IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 20, 2020, accepted January 29, 2020, date of publication February 10, 2020, date of current version February 18, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2972914*

# Study on Edge-Cloud Collaborative Production Scheduling Based on Enterprises With Multi-Factory

**JING MA[1], HUA ZHOU[1], CHANGCHUN LIU[2], MINGCHENG E[1], ZENGQIANG JIANG[1], AND QIANG WANG[3]**

[1]School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China
[2]Institute of Operations Research and Analytics, National University of Singapore, Singapore 117602
[3]Department of Industrial Engineering, Tsinghua University, Beijing 100084, China

Corresponding author: Mingcheng E (emch@bjtu.edu.cn)

This work was supported in part by the National Science Foundation of China under Grant 51775033, and in part by the Fundamental Research Funds for the Central Universities of China under Grant 2019RC023.

**ABSTRACT** The depth development and widespread application of edge intelligence technology based on the Internet of Things has led to edge-cloud collaboration and related research. In recent years, with the rapid development of the Internet of Things and the formation of super-city groups, the management characteristics of enterprises with multiple manufacturing plants served for headquarters have become increasingly obvious. The problem of order dynamic fluctuations caused by personalized customization requirements has become more prominent, which makes it impossible to do global long-period prediction or real-time short-period response relied solely on the cloud or edge. Therefore, this paper proposes a production system scheduling framework under the edge-cloud collaborative paradigm based on the dynamic fluctuation of orders under these background, and builds an edge-cloud collaborative scheduling model, which guarantees real-time distributed scheduling at the edge. It enabled the cloud to periodically predict the total completion time of production tasks at the headquarters based on the value-added data uploaded by the edge, and to support more accurate and efficient scheduling at the edge based on the prediction results. Finally, an example analysis proved the rationality of the scheduling mechanism and the effectiveness of the scheduling model. The proposed method can provide a certain reference for task scheduling in the edge-cloud collaborative production paradigm.

**INDEX TERMS** Production scheduling, edge-cloud collaboration, edge computing, Internet of Things.

## I. INTRODUCTION

Since the first industrial revolution, production paradigms have been emerged and developed with the changes of market demand and advancing of technology. In recent years, with the development and deep integration of cloud computing, Internet of Things (IoT), big data, service-oriented, and other advanced technologies, cloud manufacturing has emerged and developed into an emerging networked manufacturing mode that integrates various manufacturing resources [1]. It is mapped as a cloud service, and provides users with cloud manufacturing services on demand, which makes up for the shortcomings of existing manufacturing modes [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Chua Chin Heng Matthew.

However, with the depth development and application of increasing edge intelligence technologies, some intelligent features such as interconnected perception of manufacturing resources, embedded computing, and distributed control in factories have become more and more significant. Edge-side manufacturing data is exploding, but cloud data centers are centralized, and it is difficult to meet the storage and processing requirements of a large amount of real-time data from the manufacturing edge, which leads to network congestion, high latency, low quality of service, and data privacy and security, etc. In order to provide better quality of service to the manufacturing edge, cloud services need to migrate to the edge. In recent years, the emergence and development of edge computing technology has made up for the defects of cloud manufacturing mode in some application scenarios, but it is unable to provide enterprises with better value-added services due to a small range of covered resources and limited optimization. Therefore, exerting the value of the edge-cloud collaborative (ECC) and its production paradigm, and providing enterprises with better value-added services and satisfy the actual needs of more production scenarios have gradually become important issues [3].

Although the application of advanced technologies improves the intelligence of production systems, it also increased the system complexity. Meanwhile, the application also makes the dynamic and uncertain characteristics of production systems more prominent, and makes production scheduling more difficult. For production scheduling issues, the research for manufacturing edge is mainly aimed at deterministic environment with infinite schedulable resources. Since the decision-making system is usually located at the manufacturing edge, the connection between manufacturing resources and scheduling system is closer, and scheduling decision systems can more easily response to emergencies and uncertain events. However, in the cloud manufacturing mode, production tasks have characteristics of large-scale personalized customization [4]. Therefore, influential factors such as the dynamic arrival time of manufacturing tasks, changes in the availability of manufacturing resources, and downtime of manufacturing equipment should be considered. Compared with edge-side manufacturing scheduling, cloud manufacturing scheduling need to consider more practical manufacturing environment. For example, because the scheduling system is located in the remote cloud data centers, and connects with edge-side resources by the internet. Thus, this remote network connection will cause uncertainties and interferences events more frequently, and increase the difficult to control the cloud-side scheduling system. The above-mentioned cloud manufacturing scheduling characteristics can have a great impact on the operation of production systems [5]. Therefore, real-time predictions of interference event and system statue are important issues in the cloud manufacturing environment [6].

In recent years, with the rapid development of the IoT and the formation of super-city groups, the enterprises operation structures with multiple manufacturing factories and a headquarters have become more and more common. Accurate order and resource optimization allocation methods between headquarters and factories are also needed. Secondly, driving by global market changes and industrial revolution, the production paradigm is gradually developing from multiple varieties and small batches to personalized or large-scale personalized customization, which makes the problem of order dynamic fluctuations more prominent. Therefore, in view of the highly decentralized manufacturing resources and data explosion in the IoT environment, cloud-side scheduling or manufacturing edge-side scheduling can no longer effectively support the one headquarters and multi factory business mode and respond to the problem of orders dynamic fluctuation caused by personalized customization demands (research issues). Therefore, the emergence of the ECC production paradigm enables production systems to have the advantages of super-computing and prediction capabilities in the cloud, and real-time response, high service quality, and data security at the manufacturing edge-side [7]. However, compared with cloud manufacturing scheduling or traditional manufacturing scheduling, ECC production systems also face some new scheduling problems. Therefore, in the new ECC production paradigm, how to use the capabilities of real-time response and accurate forecasting to effectively respond to the above-mentioned research issues, has become a hot research topic of manufacturing industry.

In summary, the research object of this paper is the discrete manufacturing production system based on IoT. To address the above research issues, an ECC production scheduling framework and its cooperative scheduling mechanism are presented. Then, an ECC scheduling model is proposed which can achieve the real-time distributed scheduling and cloud-enabled periodically prediction. In the cloud data centers, value-added data is uploaded by the manufacturing edge, and used to periodically predict the completion time of production tasks. The prediction results can be directly used to improve the distributed scheduling accuracy and efficiency of the manufacturing edge. Finally, a case study of an enterprise under the Super Cities Group around Shanghai, China is used to verify the feasibility and effectiveness of the proposed framework.

The remainder of this paper is organized as follows. The framework and mechanism of ECC production scheduling are presented after reviewing the literature. Thereafter, an ECC scheduling model is proposed based on the integration of prediction model and real-time distributed scheduling model. In the ECC scheduling model, several intelligent algorithms are employed to verify the scheduling mechanism. A case study is then discussed to demonstrate ECC production scheduling method. In the last section, conclusions are drawn and the future works are discussed.

## II. LITERATURE REVIEW

This paper focuses on the scheduling problem of the ECC production paradigm. To provide a reasonable research background and research status, the following literature categories are summarized.

### A. SCHEDULING IN CLOUD MANUFACTURING PARA-DIGM

In recent years, the widespread application of IoT increased the transparency and complexity of production systems. Complex manufacturing environments with increased randomness and uncertainty of production systems make the production scheduling more difficult. However, research on service scheduling of cloud manufacturing, especially considering the dynamic characteristics of cloud manufacturing is still in an early stage. Zhou *et al.* [4] proposed an event driven dynamic task scheduling method, which was designed for dynamic scheduling problems of random arrival tasks in the cloud manufacturing environment. Although their research effectively avoided resource preemption and improved the time-validity of the server, it ignored the system uncertainty caused by failure. Wang *et al.* [8] proposed a new hybrid method to effectively solve the multi-agent rescheduling problem in cloud manufacturing, but the method did not work well in a hybrid flexible environment. Cheng *et al.* [9] analyzed the dynamic evolution and statistical characteristics of data in the cloud manufacturing environment, and predicted supply matching in the future to solved this problem for service-oriented manufacturing systems. However, with a large amount of data, dynamic matching of resource services cannot be completed based on historical data or statistical results. Zhou and Zhang [10] studied the resources preemption of dynamic task scheduling in the cloud manufacturing environment, and proposed a real-time simulation based dynamic scheduling method to solve the changes and uncertainties of manufacturing systems, but it their work is deficient in network transmission speed. For achieving real-time data-driven optimization decisions and reduce deviations, Zhang *et al.* [11] presented a real-time allocation strategy to solve the dynamic scheduling optimization problem of flexible job shop, but it is difficult to satisfy the real-time requirements with the increasing of data volume. Obviously, the main difficulty of dynamic scheduling research in the cloud manufacturing environment is that it cannot ensure the real-time scheduling requirement when facing a large amount of data.

