde{p}_{ijk} \\
+ \sum_{i\in\mathbb{U}_t} \sum_{j\in\mathbb{U}_j} \sum_{k\in\mathbb{K}_j} y_{i'jk} \cdot (S_{ij} - F_{i'j} + WT_{ij}) \cdot \tilde{p}_{k}^{\text{date}} \\
+ \sum_{i\in\mathbb{U}_t} \sum_{j\in\mathbb{U}_j} \sum_{k\in\mathbb{K}_j} x_{ijk} \cdot MT_{ijk} \cdot \tilde{p}_{k}^{\text{failure}} \right] \tag{6}
\]

C. CONSTRAINT CONDITIONS

Considering the product inserting, product reworking, station failure and part shortage, this paper introduces the set of inserted products \( I^* \), set of reworked operations \( J^* \), maintenance time \( MT_{ijk} \) and waiting time \( WT_{ij} \) into the traditional constraints of FMMAL [41]. Thus, the constraints are extended to the dynamic ones.

The constraints for the assembly production are shown as Equation (7)–(11).

1) Any operation of products is executed in a certain station.

\[
\sum_{k\in\mathbb{K}_j} x_{ijk} = 1 \quad (\forall i \in I \cup I^*, \forall j \in J \cup J^*) \tag{7}
\]

2) There is at least one assembly station at any stage or operation.

\[
K_j \neq \emptyset \cap R_j \geq 1 \quad (\forall j \in J \cup J^*) \tag{8}
\]

3) Operation duration of products equals to the sum of the assembly time, waiting time and maintenance time.

\[
\begin{align*}
F_{ij} &= S_{ij} + x_{ijk} \times (AT_{ijk} + WT_{ij} + MT_{ijk}) \\
WT_{ij} &= \text{the needed time to satisfy } Q_j \geq q_{ij} \\
&\quad (\forall i \in I \cup I^*, \forall j \in J \cup J^*, \forall k \in K_j, \exists W_{Tijk} \geq 0, \exists MT_{ijk} \geq 0)
\end{align*} \tag{9}
\]

4) The starting time point of any operation of products is later than the ending time point of its preceding operation.

\[
S_{ij} \geq F_{ij-1} \quad (\forall i \in I \cup I^*, \forall j \in J \cup J^*, j > 1) \tag{10}
\]

5) There is only one operation of products executed in a certain station at any moment.

\[
S_{ij} + L \times (1 - y_{i'jk}) \geq F_{i'j} \quad (\forall i, i' \in I \cup I^*, i' \neq i, \forall j \in J \cup J^*, \forall k \in K_j) \tag{11}
\]

The constraints for the AGV deliveries are shown as Equation (12)–(16).

1) Any delivery of parts is targeted to a certain stock buffer.

\[
\sum_{j\in\mathbb{J}_j} z_{atj} = 1 \quad (\forall a \in A, \forall t = 1, 2, 3 \ldots) \tag{12}
\]

2) Any delivery of parts is undertaken by a certain AGV.

\[
\sum_{a\in\mathbb{A}} z_{atj} = 1 \quad (\forall j \in J, \forall t = 1, 2, 3 \ldots) \tag{13}
\]

3) The AGV in any delivery of parts is loaded to capacity.

\[
Q_j^{\text{full}} - Q_j \geq z_{atj} \times q_a \quad (\forall a \in A, \forall j \in J, \forall t = 1, 2, 3 \ldots) \tag{14}
\]

4) The part stock in any stock buffer is more than zero and less than its capacity.

\[
0 \leq Q_j \leq Q_j^{\text{full}} \quad (\forall a \in A, \forall j \in J, \forall t = 1, 2, 3 \ldots) \tag{15}
\]

5) The path of AGV in any delivery of parts is from the warehouse to the targeted stock buffer then back to the warehouse.

\[
DT_{atj} = z_{atj} \times \frac{2D_j}{v_a} \quad (\forall a \in A, \forall j \in J, \forall t = 1, 2, 3 \ldots) \tag{16}
\]

V. DYNAMIC SCHEDULING ALGORITHMS

A. DISPATCHING RULES
This paper here proposes a weighted sum of properties (WSP) dispatching rule, as shown in Equation (17). On the basis of hyper-heuristic concept [15], [16], [20], the WSP dispatching rule needs the IoMT-based FMMAL actively perceive the production condition and dynamically search the priority weights to generate the priorities according to the optimization objectives. Three considered process properties of products in MC are the assembly time, due time and assembly power, which correspond to three optimization objectives of minimizing the $C_{\text{max}}$, TET and TEC.

$$\Pr_{ijk} = u_1 \cdot \frac{AT_{ijk} - AT_{\text{min}}}{AT_{\text{max}} - AT_{\text{min}}} + u_2 \cdot \frac{d_i - S_i}{d_{\text{max}}}$$
$$+ u_3 \cdot \frac{(\bar{p} \cdot AT)_{\text{max}} - (\bar{p} \cdot AT)_{\text{min}}}{(\bar{p} \cdot AT)_{\text{max}} - (\bar{p} \cdot AT)_{\text{min}}}$$ (17)

The WSP dispatching rule is actually a method for manufacturing-resource matching. In the IoMT based FMMAL, the less the priority is, the higher the matching degree is.

Besides, four heuristic dispatching rules are selected as the compared groups of WSP dispatching rule and they are the shortest processing time (SPT), first come first serve (FCFS), earliest due date (EDD) and longest processing time (LPT), respectively.

This paper here proposes a lowest stock rate (LSR) dispatching rule, as shown in Equation (18).

$$Pr_{atj} = \frac{Q_j}{Q_j^{\text{full}}}$$ (18)

The LSR dispatching rule is actually a method for delivery-resource matching. In the IoMT based FMMAL, the less the priority is, the higher the matching degree is.

B. NORMALIZATION METHOD

In order to improve the computation efficiency and satisfy the dynamic scheduling, the FAHP [43] is employed to normalize the multiple objectives to obtain the fitness function, as shown in Equation (19).

$$F = w_1 \cdot \frac{C_{\text{max}} - (C_{\text{max}})_{\text{min}}}{(C_{\text{max}})_{\text{max}} - (C_{\text{max}})_{\text{min}}}$$
$$+ w_2 \cdot \frac{TET - TET_{\text{min}}}{TET_{\text{max}} - TET_{\text{min}}}$$
$$+ w_3 \cdot \frac{TEC - TEC_{\text{min}}}{TEC_{\text{max}} - TEC_{\text{min}}}$$ (19)

In the FAHP, the average scores of three experts are adopted, and the four production modes are considered, including the synthesized, time-efficient, just-in-time and energy-saving modes. The FAHP results of the four production modes are shown in Table 1.

| Mode                  | Pairwise comparison matrices | Fuzzy weights | Defuzzified weights | Normalized weights |
|-----------------------|-----------------------------|---------------|---------------------|--------------------|
| Synthesized mode      | $C_{\text{max}}$ (1.90, 1.00, 1.00) | (0.83, 1.33, 1.83) | (0.61, 0.89, 1.67) | (0.9100, 0.3489, 0.6475) | 1.0000 | 0.3459 |
|                       | $TET$ (0.56, 0.78, 1.33)     | (1.00, 1.00, 1.00) | (0.67, 1.17, 1.67) | (0.1737, 0.3196, 0.5755) | 0.9295 | 0.3215 |
|                       | $TEC$ (0.67, 1.17, 1.67)     | (0.61, 0.89, 1.67) | (1.00, 1.00, 1.00) | (0.1776, 0.3315, 0.6245) | 0.9616 | 0.3326 |
| Time-efficient mode   | $C_{\text{max}}$ (1.00, 1.00, 1.00) | (1.83, 2.33, 2.83) | (1.67, 2.17, 2.67) | (0.3459, 0.5258, 0.7629) | 1.0000 | 0.8173 |
|                       | $TET$ (0.36, 0.43, 0.56)     | (1.00, 1.00, 1.00) | (0.61, 0.89, 1.67) | (0.1514, 0.2218, 0.3791) | 0.0985 | 0.0805 |
|                       | $TEC$ (0.38, 0.47, 0.61)     | (0.67, 1.17, 1.67) | (1.00, 1.00, 1.00) | (0.1576, 0.2524, 0.3850) | 0.1251 | 0.1022 |
| Just-in-time mode     | $C_{\text{max}}$ (1.00, 1.00, 1.00) | (0.40, 0.50, 0.67) | (0.67, 1.17, 1.67) | (0.1618, 0.2618, 0.4058) | 0.2456 | 0.1689 |
|                       | $TET$ (1.50, 2.00, 2.50)     | (1.00, 1.00, 1.00) | (1.67, 2.17, 2.67) | (0.3260, 0.5069, 0.7497) | 1.0000 | 0.6878 |
|                       | $TEC$ (0.61, 0.89, 1.67)     | (0.38, 0.47, 0.61) | (1.00, 1.00, 1.00) | (0.1556, 0.2314, 0.3985) | 0.2084 | 0.1433 |
| Energy-saving mode    | $C_{\text{max}}$ (1.00, 1.00, 1.00) | (0.67, 1.17, 1.67) | (0.38, 0.47, 0.61) | (0.1589, 0.2553, 0.3914) | 0.1739 | 0.1304 |
|                       | $TET$ (0.61, 0.89, 1.67)     | (1.00, 1.00, 1.00) | (0.38, 0.47, 0.61) | (0.1543, 0.2282, 0.3914) | 0.1602 | 0.1201 |
|                       | $TEC$ (1.67, 2.17, 2.67)     | (1.00, 1.00, 1.00) | (0.3364, 0.5164, 0.7566) | 1.0000 | 0.7496 |

The four production modes correspond to the four groups of normalization weights $\mathbf{w} = (w_1, w_2, w_3)$. Where, the weights of synthesized mode are $\mathbf{w} = (0.3459, 0.3215, 0.3326)$. The weights of time-efficient mode are $\mathbf{w} = (0.8173, 0.0805, 0.1022)$. The weights of just-in-time mode are $\mathbf{w} = (0.1689, 0.6878, 0.1433)$. The weights of energy-saving mode are $\mathbf{w} = (0.1304, 0.1201, 0.7496)$.

When the dynamic scheduling of FMMAL is carried out, the normalization weights are put into the Equation (19), corresponding to a certain production mode. And the
optimization direction in this paper is to minimizing the fitness function $F$.

C. PRESCHEDULING ALGORITHM
An improved genetic algorithm with two segments of chromosomes is proposed here to overcome the defects of the traditional real-matrix encoding. The first segment of chromosome ($Chromosome 1$) is used for the product sequencing, while the second segment of chromosome ($Chromosome 2$) is used for the station allocation. Combining the WSP dispatching rule with the improved genetic algorithm, the WSP-IGA is proposed. Based on the IoMT system, the WSP-IGA is a kind of hyper-heuristic algorithm to search for the quasi-optimal prescheduling solution of FMMAL. The flowchart of the WSP-IGA algorithm is shown as Figure 4.

![Flowchart of the WSP-IGA algorithm.](image)

1) Encoding and decoding
The two-segment-chromosome encoding is adopted by the WSP-IGA. The $Chromosome 1$ is used to search for the planned product sequence. It adopts the sequential encoding and the genes represent the planned product codes in the source, as shown in Equation (20).

$$Chromosome 1 = (g_1, g_2, \ldots, g_l, \ldots, g_{n_l}), \quad g_i \in I, \quad g_1 \neq g_2 \neq \ldots \neq g_l \neq \ldots \neq g_{n_l}, \quad i = 1, 2 \ldots n_l \quad (20)$$

Where $g_i$ is the code of product, $I$ is the set of planned products and $n_l$ is the quantity of planned products.