There have been many studies and application cases of cloud manufacturing scheduling under the influence of real-time and interference events, but yet there is still lack of researches about production scheduling prediction on the cloud-side. In fact, real-time and accurate prediction of the completion time will be conducive to assign production tasks quickly and reasonably in the manufacturing industry. Therefore, the research on using production data to make periodic predictions for production scheduling has become a trend in recent years. Prediction methods of order completion time can be classified into several categories, such as simulation, statistical analysis, and neural networks. Vinod and Sridharan [12] completed discrete-event simulation and comparison analysis based on different scenarios, which are generated by combining real-time division methods and scheduling rules in a dynamic make-to-order production system. Their research results demonstrated that the dynamic due-date assignment methods provide better performance in solving such problems. Chen [13] used fuzzy C-means method to predict the wafer fabrication cycle time. In their research, a new classification method which embedded the train results of forecasting mechanism into job classifier were proposed. Their method improved the accuracy of the job cycle prediction with long operation time. Chang *et al.* [14] analyzed the historical data of the product life cycle to construct a prediction model of back propagation neural network, which could dynamically improve the weight value and threshold value. In addition, adaptive immune algorithm was used for the optimizing forecast of makespan for an aviation company. Wang and Jiang [15] proposed a deep neural network model to realize order completion time prediction based on order data and work-in-process information collected by radio frequency identification. Their method could relatively achieve high accuracy in a short time and quickly respond to the working load changes of job shop without establishing an accurate analysis model. In summary, in dynamic manufacturing systems with high uncertainty, manufacturing data hides the operating rules of workshop. Compared to simulation technologies, deep learning models can better reflect the actual status of the workshop, and get better predictions. Compared with traditional statistical analysis methods, deep learning methods can extract high levels features from large-scale samples, and obtain valuable knowledge. Meanwhile, deep learning methods also have stronger generalization abilities and can achieve better performance when dealing with big data.

### B. DISTRIBUTED SCHEDULING ON THE EDGE-SIDE

Distributed scheduling is studied based on the distributed manufacturing background, such as cooperative production between different companies or factories. This kind of scheduling focuses on the assignment of work pieces between factories and the processing sequence in each factory to achieve the optimization of scheduling indicators. Since these optimization problems are NP-hard, intelligent algorithms are usually employed to solve them. Overtime, scholars have been using genetic algorithms to solve distributed scheduling problems. In 2005, Chan *et al.* [16] used an improved genetic algorithm to optimize the distributed multi-factory scheduling problem without considering transportation costs. In their research, a superior gene crossover method was proposed to improve the algorithm performance significantly. In 2011, Gao and Chen [17] designed a new genetic algorithm to solve scheduling problems of distributed replacement flow shop. The genetic algorithm improved the crossover and mutation operations, and achieved a better performance compared with the existed local search algorithms. During the past five years, genetic algorithm-based methods have become generalized research methods. For example, to deal with the scheduling under the complex constraints in distributed manufacturing environments, in 2017, Chang and Liu [18] applied Taguchi-enabled hybrid genetic algorithm to study a coding mechanism for the scheduling of distributed flexible job shop, which can resolve the invalid job allocation problem and optimize the makespan. in 2018, Lu *et al.* [19] improved the genetic algorithms performance to solve the scheduling problem of distributed flexible job shop. For achieving better load balancing of scheduling, they developed a one-dimensional to three-dimensional decoding method. Above-mentioned studies illustrate that distributed scheduling problems can be solved by using suitable genetic algorithms. However, researches are mainly focus on independent scenarios at the manufacturing edge-side and insufficient constraints are not considered in existed scheduling models, which is not comprehensive enough to support the ECC production scheduling.

### C. DEVELOPMENT BACKGROUND AND RELATED RESEARCH STATUS OF ECC

The emergence of ECC production paradigm is mainly based on the market demands and technology development.

1) Market demands background: super-city groups will be emerged when cities have matured and developed to a certain stage. The business mode of one headquarters and multi factories emerged and matured based on the network manufacturing technology. To solve a semiconductor production scheduling problem, Dong and Ye [20] presented a gray wolf algorithm to allocate production tasks among multiple heterogeneous factories reasonably to realize a collaborative optimization. Chung *et al.* [21] studied a collaborative strategy for distributed factories and proposed a hybrid genetic algorithm to determine the production plan of each factory. Vieira *et al.* [22] put forward the system integration method to support the application of distributed scheduling in virtual enterprises. In distributed manufacturing environment, above studied scheduling methods are still focuses in the cloud-side without developed and matured manufacturing edge techniques. The super-city groups business mode increases the system dynamic fluctuation, which raises a higher requirement for the production response period. But the single-side scheduling cannot cope with such request. Therefore, cooperative scheduling should be studied based on the edge-side and cloud-side scheduling cases and super-city groups business mode.

2) Technology development background: previously mentioned cloud manufacturing paradigm has emerged and developed with the development and integration of cloud computing, IoT, and other technologies. Resource sharing modes of cloud manufacturing make the data processing more centralized. Although cloud has distributed computation ability by taking advantage of geographically dispersed computing resources, long-distance transmission of the massive and heterogeneous data can cause a series of problems, such as resource conflicts, computing delays, energy consumption increasement, and bandwidth shortage. Wang *et al.* [23] comprehensively analyzed the cloud manufacturing characteristics, and then pointed out that real-time processing of large-scale heterogeneous manufacturing data and efficient data migration between distributed servers will make cloud manufacturing face serious challenges, such as network and data security. Mezgár and Rauschecker [24] pointed out some existed problems of cloud application such as privacy protection and data storage by comparing with the cloud computing architecture and cooperation mechanism of manufacturing enterprises. Therefore, it can be seen that higher demands of real-time, energy saving, and security for computing tasks require the assistance of edge computing to improve the utilization of computing resources. Limited by the computing and data resources of manufacturing edge, so it is difficult to cope with the contradiction between the storage ability and data volume. However, the computing architecture of ECC integrates cloud-side super-computing and real-time, energy saving and security of manufacturing edge-side. ECC architecture can not only has abundant cloud resources, but also can obtain fast, economical and secure data processing capabilities. The emergence of ECC stems from the rapid development of edge technologies. At present, companies from around the world have focused on edge operating systems or servers, edge operation architectures, edge sensors, edge software, and other edge technologies such as the "EC-IoT" and "LiteOS" systems developed by Huawei, the IoT Edge Intelligence Server introduced by Advantech, the Industrial Edge concept proposed by Siemens, a compact edge sensor developed by ABB, and the "AWS Greengrass" released by Amazon.

Driving by ECC technologies, ECC production paradigm has become a certain trend, and it has already been applied in many fields. Moon *et al.* [25] proposed a collaborative framework based on correlating of sample data. The framework could select the best edge-side model from candidate models of cloud-side to predict PM10 and PM2.5 concentrations in a future single space. Masip-Bruin *et al.* [26] studied the collaborative management problem of edge-cloud continuity, and proposed a layered model to demonstrate the application of traffic control monitoring services in smart cities. Based on the effectively integration of edge computing and cloud computing, Wang *et al.* [27] proposed a tensor-enabled edge-cloud computing framework and service model to meet user requirements on cyber-physical-social services. Similarly, the ECC paradigm has gradually been focused in the manufacturing field. Afrin *et al.* [28] designed an ECC based robotic framework to deal with the limits of emergency management robots for smart factory in performing delay-sensitive tasks. Minimizing of makespan, energy consumption, and the total cost of resources were mainly discussed in their research. Qi and Tao [3] proposed an intelligent manufacturing hierarchy structure based on edge computing, fog computing, and cloud computing that could be used in digital twin workshops in response to the problems of manufacturing systems in cloud environments. This study opened up broad prospects for intelligent manufacturing. Obviously, cloud scheduling or manufacturing edge scheduling can no longer effectively support the ECC production paradigm. At present, the research on ECC mainly focuses on environment monitoring, smart cities, and manufacturing, etc. There are few research cases for the ECC production scheduling. Some of the cases mainly focus on production optimization issues, and ignore the scheduling architecture and mechanism of ECC production paradigm.

As above mentioned, the technology and market demand can contribute to the evolution of production paradigm. ECC has been widely researched in different industries, while ECC scheduling is still in an early stage. Therefore, this paper is committed to study the ECC production scheduling mechanism.

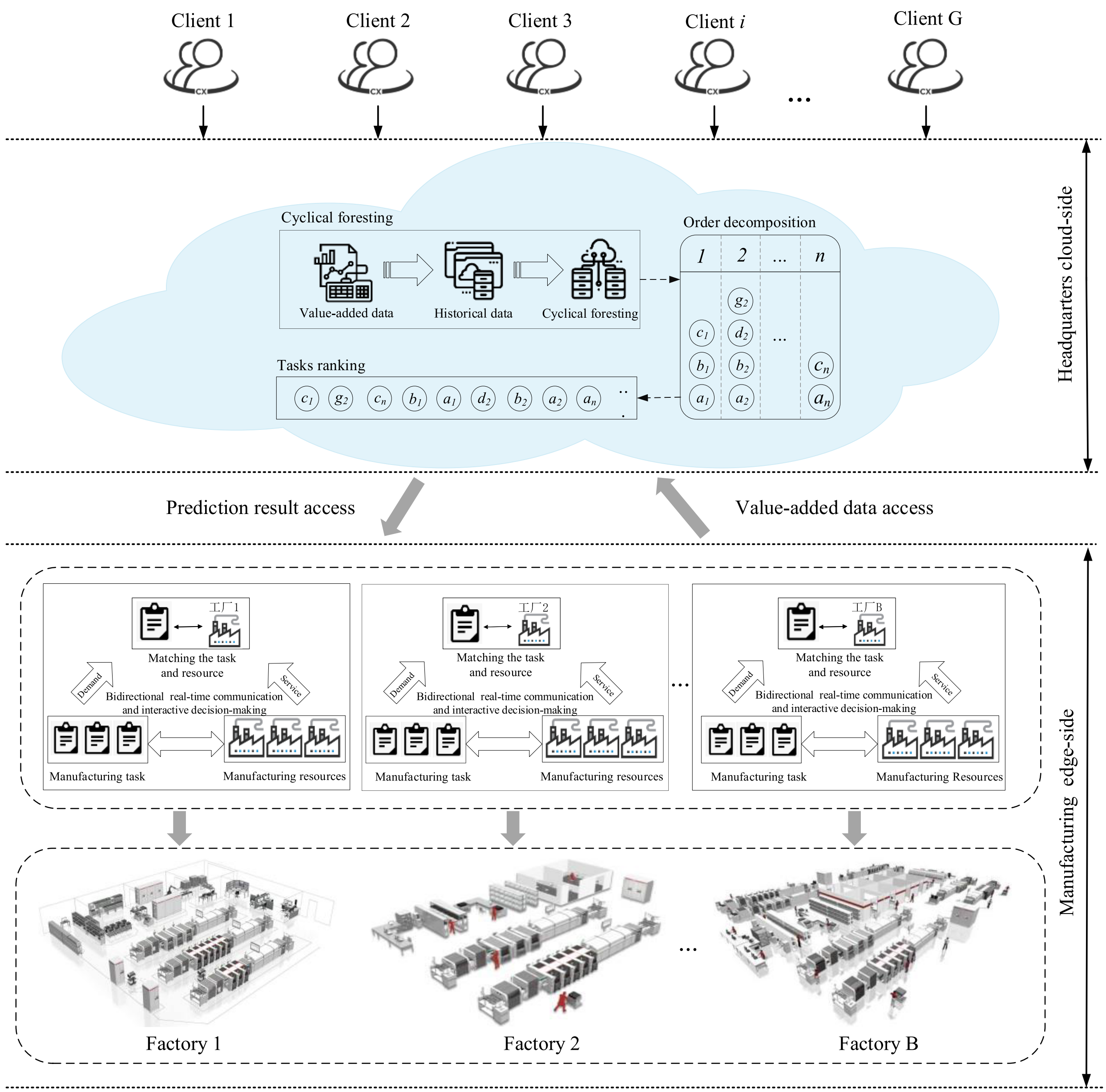



**FIGURE 1.** Scheduling framework of ECC production paradigm.

## III. FRAMEKWOR OF ECC PRODUCTION SCHEDULING

The purpose of this paper is to study the ECC scheduling mechanism and develop an ECC scheduling framework and models. Fig. 1 shows the overall ECC scheduling architecture. The proposed infrastructure integrates the real-time scheduling and periodic forecasting capability provided by manufacturing edge-side and cloud-side. Continuous scheduling optimization and optimal allocation of resources will be achieved by the edge-cloud collaboration. The proposed architecture and its mechanism will be introduced in three aspects.

(1) Components: dynamic matching of production tasks and resources at the manufacturing edge-side and the periodic prediction in the headquarters cloud-side.

(2) Manufacturing edge-side function: decompose the cloud-side orders of headquarters to match the production sub-task with multi factory or plant. Then, sub-tasks assigned to each plant will be scheduled in the plant to achieve optimization.

The manufacturing edge-side is consisted of manufacturing resources layer and tasks execution layer. Each task can communicate and interact each other autonomously through task agent and factory agent. Based on information sharing and a series of collaboration mechanisms, headquarters tasks will be assigned to each factory in accordance with constraint conditions, and then sub-task will be scheduled in each factory to ensure real-time distributed scheduling.

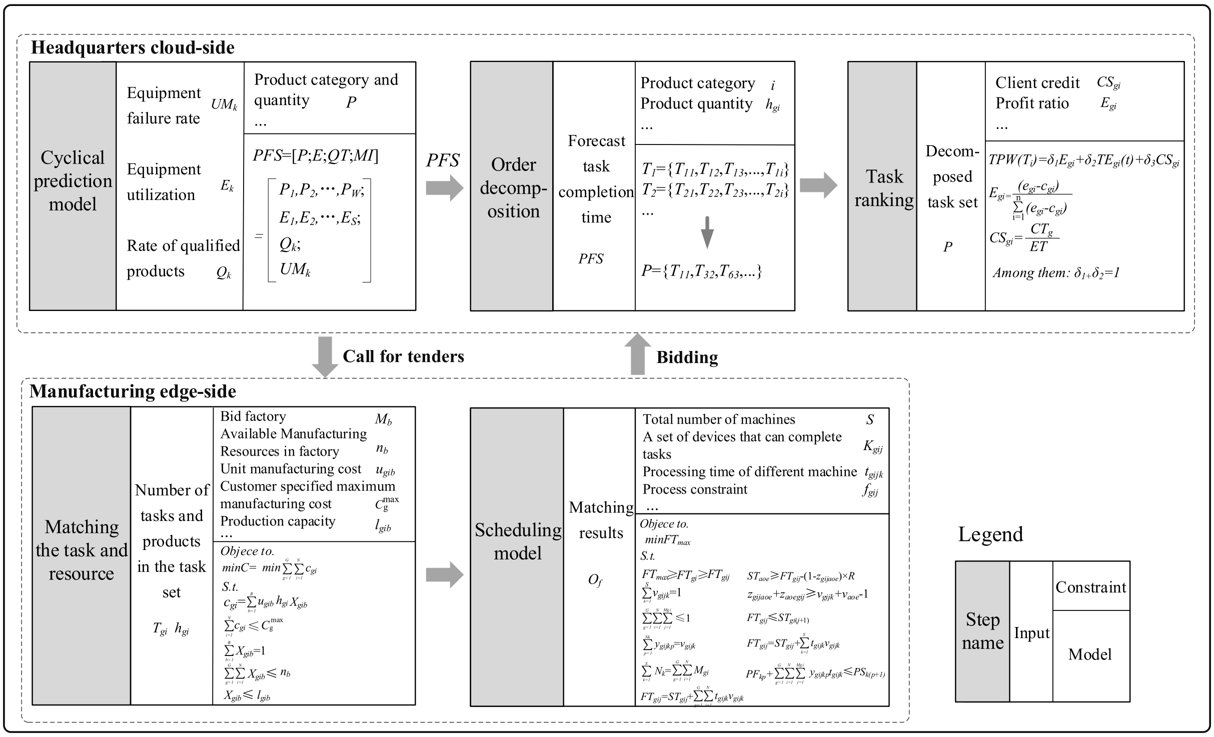


**FIGURE 2.** Conceptual diagram of the ECC production scheduling.

Headquarters cloud-side function: realize the periodic forecast of overall completion time for the headquarters production tasks. Upload and store the value-added data from the edge-side to support the periodic prediction. Decompose and sequence headquarters tasks based on periodic prediction results, and then send the processed tasks to the manufacturing edge-side factory to assistant the real-time distributed scheduling.