The decoding of $Chromosome 1$ is shown in Equation (21). The decoded $Chromosome 1$ means a feasible solution of planned product sequence in the source.

$$Product Sequence = \left( P_{g_1}, P_{g_2}, \ldots, P_{g_{m}}, \right) \quad (21)$$

Where $P_{g_i}$ is a product represented by the code $g_i$.

The $Chromosome 2$ is used to search for the priority weights of planned products. It adopts the binary encoding and three decimal numbers are represented by a chromosome, as shown in Equation (22).

$$Chromosome 2 = (g_1, g_2, \ldots, g_i, \ldots, g_{3m}), \quad g_i \in \{0, 1\}, \quad i \in \{1, 2, \ldots, 3m\} \quad (22)$$

Where $g_i$ is a binary digit and $m$ is a positive integer larger than one, which determines the precision of encoding. The value of $m$ is 5 in this paper.

The decoding of $Chromosome 2$ is shown in Equation (23)–(25). The decoded $Chromosome 2$ means a feasible solution of priority matrix of all the products to be assembled in the stations.

$$\begin{align*}
N_1 &= Dec(g_1, g_2, \ldots, g_m) \\
N_2 &= Dec(g_{m+1}, g_{m+2}, \ldots, g_{2m}) \\
N_3 &= Dec(g_{2m+1}, g_{2m+2}, \ldots, g_{3m}) \\
\end{align*} \quad (23)$$

$$\begin{align*}
u_1 &= \frac{N_1}{(N_1 + N_2 + N_3)} \\
u_2 &= \frac{N_2}{(N_1 + N_2 + N_3)} \\
u_3 &= \frac{N_3}{(N_1 + N_2 + N_3)} \quad (24)$$

$$Priority Matrix = Pr_{ijk}(u_1, u_2, u_3), \quad i \in I, \; j \in J, \; k \in K \quad (25)$$

Where $N_1, N_2, N_3$ are three decimal numbers and they are not zeros at the same time. $u_1, u_2, u_3$ are three priority weights. The calculation formula of $Pr_{ijk}(u_1, u_2, u_3)$ is shown in previous Equation (17). $I$ is the set of planned products, $J$ is the set of operations or stages and $K$ is set of stations.

2) Fitness function
The calculation of fitness function $F$ adopts the FAHP [43], which uses the expert scores to normalize the multiple objectives, as shown in previous Equation (19). The optimization direction of WSP-IGA algorithm is to minimizing the fitness function $F$. Less the $F$ is, the higher the quality of prescheduling solution is.

3) Parent selection and offspring selection
The parent selection adopts the strategies of roulette wheel and best solution clone, while the offspring selection adopts the ‘1 of 4’ strategy. The roulette wheel [44] refers that the probabilities of individuals being selected as the parents are proportional to their fitness. The best solution clone [44] refers that the individuals with the best fitness must be selected as the parents. ‘1 of 4’ strategy [44] refers that the best individuals from the parents and offspring in the current generation are selected as the new generation.

4) Crossover and mutation

The crossover operator adopts the order crossover (OX), while the mutation operator adopts the inversion mutation. The OX [44] refers that a certain number of chromosomes are selected and crossed over in pairs according to the crossover probability ($P_c$), and the fragments between two random points of the pairwise chromosomes are exchanged. The inversion mutation [44] refers that a certain number of chromosomes are selected and mutated according to the mutation probability ($P_m$), and the random fragment of the chromosomes are inversed.

5) Iterative evolution

The WSP-IGA termination condition is that the iteration number reaches the generation number. In order to expand the search scope and avoid the prematurity, the initial values of $P_c$ and $P_m$ are set high ($P_c=0.8, P_m=0.1$). In order to preserve the excellent chromosome structures and accelerate the convergence rates, the $P_c$ and $P_m$ decrease 0.001 in each iteration, respectively.

D. RESCHEDULING ALGORITHM

According to the characteristics of FMMAL, the change of product sequence is relatively difficult, while the change of priority matrix is relatively easy. Therefore, combining the priority weights search with the simulated annealing, the PWS-SA is proposed here. The flowchart of the PWS-SA algorithm is shown as Figure 5. When a disruption event occurs, including the station failure, product inserting and product reworking, the PWS-SA will be triggered to search for the priority weights again, which obtains the rescheduling solution to improve the quality of prescheduling solution.

1) Priority weights of prescheduling solution

The starting point of PWS-SA are the priority weights of prescheduling solution $\mathbf{u} = (u_1, u_2, u_3)$ and the corresponding fitness $F = F(\mathbf{u})$ after a certain disruption event occurs. The parameters of initial temperature $T_0$, ending temperature $T_{min}$, Markov chain length $L_{max}$ and cooling coefficient $K$ are set here.

2) Generating the new priority weights by disturbance

The random disturbance integer $r$ and cooling disturbance rate $\theta$ are used to add a random vector $\Delta \mathbf{u}(r, \theta)$ to the old priority weights, so as to obtain the new priority weights $\mathbf{u}' = \mathbf{u} + \Delta \mathbf{u}(r, \theta)$ and the corresponding new fitness $F' = F(\mathbf{u}')$. The specific methods are shown in Equation (26), (27).

$$\Delta \mathbf{u}(r, \theta) = \left(\Delta u_1(r, \theta), \Delta u_2(r, \theta), \Delta u_3(r, \theta)\right)$$

$$= \begin{cases} 
-\theta u_1, +\theta u_3, 0, & \text{if } r = 1 \\
-\theta u_1, 0, +\theta u_1, & \text{if } r = 2 \\
+\theta u_2, -\theta u_2, 0, & \text{if } r = 3 \\
0, -\theta u_2, +\theta u_2, & \text{if } r = 4 \\
+\theta u_3, 0, -\theta u_3, & \text{if } r = 5 \\
0, +\theta u_3, -\theta u_3, & \text{if } r = 6 
\end{cases}$$

3) Judging whether the new priority weights can be accepted or not

If the new fitness is less than the old fitness ($F' < F$), the new priority weights and fitness can be accepted ($\mathbf{u} = \mathbf{u}'$, $F = F'$). If the new fitness is larger than the old fitness ($F' \geq F$), the new priority weights and fitness need to be accepted by the Metropolis rule [45], as shown in Equation (28).

$$\mathbf{u} = \mathbf{u}', F = F', \text{ if } e^{-\frac{F' - F}{T}} > p, \text{ random } p \in [0,1)$$

4) Cooling loops and Markov loops
At the temperature $T$, the two steps of 2) generating the new priority weights by disturbance and 3) judging whether the new priority weights can be accepted or not need to enter the cooling loops. While $T > T_{min}$, $T = T \cdot K$ in each cooling loop. Then the two steps need to enter the Markov chain loops. While $L \leq L_{max}$, $L=L+1$ in each Markov chain loop.