(3) Relationship: under the proposed ECC scheduling framework, each edge-side factory and machine can be seen as an independent agent. The cloud-side tasks agent and edge-side resources agent can interconnect and have autonomy. In this paper, we assume that each machine agent can store time-sensitive data, perform preliminary data processing, data screening, and other value-added processing in the manufacturing edge-side. The value-added data such as device utilization rate, product qualification rate, machine failure rate will be transferred to the headquarters cloud to support the periodic forecast. The forecast results can directly affect the headquarters order decomposition, and support the tasks sequencing based on a series of evaluation criteria. The decomposed sub-tasks will be sent to each factory and scheduled within factory. That is the closed-loop mechanism of ECC scheduling.

In summary, the proposed ECC scheduling framework can cope with the order fluctuations problem caused by the personalized customization requirements. It can help the enterprise with multi-factories to achieve accurate prediction and real-time scheduling optimization when facing with the order fluctuations.

## IV. MODELLING AND CASE VERIFICATION

The focus of the ECC scheduling is how to integrate the capabilities of edge-side real-time response and cloud-side accurate forecasting to effectively respond to the order fluctuations. That is, the accuracy of cloud-side periodic prediction and its supported real-time scheduling of edge-side is the key research issues. Therefore, a collaborative production scheduling model consisted of the cloud-side prediction model and edge-side scheduling optimization model is presented based on the neural network and generic algorithm. The proposed collaborative scheduling model is aim to achieve the overall completion time optimization for headquarters order. Based on the above mechanism, a conceptual diagram of the ECC scheduling is built (as shown in Fig. 2).

### *A. HEADQUARTERS CLOUD-SIDE PREDICTION MODEL*

The input data of collaborative scheduling model includes three categories of data. The first is value-added data (e.g. device utilization rate, product qualification rate, machine failure rate, etc.), which is produced and sent by edge-side factory. The second is some known data (e.g. production capacity), which can reflect factory characteristics. The last category is the customer orders. Based on these data, the periodic forecast for task completion time will be performed and the results will be used to support a more accurate manufacturing edge-side scheduling.

Suppose there is an enterprise with $B$ factories, each factory has $S$ equipments, and there are $W$ orders from different clients.

The prediction model is shown in (1).

$$PFS = [P; E; QT; MI] = \begin{bmatrix} P_1, P_2, \ldots, P_W; \\ E_1, E_2, \ldots, E_S; \\ Q_k; \\ UM_k \end{bmatrix} \quad k = 1, 2 \ldots, S \; b = 1, 2, \ldots, B \tag{1}$$

where $P$ represents the product type and product quantity included in the order. $E$ represents the utilization rate of the equipment, and $E_a$ represents the utilization rate of the equipment. $QT$ refers to the qualification rate of products in the most recently completed order, and $Q_s$ refers to the qualification rate of products processed by equipments in the recently completed order. $MI$ refers to the equipment operation information, and $UM_s$ represents failure rate of equipment $s$.

Data fitting is competed to support the prediction based on the feature data set of task completion time prediction. The periodic prediction formula is

$$OCT = f(PFS) \tag{2}$$

where $f$ represents the correspondence between $PFS$ and $OCT$.

After periodic forecast of headquarters cloud-side, the customer order is decomposed into multiple sub-tasks by considering the available resources of each edge-side factory. Therefore, the order of G customers are decomposed into multiple sub-tasks, and then produce a task set $P = \{T_{11}, T_{21}, \ldots, T_{gi}\}$. A task package can contain several tasks from a certain order or different orders. The task execution sequence and machining process of each task have timing and sequence constraints. The specific process is described as follows. Each factory makes full use of its advantages in resources to chooses capability-matching tasks to complete the headquarters cloud-side order. Weighted mean-based tasks sorting method is presented to determine the tasks sequencing based on the value of sorting weight $TPW(T_{gi})$. Higher ranking weight of tasks indicate that the customer's needs should be prioritized. From there, $\delta_1 + \delta_2 = 1$.

$$TPW(T_{gi}) = \delta_1 E_{gi} + \delta_2 CS_{\mathrm{gi}} \tag{3}$$

$$E_{gi} = \frac{e_{gi} - c_{gi}}{\sum_{g=1}^{G}\sum_{i=1}^{N}(e_{gi} - c_{gi})} \tag{4}$$

$$CS_{gi} = \frac{CT_{gi}}{ET} \tag{5}$$

where $E_{gi}$ represents the profit margin of $T_{gi}$. $e_{gi}$ denotes the expected revenues of $T_{gi}$. $c_{gi}$ represents the expected cost of $T_{gi}$. $CS_{gi}$ represents the credibility of customer $g$ from $T_{gi}$. $CT_{gi}$ indicates the cooperative time of customer $g$ from $T_{gi}$, and $ET$ indicates the establishment time of the group of companies. The longer the cooperation time, the better the credibility of customer.

### B. EDGE-SIDE SCHEDULING OPTIMIZATION MODEL

Edge-side scheduling optimization is an important part of the ECC scheduling model. The model is based on the following conditions and assumptions:

(1) Any sub-task can only be processed in one factory.

(2) Only one piece of equipment can be selected for any machining process.

(3) Once each of machining process of each work-piece started, it cannot be interrupted or preempted.

(4) Ignore nonproductive time and cost, such as changeover, transportation, etc.

To design the scheduling strategy of ECC production paradigms, the optimization goals and constraints must be determined. The parameters of the manufacturing edge-side model are shown in Table 1. For achieving the task-resource matching of edge-side, the cost minimization is chose as the goal to complete tasks allocation.

The objective function can be expressed as:

$$\min C = \min \sum_{g=1}^{G}\sum_{i=1}^{N} c_{gi} \tag{6}$$

During the matching process, the following constraints exist:

$$c_{gi} = \sum_{b=1}^{B} u_{gib} h_{gi} X_{gib}, \quad \forall g, i \tag{7}$$

$$\sum_{i=1}^{N} c_{gi} \leq C_g^{\max}, \quad \forall g \tag{8}$$

$$\sum_{b=1}^{B} X_{gib} = 1, \quad g \in [1, G], \; i = [1, N] \tag{9}$$

$$\sum_{g=1}^{G}\sum_{i=1}^{N} X_{gib} \leq n_b, \quad b \in [1, B] \tag{10}$$

$$X_{gib} \leq l_{gib}, \quad \forall g, i, b \tag{11}$$

The cost of $T_{gi}$ is defined in the (7). The manufacturing cost of $T_{gi}$ cannot exceed the maximum manufacturing cost specified by customers is shown in (8). A sub-task can only select one candidate factory is indicated in (9). (10) and (11) present the constraints of production capacity. Factory with sufficient resources to manufacture the product can carry out production.

After the completion task-resource matching, production scheduling is performed within each factory, which is relatively independent. Aiming at minimize the makespan, the objective function is expressed as:

$$\min FT_{\max} \tag{12}$$

Scheduling process has the following constraints:

$$FT_{\max} \geq FT_{gi} \geq FT_{gij}, \quad \forall j \tag{13}$$

$$\sum_{k=1}^{S} v_{gijk} = 1, \quad \forall g, i, j \tag{14}$$

**TABLE 1.** Parameters of the manufacturing edge-side model.

| Parameters | Explanation |
|---|---|
| $g$, $a$ | Index of customers |
| $i$, $o$ | Index of tasks |
| $j$, $e$ | Index of operations (1,…., $M_{gi}$) |
| k | Index of machines (1,…, $S$) |
| $p$ | Index of the processing location of the operation on the machine (1,…, $N_k$) |
| $S$ | Total number of machines |
| $T_{gi}$ | The $i$-th sub-task from client g, T={$T_{gi}$\| $g$=1,…., $G$; $i$=1,…., $N$} |
| $M_b$ | The $b$-th bid factory, $M$={$M_b$\| $b$=1,2,…, $B$} |
| $n_b$ | Available manufacturing resources in factory $b$ |
| $u_{gib}$ | Unit manufacturing cost of sub-task $T_{gi}$ in factory $b$ |
| $h_{gi}$ | The product quantity of sub-task $T_{gi}$ |
| $C_g^{max}$ | Maximum manufacturing cost specified by client $g$ |
| $l_{gib}$ | Describe the capable factory b is assigned to sub-task $T_{gi}$ |
| $c_{gi}$ | The manufacturing cost of sub-task |
| $X_{gib}$ | Whether sub-task $T_{gi}$ is produced in factory and refers to the variables 0 and 1 |
| $f_{gij}$ | Operation $j$ of sub-task $i$ from customer $g$ |
| $M_{gi}$ | Total operation number of sub-task $T_{gi}$ |
| $K_{gij}$ | Equipment set that can complete $f_{gij}$ |
| $t_{gijk}$ | Processing time of $f_{gij}$ if performed on machine k |
| $R$ | A large number |
| $FT_{gi}$ | Completion time of sub-task $T_{gi}$ |
| $FT_{gij}$ | Completion time of operation $f_{gij}$ |
| $ST_{gij}$ | Starting time of operation $f_{gij}$ |
| $PS_{kp}$ | Start time of the $p$-th processing step on machine $k$ |
| $PF_{kp}$ | Completion time of the $p$-th processing step on machine $k$ |
| $N_k$ | The number of assigned operations to machine $k$ |
| $v_{gijk}$ | Whether $f_{gij}$ is processed on machine $k$ and refers to the variables 0 and 1 |
| $y_{gijkp}$ | whether $f_{gij}$ is processed at the $p$-th position of machine $k$ and refers to the variables 0 and 1 |
| $z_{gijaoe}$ | Whether $f_{gij}$ is processed before $f_{aoe}$ and refers to the variables 0 and 1 |

$$\sum_{g=1}^{G}\sum_{i=1}^{N}\sum_{j=1}^{M_{gi}} y_{gijkp} \leq 1, \quad \forall k, p \tag{15}$$

$$\sum_{p=1}^{N_k} y_{gijkp} = v_{gijk}, \quad \forall g, i, j \tag{16}$$

$$\sum_{k=1}^{S} N_k = \sum_{g=1}^{G}\sum_{i=1}^{N} M_{gi} \tag{17}$$

$$FT_{gij} = ST_{gij} + \sum_{k=1}^{S} t_{gijk} v_{gijk}, \quad \forall g, i, j \tag{18}$$

$$ST_{aoe} \geq FT_{gij} - \left(1 - z_{gijaoe}\right) \times R, \quad \forall i, o, g, a, e, j,\ \forall k \in K_{gij} \cap K_{aoe} \tag{19}$$

$$z_{gijaoe} + z_{aoegij} \geq v_{gijk} + v_{aoe} - 1, \quad \forall g, i, j, a, e, j, k \tag{20}$$

$$FT_{gij} \leq ST_{gi(j+1)}, \quad \forall g, i,\ \forall j = 1, \ldots, M_{gi} - 1 \tag{21}$$

$$FT_{gij} = ST_{gij} + \sum_{k=1}^{S} t_{gijk} v_{gijk}, \quad \forall g, i, j \tag{22}$$

$$PF_{kp} + \sum_{g=1}^{G}\sum_{i=1}^{N}\sum_{j=1}^{M_{gi}} y_{gijkp} t_{gijk} \leq PS_{k(p+1)}, \quad \forall k, p = 1, \ldots, N_k - 1 \tag{23}$$

The definition of makespan is shown in (13). (14) and (15) illustrate that each operation of each sub-task can only be assigned to one machine, and each machine can process only one operation of the sub-task at the same time. Each operation in each sub-task can only be assigned to one location of machine that is indicated in (16). (17) shows that the total number of processes be processed for all equipments are equal to the total number of sub-tasks. The operation cannot be interrupted before completion, which is guaranteed by (18). The processing sequence for two non-continuous operations which are processed on the same machine are defined in (19) and (20). The processing sequence of two consecutive processes are clarified in (21) and (22). (23) points out that one equipment can only complete one process at a time.

### C. CASE VALIDATION

The headquarters of a mechanical products manufacturing enterprise locates in the center of Shanghai, China. The enterprise establishes some edge-side factories at the border of the city. The edge-side factories are developed, and the logistics distance between headquarters and each factory are less than 2 hours, so the impact of logistics on task allocation is not considered. On the basis of ignoring logistics costs, orders can be assigned to any factories. Only the data collaboration between the edge-side and cloud-side models should be verified, since the cloud and edge test environments are quite mature under current conditions. The algorithm of this case study is implemented with Matlab 2018b. For task matching and factory manufacturing, the genetic algorithms is employed. The algorithm parameters are set as follows: the population size is 100, the iteration number is 100, the crossover rate is 0.7, and the mutational rate is 0.1. For completion time prediction from cloud, set its hidden layers as 4.

Existing orders from 10 customers ($g_1$-$g_{10}$) are issued to the headquarters. The order contains 10 different types of mechanical parts ($A$-$J$), each of which has different process routes. Then production tasks in each order are allocated to 5 factories for production, and each plant has 10 machines to assist in manufacturing (where parts $A$-$J$ are represented

**TABLE 2.** Partial decomposition results of cloud-side order tasks.

| Sub-task | Customers | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $g_1$ | $g_2$ | $g_3$ | $g_4$ | $g_5$ | $g_6$ | $g_7$ | $g_8$ | $g_9$ | $g_{10}$ |
| $i_1$ | 652 | 620 | - | 549 | 751 | - | 702 | 557 | 428 | 376 |
| $i_2$ | 679 | - | 352 | 515 | 736 | 850 | - | 603 | 560 | 423 |
| $i_3$ | 642 | 750 | 476 | 519 | 425 | 746 | 668 | - | 603 | 410 |
| | | | | | ... | | | | | |
| $i_9$ | 582 | 518 | 545 | 568 | 673 | 813 | 695 | - | 714 | 429 |
| $i_{10}$ | 664 | 747 | 542 | 411 | 715 | 375 | 691 | 624 | 644 | - |

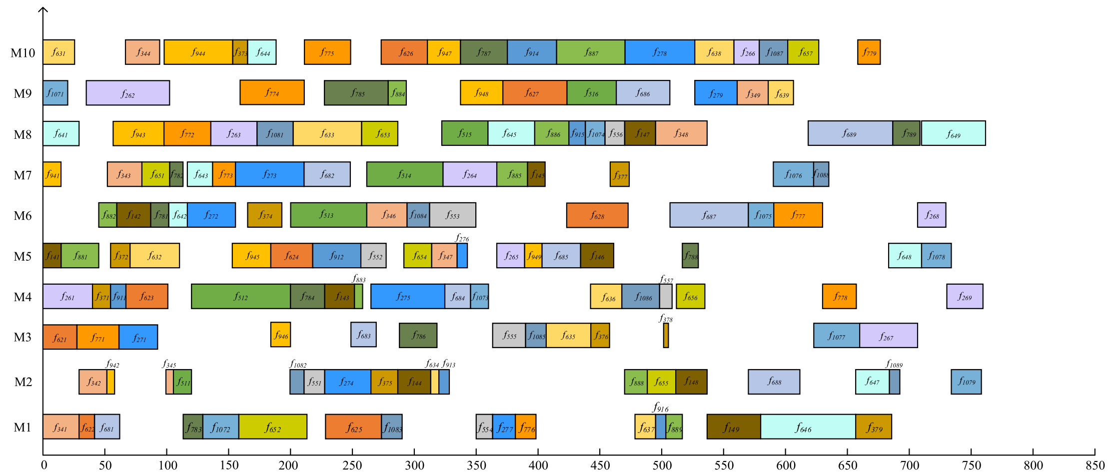



**FIGURE 3.** Gantt chart for process scheduling.

by $i_1$-$i_{10}$). The customer releases information of orders in the cloud and decomposes it according to the sub-task type. Partial decomposition results are shown in Table 2.