5) Priority weights of rescheduling solution

After the whole loops are ended, the priority weights of rescheduling solution $u = (u_1, u_2, u_3)$ and the corresponding fitness $F = F(u)$ after a certain disruption event occurs can be accepted at the ending point.

E. METHODOLOGY

The flowchart of the IoMT dynamic scheduling optimization are shown as Figure 6.

Step 1: The EIS assigns the order tasks and planning to the MAS.

Step 2: The MAS is set with the objectives, namely minimizing the $C_{max}$, $TET$ and $TEC$. And the objectives are normalized via the FAHP.

Step 3: The MAS is imported with the constraints, namely the data of products, processes and resources.

Step 4: The MAS generates an optimal prescheduling solution including the product sequence and priority matrix via the WSP-IGA.

Step 5: The MAS controls the operations and monitors the states of ubiquitous shop floor in real time.

Step 6: When a disruption event occurs, the MAS generates an optimal rescheduling solution which just updating the priority matrix via the PWS-SA. Then the procedure returns to step 5.

Step 7: Until all the order tasks are completed, the MAS feeds back the scheduling results to the EIS.

VI. CASE STUDY AND DISCUSSION

A. CASE DESCRIPTION

To demonstrate the practicability and validity of the IoMT-based FMMAL architecture, WSP-IGA algorithm and PWS-SA algorithm in the dynamic scheduling, this paper applies the proposed theories into a numerical case of assembly process of modular products in MC. The studied numerical case of a FMMAL with three stages, three stations in each stage and two AGVs for ten planned products in four production modes is described as a complicated problem (10 products × 3 stages × 3 stations × 2 AGVs × 4 modes). To our best knowledge, there is not any benchmark data available for this problem. So, some real-life industrial data from a China non-standard equipment manufacturer are used to generate a data set. And the upper and lower boundaries of these data from this manufacturer EIS are taken to generate the data set via random uniform distribution to increase the universality. The numerical parameters of this data set are as follows: $AT_{ijk} \sim U(5, 15)$ min, $d_{ij} \sim U(25, 50)$ min, $\overline{p}_{ijk} \sim U(10.0, 30.0)$ kW, $p_{idle} \sim U(1.0, 3.0)$ kW, $\overline{p}_{k}^{\text{iflare}} \sim U(5.0, 15.0)$ kW, $C_{max} \in [48.5, 76.6]$ min, $TET \in [30.8, 170.4]$ min, $TEC \in [83.6, 149.2]$ kWh, $\overline{p}_{0} = 10$ kW, $d_{ij} \sim U(3.0, 6.0)$ rounded, $n_{d} = 2$, $a_{d} = 5$, $v_{d} = 0.2$ m/s, $D_{1} = 10$ m, $D_{2} = 20$ m, $D_{3} = 30$ m, $Q_{t}^{\text{iflare}} = 30$, initial $Q_{t} = 15$. The Siemens Tecnomatix Plant Simulation 13.1 package is employed to conduct a discrete event simulation [46] of the FMMAL, as shown in Figure 7.

The various plant simulation objects are used to build the model frame of the FMMAL with three DSUs, while the movable units are used to represent the products, parts and AGVs. The attributes and methods of various agents in the DSUs are written by the Simtalk script language of this package. The running environment of the simulation is a Lenovo ThinkPad E570 computer with an Intel Core i5-7200U CPU and an 8.00 GB RAM.

The normalized weights of fitness in the four production modes are shown in previous Table 1. The process properties of the products in the different stations are shown in Table 2. The required parts of the different operations of the products are shown in Table 3, namely the BOM. The average idle powers and average failure powers of the stations are shown in Table 4.
Figure 7. Discrete event simulation model of the FMMAL.

Table 2. Process properties of the products in the different stations.

| Operation | Assembly time $t_A$ (min) | Due time $t_D$ (min) | Average assembly power $\beta$ (kW/min) |
|-----------|---------------------------|---------------------|--------------------------------------|
| Station   | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 |
| OP1       | 6.99/48.59/23.62 | 5.46/48.59/23.52 | 9.30/48.59/21.17 | 9.82/48.59/18.38 | 7.65/48.59/16.20 | 12.00/48.59/15.60 | 11.67/48.59/18.94 | 11.05/48.59/22.07 | 7.36/48.59/10.38 |
| OP2       | 6.10/38.16/19.26 | 5.67/38.16/22.87 | 13.90/38.16/24.59 | 10.13/38.16/22.77 | 8.48/38.16/29.85 | 5.11/38.16/15.76 | 15.80/38.16/14.01 | 11.92/38.16/15.75 | 11.06/38.16/29.77 |
| OP3       | 12.46/27.99/23.99 | 11.07/27.99/14.69 | 12.97/27.99/29.95 | 12.07/27.99/17.49 | 14.58/27.99/21.84 | 21.08/27.99/26.46 | 14.65/27.99/24.79 | 8.87/27.99/10.10 | 12.39/27.99/25.46 |
| Planned product | 12.76/27.99/23.99 | 11.87/27.99/14.69 | 12.97/27.99/29.95 | 12.07/27.99/17.49 | 14.58/27.99/21.84 | 21.08/27.99/26.46 | 14.65/27.99/24.79 | 8.87/27.99/10.10 | 12.39/27.99/25.46 |

Table 3. Required parts of the different operations of the products.

| Product | OP1 | OP2 | OP3 |
|---------|-----|-----|-----|
| Part1 (pcs) | A  | B  | C  | D  | E  | F  | G  | H  | I  | J  | K  | L  |
| 3      | 5   | 4   | 5   | 5   | 4   | 5   | 3   | 5   | 4   |
| OP2    | Part2 (pcs) | 5   | 5   | 5   | 4   | 5   | 3   | 6   | 4   | 5   |
| OP3    | Part3 (pcs) | 5   | 5   | 5   | 4   | 4   | 6   | 5   | 6   |

Table 4. Average idle powers and average failure powers of the stations.

| Station   | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 |
|-----------|----|----|----|----|----|----|----|----|----|
| Average idle power (kW) | 1.78 | 1.36 | 1.01 | 2.90 | 2.03 | 1.55 | 2.81 | 1.20 | 1.87 |
| Average failure power (kW) | 14.81 | 6.99 | 10.82 | 8.77 | 11.47 | 6.16 | 9.26 | 12.69 | 11.89 |

B. PRESCHEDULING RESULTS

In the four production modes of synthesized mode, time-efficient mode, just-in-time mode and energy-saving mode, the WSP-IGA algorithm is used to obtain the prescheduling solutions of the FMMAL. And its population size is 30, generation number is 100, $P_c$ is 0.8, and $P_m$ is 0.1. With the same parameters, four traditional meta-heuristics [42] are used as the compared groups of WSP-IGA, including the shortest processing time-simple genetic algorithm (SPT-SGA), first come first serve-simple genetic algorithm (FCFS-SGA), earliest due date-simple genetic algorithm (EDD-NGA) and longest processing time-simple genetic algorithm (LPT-NGA).

Their convergence curves are shown as Figure 8, and their performance comparisons are shown as Table 5 and Figure 9.
FIGURE 8. Convergence curves of the prescheduling algorithms: (a) Synthesized mode, (b) Time-efficient mode, (c) Just-in-time mode and (d) Energy-saving mode.

TABLE 5. Performance comparisons of the prescheduling algorithms.

| Mode                | LPT-SGA | EDD-SGA | FCFS-SGA | SPT-SGA | WSP-IGA |
|---------------------|---------|---------|----------|---------|---------|
| Synthesized mode    | 0.441/976.636 | 0.2586/58.296 | 0.1296/38.386 | 0.1578/78.450 | 0.095/90.767 |
| Time-efficient mode | 0.4363/83.939 | 0.2316/59.077 | 0.1573/52.832 | 0.1222/79.385 | 0.0731/94.753 |
| Just-in-time mode   | 0.4134/84.868 | 0.2238/65.887 | 0.1119/47.183 | 0.1400/80.709 | 0.0704/98.961 |
| Energy-saving mode  | 0.4186/75.799 | 0.2516/55.680 | 0.1568/36.986 | 0.1265/70.327 | 0.0729/93.934 |
| Mean                | 0.4275/80.310 | 0.2414/59.735 | 0.1339/46.134 | 0.1366/77.218 | 0.0779/94.612 |
| Standard deviation  | 0.0137/4.753  | 0.0164/4.351  | 0.0223/2.780  | 0.0160/4.687  | 0.0115/3.376  |

As shown in Figure 8, in the four production modes, the best fitness values of WSP-IGA are less than those of the LPT-SGA, EDD-SGA, FCFS-SGA and SPT-SGA, and the convergence rates of WSP-IGA are relatively fast. Besides, as shown in Figure 9 and Table 5, the mean value of best fitness of WSP-IGA is respectively 18.22%, 32.27%, 56.08% and 57.03% of those of the four referenced algorithms and the best fitness fluctuations of WSP-IGA are relatively small. Therefore, WSP-IGA can obtain much better prescheduling solutions and its search performance is much better than those of the four referenced algorithms. However, the running time values of WSP-IGA are more than those of the four referenced algorithms. The mean value of running time of WSP-IGA is respectively 1.18, 1.58, 2.05 and 1.23 times as much as those of the four referenced algorithms and the running time fluctuations of WSP-IGA are relatively small. Therefore, the running rate of WSP-IGA is slightly slower than those of the four referenced algorithms. Generally, with small loss of running rate, the WSP-IGA is superior to the four referenced algorithms to obtain much better prescheduling solutions.

In the four production modes, the Gantt charts and results of the prescheduling solutions with the WSP-IGA algorithm are shown as Figure 10 and Table 6, respectively. In the synthesized mode, the $C_{max}$, TET and TEC are relatively balanced but not the minimum values compared to the other modes. In the time-efficient mode, the $C_{max}$ is the minimum value 49.47 min of all the modes. In the just-in-time mode,
the $TET$ is the minimum value 31.44 min of all the modes. In the energy-saving mode, the $TEC$ is the minimum value 85.35 kWh of all the modes. The priority weights $(u_1, u_2, u_3)$ of the four production modes are different between each other.

Therefore, the WSP-IGA can realize the prescheduling of the FMMAL in the four production modes and obtain its prescheduling solutions successfully.