According to (3), the sort weights $TPW$ ($T_{gi}$) of the task has a certain impact on the internal scheduling of the manufacturing edge. We assume that $\delta_1 = 0.6$ and $\delta_2 = 0.4$, and the results in descending order: $T_{34}$, $T_{64}$, $T_{68}$, $T_{14}$, $T_{88}$, $T_{78}$, $T_{62}$, $T_{77}$, $T_{27}$, $T_{26}$, $T_{108}$, $T_{63}$, $T_{94}$, $T_{65}$, $T_{107}$, $T_{37}$, $T_{91}$, $T_{55}$, and $T_{51}$.

In this study, the following information are known: 1) the unit cost of each sub-task from different customers in different factories. 2) the highest manufacturing cost that each customer can accept. 3) the restriction of production capacity for the five factories. 4) the production resources of each factory. Taking the above data as input, assigning each sub-task to five factories, then the minimum total cost is 51234682.37 Chinese dollar. The allocation results are shown in Table 3.

From the allocation results, there are 19 sub-tasks for processing in Factory 1, which are $T_{14}$, $T_{26}$, $T_{27}$, $T_{34}$, $T_{37}$, $T_{51}$, $T_{55}$, $T_{62}$, $T_{63}$, $T_{64}$, $T_{65}$, $T_{68}$, $T_{77}$, $T_{78}$, $T_{88}$, $T_{91}$, $T_{94}$, $T_{107}$ and$T_{108}$.

Factory 2, Factory 3, Factory 4, and Factory 5 have 12, 17, 22, and 17 processing tasks, respectively, and the procedures of each sub-task are different. In Factory 1, for example, production production scheduling is performed inside the factory. The information of available machine of Factory 1 is shown in Table 4 and the processing time of each process is shown in Table 5.

**TABLE 3.** The allocation results among sub-task and factory.

| Sub-task | Customers | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $g_1$ | $g_2$ | $g_3$ | $g_4$ | $g_5$ | $g_6$ | $g_7$ | $g_8$ | $g_9$ | $g_{10}$ |
| $i_1$ | 3 | 5 | - | 2 | 1 | - | 4 | 3 | 1 | 2 |
| $i_2$ | 4 | - | 5 | 3 | 4 | 1 | - | 4 | 2 | 5 |
| $i_3$ | 3 | 3 | 2 | 3 | 5 | 1 | 4 | - | 4 | 5 |
| $i_4$ | 1 | 5 | 1 | 4 | 4 | 1 | 3 | 2 | 1 | 3 |
| $i_5$ | - | 4 | - | 4 | 1 | 1 | 2 | 5 | 4 | 5 |
| $i_6$ | 3 | 1 | 3 | - | 3 | 4 | - | 3 | 4 | 5 |
| $i_7$ | 4 | 1 | 1 | 2 | 4 | 5 | 1 | 3 | 4 | 1 |
| $i_8$ | 2 | 3 | 4 | 4 | - | 1 | 1 | 1 | - | 1 |
| $i_9$ | 5 | 4 | 4 | 5 | 2 | 5 | 3 | - | 3 | 5 |
| $i_{10}$ | 2 | 4 | 3 | 5 | 2 | 2 | 4 | 5 | 5 | - |

According to the proposed scheduling strategy, the scheduling results are shown in Fig. 3. The completion times of the 19 sub-tasks are 406, 467.52, 479.98, 501.34, 518.18, 526.64,

**TABLE 4.** The information of available machine in Factory 1.

| Sub-task | Operations | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $j_1$ | $j_2$ | $j_3$ | $j_4$ | $j_5$ | $j_6$ | $j_7$ | $j_8$ | $j_9$ |
| $T_{14}$ | 5 | 6 | 4 | [2,9] | [3,7] | 5 | [8,10] | 2 | 1 |
| $T_{26}$ | 4 | [2,9] | 8 | [6,7] | 5 | [1,10] | 3 | 6 | 4 |
| $T_{27}$ | 3 | [6,8] | 7 | [2,1] | [4,10] | 5 | 1 | 10 | 9 |
| $T_{34}$ | 1 | 2 | [4,7] | 10 | [2,5] | [3,6] | 5 | 8 | 9 |
| $T_{37}$ | [4,5] | 5 | [9,10] | 6 | 2 | [3,8] | 7 | 3 | 1 |
| $T_{51}$ | [2,6] | 4 | [6,9] | 7 | 8 | [3,9] | - | - | - |
| $T_{55}$ | 2 | [5,7] | 6 | [1,10] | 3 | 8 | 4 | - | - |
| $T_{62}$ | 3 | [2,1] | [4,10] | 5 | 1 | 10 | 9 | 6 | - |
| $T_{63}$ | [6,10] | 5 | 8 | 2 | 3 | [4,6] | 1 | 10 | 9 |
| $T_{64}$ | 8 | 6 | [3,7] | 10 | 8 | 1 | [2,9] | 5 | 8 |
| $T_{65}$ | 7 | 1 | 8 | 5 | 2 | [4,7] | 10 | - | - |
| $T_{68}$ | 1 | 7 | 3 | [4,5] | 5 | [9,10] | 6 | 2 | [3,8] |
| $T_{77}$ | [3,5] | 8 | [6,7] | 9 | [1,10] | 1 | 6 | 4 | 10 |
| $T_{78}$ | [6,8] | 7 | [2,1] | [4,5] | 9 | 3 | 10 | 5 | 8 |
| $T_{88}$ | 5 | 6 | 4 | [3,9] | [5,7] | 8 | [8,10] | 2 | 1 |
| $T_{91}$ | 4 | 5 | [2,7] | 10 | 8 | 1 | - | - | - |
| $T_{94}$ | [4,7] | 2 | 8 | 10 | [1,5] | [3,6] | 10 | 9 | 5 |
| $T_{107}$ | [8,9] | 1 | 4 | 8 | 6 | 7 | 3 | 5 | 2 |
| $T_{108}$ | 8 | [2,5] | 1 | 6 | 3 | 4 | 10 | 7 | 2 |

Notes: The available machines in the table is expressed in the format of "[ , ]", which means that there are two optional machines for this operation. For example, the optional machines [8,10] of the 7th operation $j_7$ for sub-task $T_{14}$ indicates that it the 7th operation $j_7$ of $T_{14}$ can be processed on either the machine 8 or the machine 10.

**TABLE 5.** The processing time of each operation in Factory 1.

| Sub-task | Operations | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $j_1$ | $j_2$ | $j_3$ | $j_4$ | $j_5$ | $j_6$ | $j_7$ | $j_8$ | $j_9$ |
| $T_{14}$ | 3 | 5 | 3 | [5,4] | [3,3] | 5 | [6,7] | 3 | 8 |
| $T_{26}$ | 6 | [8,9] | 5 | [2,5] | 3 | [3,3] | 6 | 3 | 5 |
| $T_{27}$ | 4 | [5,7] | 7 | [5,5] | [9,11] | 1 | 3 | 8 | 6 |
| $T_{34}$ | 7 | 3 | [4,6] | 3 | [1,2] | [3,6] | 4 | 7 | 5 |
| $T_{37}$ | [3,4] | 4 | [3,3] | 8 | 5 | [4,5] | 5 | 2 | 7 |
| $T_{51}$ | [3,2] | 10 | [8,7] | 9 | 4 | [5,4] | - | - | - |
| $T_{55}$ | 5 | [6,5] | 8 | [3,3] | 6 | 5 | 5 | 1 | 7 |
| $T_{62}$ | 4 | [2,1] | [4,5] | 5 | 6 | 4 | 6 | 5 | - |
| $T_{63}$ | [3,5] | 6 | 8 | 1 | 5 | [4,7] | 3 | 5 | 2 |
| $T_{64}$ | 5 | 3 | [1,2] | 3 | 5 | 9 | [3,5] | 4 | 7 |
| $T_{65}$ | 3 | 8 | 5 | 4 | 6 | [5,6] | 5 | [5,5] | [4,6] |
| $T_{68}$ | 2 | 5 | 3 | [2,1] | 4 | [6,4] | 7 | 5 | [5,8] |
| $T_{77}$ | [4,6] | 5 | [2,2.5] | 7 | [3,5] | 2 | 5 | 4 | 2 |
| $T_{78}$ | [2,3] | 3 | [1,2] | [5,5] | 8 | 5 | 7 | 3 | 4 |
| $T_{88}$ | 6 | 2 | 1 | [3,2] | [4,5] | 4 | [9,10] | 3 | 5 |
| $T_{91}$ | 3 | 7 | [2,3] | 7 | 5 | 1 | [3,2] | 5 | 6 |
| $T_{94}$ | [5,3] | 1 | 7 | 8 | [3,5] | [3,2] | 5 | 7 | 3 |
| $T_{107}$ | [6,5] | 5 | 3 | 2 | 4 | 6 | 8 | 5 | 3 |
| $T_{108}$ | 7 | [3,4] | 3 | 6 | 5 | 6 | 5 | 4 | 3 |