![Gantt charts of the prescheduling solutions with the WSP-IGA algorithm: (a) Synthesized mode, (b) Time-efficient mode, (c) Just-in-time mode and (d) Energy-saving mode.](image)

**TABLE 6.** Results of the prescheduling solutions with the WSP-IGA algorithm.

| Mode                | $C_{max}$ (min)/$u_1$ | $TET$ (min)/$u_2$ | $TEC$ (kWh)/$u_3$ | Best fitness |
|---------------------|------------------------|-------------------|-------------------|--------------|
| Synthesized mode    | 51.61/0.4737           | 47.28/0.4737       | 87.32/0.0526      | 0.0951       |
| Time-efficient mode | 49.47/0.1714           | 73.43/0.5143       | 96.66/0.3143      | 0.0731       |
| Just-in-time mode   | 55.13/0.5000           | 31.43/0.4231       | 96.12/0.0769      | 0.0704       |
| Energy-saving mode  | 51.87/0.0000           | 74.61/0.5789       | 85.35/0.4211      | 0.0729       |

C. RESCHEDULING RESULTS

In the four production modes of synthesized mode, time-efficient mode, just-in-time mode and energy-saving mode, the PWS-SA algorithm is used to obtain the rescheduling solutions of the FMMAL. And its initial temperature $T_0$ is 200, ending temperature $T_{min}$ is 0.1, length of Markov chain $L$ is 10 and cooling coefficient is 0.95. A total of six disruption events are set, including two station failure events as shown in Table 7, two product inserting events as shown
in Table 8 and two product reworking events as shown in Table 9. When each of the disruption events occurs, the PWS-SA algorithm is triggered to search for the priority weights again and generates a new optimal rescheduling solution to improve the prescheduling solution or old rescheduling solution. With the same parameters, four traditional meta-heuristics [42] are used as the compared groups of PWS-SA, including LPT-SGA, EDD-SGA, FCFS-SGA and SPT-SGA, which do not trigger the priority weights search. Their comparison charts of best fitness are shown as Figure 11, their average best fitness values are shown in Table 10 and the running times of the PWS-SA algorithm are shown in Table 11.
As shown in Figure 11 and Table 10, in the four production modes, the disruption events have a bad effect on the best fitness values of FMMAL scheduling solutions and they generally degrade the qualities of the solutions, although they occasionally do not change or even improve the qualities slightly. The best fitness curves of PWS-SA are always below those of the LPT-SGA, EDD-SGA, FCFS-SGA and SPT-SGA and the overall average value of PWS-SA is respectively 45.69%, 56.20%, 62.65% and 78.73% of those of the four referenced algorithms. Therefore, when a disruption event occurs, the rescheduling solutions generated by the PWS-SA are superior to those of the four referenced algorithms and they can offset the effects of disruption events to some extent. In the four referenced algorithms, the adaptive scheduling solution of SPT-SGA have the best robustness, while the adaptive scheduling solution of LPT-SGA have the worst robustness. As shown in Table 11, the running times of PWS-SA are relative short and the average running times of PWS-SA for rescheduling in the four production modes are respectively 61.70%, 59.69%, 58.64% and 58.72% of those of WSP-IGA for prescheduling. Therefore, the running rate of PWS-SA is relatively fast and it is very suitable to the rescheduling occasions which need some algorithms with fast running rates.

In the four production modes, the final Gantt chart and results of the rescheduling solutions with the PWS-SA algorithm after the six disruption events occurred are shown as Figure 12 and Table 12, respectively.

FIGURE 12. Gantt charts of the rescheduling solutions with the PWS-SA algorithm: (a) Synthesized mode, (b) Time-efficient mode, (c) Just-in-time mode and (d) Energy-saving mode.
TABLE 12. Final results of the rescheduling solutions with the PWS-SA algorithm.

| Mode                  | $C_{\text{max}}$ (min)/$u_1$ | $TET$ (min)/$u_2$ | $TEC$ (kWh)/$u_3$ | Best fitness |
|-----------------------|-------------------------------|-------------------|-------------------|-------------|
| Synthesized mode      | 65.24/0.3750                 | 113.39/0.2917     | 124.75/0.3333     | 0.6052      |
| Time-efficient mode   | 63.74/0.2857                 | 143.49/0.4857     | 123.62/0.2286     | 0.5716      |
| Just-in-time mode     | 68.01/0.3415                 | 93.05/0.4146      | 121.11/0.2439     | 0.5060      |
| Energy-saving mode    | 70.70/0.0385                 | 134.76/0.6538     | 114.92/0.3077     | 0.5504      |

The station failure, product inserting and product reworking are well addressed and brought into the rescheduling solutions. Finally, in the synthesized mode, the $C_{\text{max}}$, $TET$ and $TEC$ are relatively balanced but not the minimum values compared to the other modes. In the time-efficient mode, the $C_{\text{max}}$ is the minimum value 63.74 min of all the modes. In the just-in-time mode, the $TET$ is the minimum value 93.05 min of all the modes. In the energy-saving mode, the $TEC$ is the minimum value 114.92 kWh of all the modes. The priority weights ($u_1$, $u_2$, $u_3$) of the four production modes in the rescheduling are different between each other and also different from those in the prescheduling. Therefore, the PWS-SA can realize the rescheduling of the FMMAL in the four production modes and obtain its rescheduling solutions successfully.

VII. CONCLUSIONS AND FUTURE WORKS

In MC, the assembly operations of modular products tend to be organized as the form of FMMAL. In this study, firstly, the IoMT-based FMMAL are constructed via the MAS and ubiquitous environment. Secondly, a mathematical model of FMMAL are formulated with the decision variables, optimization objectives and constraint conditions. Thirdly, the IoMT-oriented algorithms for dynamic scheduling are proposed including the FAHP for normalization, WSP-IGA for prescheduling, PWS-SA for rescheduling. Lastly, a discrete event simulation of a numerical case is conducted to demonstrate the practicality and validity of proposed theories and algorithms. The main conclusions are as follows. 1) The IoMT-based FMMAL with the DSUs are the foundation of the dynamic scheduling optimization. 2) At the beginning, the WSP-IGA manages to realize the prescheduling of FMMAL in the four production modes, which is superior to the referenced algorithms and obtains much better solutions with small loss of running rate. 3) When the disruption events occur, the PWS-SA manages to realize the rescheduling of FMMAL in the four production modes, which is superior to the referenced algorithms and can partly offset the effects of the disruption events on scheduling solutions with a relatively fast running rate.

The future works will focus on the following aspects. 1) The technologies of machine learning and deep learning should be investigated to analyze and predict the states of devices, which will help the decisions of production and maintenance. 2) The technologies of cloud computation and edge computation should be employed to increase the computing powers, which will help the dynamic scheduling with the Pareto solution set and more complicated algorithms. 3) The supply chain and warehouse logistics management in MC should be studied to form a complete IoMT ecosystem, which will help the extension of intelligent manufacturing.

REFERENCES

[1] L. Li, “China’s manufacturing locus in 2025: With a comparison of “Made-in-China 2025” and “Industry 4.0”,” Technol. Forecast. Soc. Chang., vol. 135, pp. 66-74, 2018.
[2] L. S. Dalenogare, G. B. Benitez, N. F. Ayala, and A. G. Frank, “The expected contribution of Industry 4.0 technologies for industrial performance,” Int. J. Prod. Econ., vol. 204, pp. 383-394, 2018.
[3] D. A. Rossit, F. Tohmé, and M. Frutos, “Industry 4.0: smart scheduling,” Int. J. Prod. Res., vol. 57, no. 12, pp. 3802-3813, 2019.
[4] L. X. Xu, E. L. Xu, and L. Li, “Industry 4.0 state of the art and future trends,” Int. J. Prod. Res., vol. 56, no. 8, pp. 2941-2962, 2018.
[5] B. Esmaeilian, S. Behdad, and B. Wang, “The evolution and future of manufacturing: A review,” J. Manuf. Syst., vol. 39, pp. 79-100, 2016.
[6] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, “Smart factory of industry 4.0: Key technologies, application case, and challenges,” IEEE Access, vol. 6, pp. 6505-6519, 2018.
[7] L. D. Xu, W. He, and S. Li, “Internet of things in industries: A survey,” IEEE Trans. Ind. Informat., vol. 10, no. 4, pp. 2233-2243, Nov. 2014.
[8] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, “Industrial internet of things: Challenges, opportunities, and directions,” IEEE Trans. Ind. Informat., vol. 14, no. 11, pp. 4724-4734, Nov. 2018.
[9] Y. Zhang, W. Wang, S. Liu, and G. Xie, “Real-time shop-floor production performance analysis method for the internet of manufacturing things,” Adv. Mech. Eng., vol. 6, pp. 270749, 2014.
[10] Y. Zhang, G. Zhang, J. Wang, S. Sun, S. Si, and T. Yang, “Real-time information capturing and integration framework of the internet of manufacturing things,” Int. J. Comput. Integr. Manuf., vol. 28, no. 8, pp. 811-822, 2015.
[11] S. Tian, T. Wang, L. Zhang, and X. Wu, “An energy-efficient scheduling approach for flexible job shop problem in an internet of manufacturing things environment,” IEEE Access, vol. 7, pp. 62695-62704, 2019.
[12] D. Ouelhadj, and S. Petrovic, “A survey of dynamic scheduling in manufacturing systems,” J. Scheduling, vol. 12, no. 4, pp. 417, 2009.
[13] J. Jerald, P. Asokan, G. Prabaharan, and R. Saravanam, “Scheduling optimisation of flexible manufacturing systems using particle swarm optimisation algorithm,” Int. J. Adv. Manuf. Tech., vol. 25, no. 9, pp. 964-971, 2005.
[14] P. Fattahi, and A. Fallahi, “Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability,” CIRP J. Manuf. Sci. Tech., vol. 2, no. 2, pp. 114-123, 2010.
[15] C. W. Pickardt, T. Hildebrandt, J. Branke, J. Heger, and B. Scholz-Reiter, “Evolutionary generation of dispatching rule sets for complex dynamic scheduling problems,” Int. J. Prod. Econ., vol. 145, no. 1, pp. 67-77, 2013.
[16] S. Nguyen, M. Zhang, M. Johnston, and K. C. Tan, “Automatic programming via iterated local search for dynamic job shop scheduling,” IEEE Trans. Cybern., vol. 45, no. 1, pp. 1-14, Jan. 2015.
[17] D. Tang, M. Dai, M. A. Salido, and A. Giret, “Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimisation algorithm,” Comput. Ind., vol. 81, pp. 82-95, 2016.
[18] F. Liu, S. Wang, Y. Hong, and X. Yue, “On the robust and stable flowshop scheduling under stochastic and dynamic disruptions,” IEEE Trans. Eng. Manag., vol. 64, no. 4, pp. 539-553, Nov. 2017.
[19] S. Zhang, and S. Wang, “Flexible assembly job-shop scheduling with sequence-dependent setup times and part sharing in a dynamic

17
environment: Constraint programming model, mixed-integer programming model, and dispatching rules,” IEEE Trans. Eng. Manag., vol. 65, no. 3, pp. 487-504, Aug. 2018.

[20] Y. Zhou, J. Yang, and L. Zheng, “Hyper-heuristic coevolution of machine assignment and job sequencing rules for multi-objective dynamic flexible shop scheduling,” IEEE Access, vol. 7, pp. 68-88, 2019.

[21] N. Boysen, M. Fleddner, and A. Scholl, “Sequencing mixed-model assembly lines: Survey, classification and model critique,” Eur. J. Oper. Res., vol. 192, no. 2, pp. 349-373, 2009.

[22] O. S. Akgündüz, and S. Tunali, “A review of the current applications of genetic algorithms in mixed-model assembly line sequencing,” Int. J. Prod. Res., vol. 49, no. 15, pp. 4483-4503, 2011.

[23] N. Boysen, A. Scholl, and N. Wopperer, “Resequencing of mixed-model assembly lines: Survey and research agenda,” Eur. J. Oper. Res., vol. 216, no. 3, pp. 594-604, 2012.

[24] C. Öztürk, S. Tunali, B. Hнич, and A. Örnek, “Balancing and scheduling of flexible mixed model assembly lines with parallel stations,” Int. J. Adv. Manuf. Tech., vol. 67, no. 9-12, pp. 2577-2591, 2013.

[25] C. Öztürk, S. Tunali, B. Hнич, and A. Örnek, “Cyclic scheduling of flexible mixed model assembly lines with parallel stations,” J. Manuf. Syst., vol. 36, pp. 147-158, 2015.