Notes: The unit of processing time for the above tasks is: hours. Due to the large number of tasks for each sub-task with the long completion time, we reduced the above processing time by 100 times during the model calculation.

568.19, 587.5, 594.39, 609.31, 634.12, 679.13, 684.2, 685.14, 693.13, 712.46, 753.9, 766.65, and 768.97.

The edge-side real-time production data of Factory 1 is processed, and the value-added data is sent to headquarters cloud-side to predict the task completion time periodically by (1). On the basis of $T_{94}$, $T_{51}$ and $T_{62}$, $T_{91}$ is predicted on the 20th day, and the predicted completion time is 483.26. Based on the previously completed production tasks, the completion

**TABLE 6. Comparison of actual completion time and predicted completion time.**

| Sub-task ID | Actual completion time | Predicted completion time | Relative error |
|---|---|---|---|
| $T_{91}$ | 501.34 | 477.26 | 4.80% |
| $T_{27}$ | 568.19 | 548.36 | 3.49% |
| $T_{65}$ | 634.12 | 615.29 | 2.96% |
| $T_{108}$ | 693.13 | 674.36 | 2.71% |
| $T_{64}$ | 768.97 | 749.64 | 2.51% |

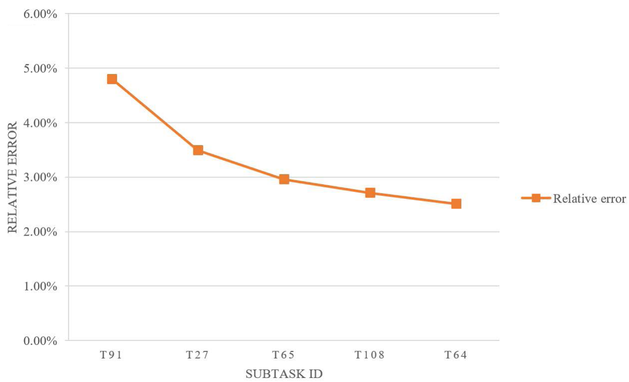



**FIGURE 4. Tend graph of relative errors.**

times of $T_{27}$, $T_{65}$, $T_{108}$, and $T_{64}$ are predicted on the 22th, 26th, 29th and 32th days with the results of 548.36, 615.29, 674.36 and 749.64. Then comparison results of the actual completion time and the predicted completion time about these three tasks are shown in Table 6, and a relative error trend is shown in Fig. 4. It is obvious that the relative error between the actual value and predicted value is smaller with the increasing of data volume, and the prediction results become more accurate when data mining is performed in the cloud. At the same time, we randomly generate 1000 samples, and the calculation results show that the edge-side can still satisfy the needs of computing when facing with a large amount of data. Therefore, it is verified that the proposed ECC scheduling method can well realize data coordination for the headquarters-factory paradigm, and achieve accurate scheduling at the edge-side based on the cloud-side periodic prediction results. With the increasing of data scale, the manufacturing edge-side can still support the current equivalent level of calculation, so this mechanism is also effective in the case of a sudden increase in data.

## V. CONCLUSION AND FUTURE WORK

In conclusion, cloud-side scheduling or manufacturing edge-side scheduling can no longer effectively support the headquarters-factory business mode and response to the order dynamic fluctuations caused by personalized customization demands. Driving by the ECC technologies and its enabled production paradigm, an ECC scheduling method and its framework, mechanism, and model are discussed and developed in this paper. A case study is conducted to demonstrate the proposed method. The contributions of this paper are summarized as follows:

1) The concept of ECC scheduling is proposed, which aims at coping with order dynamic fluctuations.
2) To address the ECC scheduling method, the framework, mechanism, and models are presented, which support the real-time distributed scheduling and periodic prediction to achieve optimal allocation of cloud-side headquarters orders and edge-side factory resources.

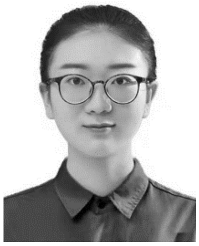
At present, the ECC production scheduling is still in an early stage. This paper proposes an ECC scheduling method and verifies its rationality. However, there are still some unconsidered factors, such as uncertainties of real-time workflows and others. Therefore, some further research will be carried out in the future. The development of edge technology, ECC scheduling mechanisms and methods will promote the ECC scheduling research. In addition, some diversified verification methods and technologies (e.g. virtual simulation and the digital twin, etc.) will be used to improve the accuracy of real-time scheduling decisions.

## REFERENCES

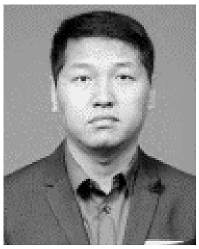
[1] B. H. Li, L. Zhang, L. Ren, X. D. Chai, and F. Tao, ''Further discussion on cloud manufacturing,'' *CIMS*, vol. 17, no. 03, pp. 449–457, 2011.
[2] B.-H. Li, L. Zhang, S.-L. Wang, F. Tao, J. W. Cao, X. D. Jiang, X. Song, and X. D. Chai, ''Cloud manufacturing: A new service-oriented networked manufacturing model,'' *Comput. Integr. Manuf. Syst.*, vol. 16, no. 1, pp. 1–7, 2010.
[3] Q. Qi and F. Tao, ''A smart manufacturing service system based on edge computing, fog computing, and cloud computing,'' *IEEE Access*, vol. 7, pp. 86769–86777, 2019, doi: 10.1109/access.2019.2923610.
[4] L. Zhou, L. Zhang, B. R. Sarker, Y. Laili, and L. Ren, ''An event-triggered dynamic scheduling method for randomly arriving tasks in cloud manufacturing,'' *Int. J. Comput. Integr. Manuf.*, vol. 31, no. 3, pp. 318–333, Mar. 2018, doi: 10.1080/0951192x.2017.1413252.
[5] Y. Yadekar, E. Shehab, and J. Mehnen, ''Uncertainties in cloud manufacturing,'' in *Proc. CE*, Beijing, China, 2014, pp. 297–306.
[6] J. Lartigau, L. Nie, X. Xu, D. Zhan, and T. Mou, ''Scheduling methodology for production services in cloud manufacturing,'' in *Proc. Int. Joint Conf. Service Sci.*, May 2012, pp. 34–39.
[7] D. Georgakopoulos, P. P. Jayaraman, M. Fazia, M. Villari, and R. Ranjan, ''Internet of Things and edge cloud computing roadmap for manufacturing,'' *IEEE Cloud Comput.*, vol. 3, no. 4, pp. 66–73, Jul. 2016, doi: 10.1109/mcc.2016.91.
[8] L. Wang, C. Guo, S. Guo, B. Du, X. Li, and R. Wu, ''Rescheduling strategy of cloud service based on shuffled frog leading algorithm and Nash equilibrium,'' *Int. J. Adv. Manuf. Technol.*, vol. 94, nos. 9–12, pp. 3519–3535, Feb. 2018, doi: 10.1007/ s00170-017-1055-x.