[26] M. Wang, R. Y. Zhong, Q. Dai, and G. Q. Huang, “A MPN-based scheduling model for IoT-enabled hybrid flow shop manufacturing,” Adv. Eng. Inform., vol. 30, no. 4, pp. 728-736, 2016.

[27] R. Ruiz, and J. A. Vázquez-Rodríguez, “The hybrid flow shop scheduling problem,” Eur. J. Oper. Res., vol. 205, no. 1, pp. 1-18, 2010.

[28] J. Du, Y. Y. Jiao, and J. Jiao, “Integrated BOM and routing generator for variety synchronization in assembly-to-order production,” J. Manuf. Technol. Manage., vol. 16, no. 2, pp. 233-243, 2005.

[29] Y.-C. Ho, and S.-H. Chien, “A simulation study on the performance of task-determination rules and delivery-dispatching rules for multiple-load AGVs,” Int. J. Prod. Res., vol. 44, no. 20, pp. 4193-4222, 2006.

[30] D. Rahmani, and R. Ramezanian, “A stable reactive approach in dynamic flexible flow shop scheduling with unexpected disruptions: A case study,” Comput. Ind. Eng., vol. 98, pp. 360-372, 2016.

[31] A. Dorri, S. S. Kanhere, and R. Jurdak, “Multi-agent manufacturing systems: A survey,” IEEE Access, vol. 6, pp. 28573-28593, 2018.

[32] W. Xiang, and H. P. Lee, “Ant colony intelligence in multi-agent dynamic manufacturing scheduling,” Eng. Appl. Artif. Intell., vol. 21, no. 1, pp. 73-85, 2008.

[33] R. Erol, C. Sahin, A. Baykasoglu, and V. Kaplanoglu, “A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems,” Appl. Soft Comput., vol. 12, no. 6, pp. 1720-1732, 2012.

[34] Z. Jiang, Y. Jin, E. Mingheng, and Q. Li, “Distributed dynamic scheduling for cyber-physical production systems based on a multi-agent system,” IEEE Access, vol. 6, pp. 1855-1869, 2018.

[35] F. Tao, and Q. Qi, “New IT driven service-oriented smart manufacturing: framework and characteristics,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 49, no. 1, pp. 81-91, 2017.

[36] J. Wang, Y. Zhang, Y. Liu, and N. Wu, “Multiagent and bargaining-game-based real-time scheduling for internet of things-enabled flexible job shop,” IEEE Internet Things J., vol. 6, no. 2, pp. 2518-2531, April 2019.

[37] Y. Feng, Q. Wang, Y. Gao, J. Cheng, and J. Tan, “Energy-efficient job-shop dynamic scheduling system based on the cyber-physical energy-monitoring system,” IEEE Access, vol. 6, pp. 52238-52247, 2018.

[38] Y. Fang, C. Peng, P. Lou, Z. Zhou, J. Hu, and J. Yan, “Digital-Twin-Based Job Shop Scheduling Toward Smart Manufacturing,” IEEE Trans. Ind. Informat., vol. 15, no. 12, pp. 6425-6435, Dec. 2019.

[39] B. Costa, P. F. Pires, and F. C. Delicato, “Towards the adoption of OMG standards in the development of SOA-based IoT systems,” J. Syst. Software, vol. 169, pp. 110720, 2020.

[40] L. Shi, G. Guo, and X. Song, “Multi-agent based dynamic scheduling optimisation of the sustainable hybrid flow shop in a ubiquitous environment,” Int. J. Prod. Res., pp. 1-22, 2019.

[41] F. Tonelli, M. Paolucci, D. Anghinolfi, and P. Taticchi, “Production planning of mixed-model assembly lines: a heuristic mixed integer programming based approach,” Prod. Plan. Control, vol. 24, no. 1, pp. 110-127, 2013.

[42] S. Yang, Y. Jiang, and T. T. Nguyen, “Metaheuristics for dynamic combinatorial optimisation problems,” IMA J. Manag. Math., vol. 24, no. 4, pp. 451-480, Oct. 2013.

[43] S. Kuhler, J. Robert, W. Derigent, A. Voisin, and Y. Le Traon, “A state-of-the-art survey & testbed of fuzzy AHP (FAHP) applications,” Expert Syst. Appl., vol. 65, pp. 398-422, 2016.

[44] M. Affenzeller, S. Wagner, S. Winkler, and A. Beham, Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications. Boca Raton, FL, USA: CRC Press, 2009.

[45] T. Loukil, J. Teghem, and P. Fortemps, “A multi-objective production scheduling case study solved by simulated annealing,” Eur. J. Oper. Res., vol. 179, no. 3, pp. 709-722, 2007.

[46] S. S. Padhi, S. M. Wagner, T. T. Niranjan, and V. Aggarwal, “A simulation-based methodology to analyse production line disruptions,” Int. J. Prod. Res., vol. 51, no. 6, pp. 1885-1897, 2013.

LEI SHI received the B.S. degree in process equipment and control engineering from Xi’an Jiaotong University, Xi’an, Shaanxi, China, in 2011, and the M.S. degree in chemical process machinery from Tianjin University, Tianjin, China, in 2014. He is currently pursuing the Ph.D. degree in mechanical engineering with the School of Mechanical Engineering, Chongqing University, Chongqing, China. His current research interests include manufacturing systems engineering, intelligent manufacturing and production scheduling.

GANG GUO received the B.S., M.S., and Ph.D. degrees in automotive engineering from Chongqing University, Chongqing, China, in 1982, 1984, and 1994, respectively. He is currently the Chair and a Professor with the School of Automotive Engineering, Chongqing University. He is also an Associate Director of the Chongqing Auto-motive Collaborative Innovation Center. He has authored or coauthored over 100 refereed journal and conference publications. His research interests include connected vehicles, multi-sense perception, human–machine interface, brain–computer interface, intelligent manufacturing, and user experience. He is a Senior Member of the China Mechanical Engineering Society. He is also a member of the China User Experience Alliance Committee. He is also the Director of the China Automotive Engineering Society.