[9] Y. Cheng, F. Tao, L. Zhang, and D. Zhao, "Dynamic supply-demand matching for manufacturing resource services in service-oriented manufacturing systems: A hyper network-based solution framework," in *Proc. ICME*, Charlotte, NC, USA, 2015, p. 1.
[10] L. Zhou and L. Zhang, "A dynamic task scheduling method based on simulation in cloud manufacturing," in *Proc. SCS Autumn Simulation Multi-Conf.*, Beijing, China, 2016, pp. 20–24.
[11] Y. Zhang, J. Wang, S. Liu, and C. Qian, "Game theory based real-time shop floor scheduling strategy and method for cloud manufacturing," *Int. J. Intell. Syst.*, vol. 32, no. 4, pp. 437–463, Apr. 2017, doi: 10.1002/int.21868.
[12] V. Vinod and R. Sridharan, "Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system," *Int. J. Prod. Econ.*, vol. 129, no. 1, pp. 127–146, Jan. 2011.
[13] T. Chen, "Embedding a back propagation network into fuzzy C-means for estimating job cycle time: Wafer fabrication as an example," *J. Ambient Intell. Hum. Comput.*, vol. 7, no. 6, pp. 789–800, Dec. 2016, doi: 10.1007/s12652-015-0336-1.
[14] J. Chang, X. Kong, and L. Yin, "A novel approach for product makespan prediction in production life cycle," *Int. J. Adv. Manuf. Technol.*, vol. 80, nos. 5–8, pp. 1433–1448, Sep. 2015, doi: 10.1007/s00170-015-7093-3.
[15] C. Wang and P. Jiang, "Deep neural networks based order completion time prediction by using real-time job shop RFID data," *J. Intell. Manuf.*, vol. 30, no. 3, pp. 1303–1318, Mar. 2019, doi: 10.1007/s10845-017-1325-3.
[16] F. Chan, S. Chung, and P. Chan, "An adaptive genetic algorithm with dominated genes for distributed scheduling problems," *Expert Syst. Appl.*, vol. 29, no. 2, pp. 364–371, Aug. 2005, doi: 10.1016/j.eswa.2005.04.009.
[17] J. Gao and R. Chen, "A hybrid genetic algorithm for the distributed permutation flowshop scheduling problem," *Int. J. Crit. Infrastruct.*, vol. 4, no. 4, p. 497, 2011, doi: 10.2991/ijcis.2011.4.4.9.
[18] H.-C. Chang and T.-K. Liu, "Optimisation of distributed manufacturing flexible job shop scheduling by using hybrid genetic algorithms," *J. Intell. Manuf.*, vol. 28, no. 8, pp. 1973–1986, Dec. 2017, doi: 10.1007/s10845-015-1084-y.
[19] P.-H. Lu, M.-C. Wu, H. Tan, Y.-H. Peng, and C.-F. Chen, "A genetic algorithm embedded with a concise chromosome representation for distributed and flexible job-shop scheduling problems," *J. Intell. Manuf.*, vol. 29, no. 1, pp. 19–34, Jan. 2018, doi: 10.1007/s10845-015-1083-z.
[20] J. Dong and C. Ye, "Research on collaborative optimization of green manufacturing in semiconductor wafer distributed heterogeneous factory," *Appl. Sci.*, vol. 9, no. 14, p. 2879, Jul. 2019, doi: 10.3390/app9142879.
[21] S. H. Chung, F. T. S. Chan, and W. H. Ip, "Minimization of order tardiness through collaboration strategy in multifactory production system," *IEEE Syst. J.*, vol. 5, no. 1, pp. 40–49, Mar. 2011, doi: 10.1109/jsyst.2010.2100194.
[22] G. Varela, M. L. R. Varela, and G. D. Putnik, "Technologies integration for distributed manufacturing scheduling in a virtual enterprise," in *Proc. CCIS*, Ofir, Portugal, 2012, pp. 337–347.
[23] P. X. Wang, R Gao and Z. Fan, "Cloud computing for cloud manufacturing: Benefits and limitations," *J. Manuf. Sci. Eng.*, vol. 137, no. 4, 2015, doi: 10.1115/1.4030209.
[24] I. Mezgár and U. Rauschecker, "The challenge of networked enterprises for cloud computing interoperability," *Comput. Ind.*, vol. 65, no. 4, pp. 657–674, May 2014, doi: 10.1016/j.compind.2014.01.017.
[25] J. Moon, S. Kum, and S. Lee, "A heterogeneous IoT data analysis framework with collaboration of edge-cloud computing: Focusing on indoor PM10 and PM2.5 status prediction," *Sensors*, vol. 19, no. 14, p. 3038, Jul. 2019, doi: 10.3390/s19143038.
[26] X. Masip-Bruin, E. Marin-Tordera, A. Jukan, and G.-J. Ren, "Managing resources continuity from the edge to the cloud: Architecture and performance," *Future Gener. Comput. Syst.*, vol. 79, pp. 777–785, Feb. 2018, doi: 10.1016/j.future.2017.09.036.
[27] X. Wang, L. T. Yang, X. Xie, J. Jin, and M. J. Deen, "A cloud-edge computing framework for cyber-physical-social services," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 80–85, Nov. 2017, doi: 10.1109/mcom.2017.1700360.
[28] M. Afrin, J. Jin, A. Rahman, Y.-C. Tian, and A. Kulkarni, "Multi-objective resource allocation for Edge Cloud based robotic workflow in smart factory," *Future Gener. Comput. Syst.*, vol. 97, pp. 119–130, Aug. 2019, doi: 10.1016/j.future.2019.02.062.

**JING MA** was born in Guyuan, Ningxia, China, in 1987. He received the B.S. and Ph.D. degrees in mechanical engineering from the Hefei University of Technology, Hefei, China, in 2010 and 2015, respectively. From 2016 to 2018, he was a Postdoctoral Fellow with Tsinghua University. Since 2018, he has been an Assistant Professor with the Mechanical Engineering Department, Beijing Jiaotong University. His research interests include intelligent manufacturing systems planning, modeling and optimization, and remaining useful life prediction and maintenance optimization.



**HUA ZHOU** was born in Yinchuan, Ningxia, China. She received the B.S. degree in industrial engineering from Northeast Forestry University, Harbin, China, in 2018. Since 2018, she has been a Postgraduate Researcher with Beijing Jiaotong University. Her research interest includes operation optimization theory of intelligent manufacturing systems.



**CHANGCHUN LIU** was born in Shandong, China, in 1990. He received the Ph.D. degree from the Department of Industrial Engineering, Tsinghua University, China, in 2019. Since 2019, he has been a Postdoctoral Fellow with the National University of Singapore. His research interests include operations research on container terminal, production scheduling, and logistic. His work has appeared in EJOR, CAIE, IJPR, TRE, and COR.

**MINGCHENG E** was born in Dalian, Liaoning, China. He received the Ph.D. degree in mechanical engineering from Tianjin University, Tianjin, China, in 1997. From 1997 to 2000, he was an Assistant Professor with Beijing Jiaotong University, where he has been an Associate Professor with the Mechanical Engineering Department, since 2000. His research interests manufacturing system modeling and optimization include manufacturing system modeling and optimization, remaining useful life estimation, and optimal maintenance policy.

**ZENGQIANG JIANG** was born in Yixing, Jiangsu, China. He received the B.S. and Ph.D. degrees in mechanical engineering from the Hefei University of Technology, Hefei, Anhui, China, in 2002 and 2006, respectively. From 2006 to 2008, he was an Assistant Professor with the Hefei University of Technology, and became an Associate Professor, in 2008. Since 2011, he has been an Associate Professor with the Mechanical Engineering Department, Beijing Jiaotong University, where he has also been a Professor, since 2020. His research interests include intelligent manufacturing systems, life prediction, and maintenance optimization.

**QIANG WANG** was born in Hefei, Anhui, China, in1987. He received the B.S. and Ph.D. degrees in mechanical engineering from the Hefei University of Technology, Hefei, China, in 2009 and 2016, respectively. Since 2017, he has been a Postdoctoral Fellow with Tsinghua University. His research interests include intelligent manufacturing systems, remaining useful life prediction, and maintenance optimization.