A Proactive Manufacturing Resources Assignment Method Based on Production Performance Prediction for the Smart Factory

Wenbo Wang, Yingfeng Zhang, *Member, IEEE*, Jinan Gu, and Jin Wang

***Abstract*—With the wide application of advanced industrial Internet of Things (IIoT) and cyber physical system (CPS) technologies, the manufacturing resources assignment method is transformed from manual and passive mode to intelligent and active mode. However, due to the lack of real-time analysis and accurate prediction of production performance, the production adjustment demands are often released after production exceptions happen, and production decisions are often made based on historical production information, which may lead to the problem of production interruption or performance reduction. To address this issue, a proactive manufacturing resources assignment (PMRA) method based on production performance prediction for the smart factory is proposed. Firstly, the advanced IIoT and CPS technologies are applied to create a cloud-edge cooperation environment for smart factory, where the resources are made smart with distributed control capacity, and cloud center and edge resources can collaborate dynamically. Secondly, a real-time colored Petri net (RCPN) enabled key production performance indicators (KPPIs) analysis and prediction method are proposed to extract real-time production information and predict future production status accurately. Then, the PMRA method is presented to assign the resources before production exceptions happen. Finally, a case study from a typical manufacturer for computer numerical control (CNC) machine tools in North China is used to validate the proposed method and results show that the proposed PMRA method can largely reduce the total tardiness and the total energy consumption.**

Manuscript received June 14, 2020; revised March 3, 2021 and April 10, 2021; accepted April 11, 2021. This work was supported in part by Key Project of National Science Foundation of China under Grant U2001201, in part by the National Science Foundation of China under Grant 51875266, in part by the Scientific Research Program Funded by Shaanxi Provincial Education Department under Grant 20JK0920, and in part by the Key Research and Development Program of Shaanxi under Grant 2021GY-069. (Corresponding author: Yingfeng Zhang).

W. Wang is with the School of Mechanical Engineering, Jiangsu University, Zhenjiang 212013, China and also with the Key Laboratory of Industrial Engineering and Intelligent Manufacturing, Ministry of Industry and Information Technology, School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: wangwb@ujs.edu.cn).

Y. Zhang is with the Key Laboratory of Industrial Engineering and Intelligent Manufacturing, Ministry of Industry and Information Technology, School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: zhangyf@nwpu.edu.cn).

J. Gu is with the School of Mechanical Engineering, Jiangsu University, Zhenjiang 212013, China (e-mail: gjnan@ujs.edu.cn).

J. Wang is with the School of Modern Posts, Xi'an University of Posts & Telecommunications, Xi'an 710061, China (e-mail: jinwang@xupt.edu.cn).

***Index Terms*—Smart factory; Manufacturing resources assignment; Proactive decision making; Key production performance prediction; Self-adaptive optimization.**

I. INTRODUCTION

WITH the wide application of cutting-edge technologies, such as industrial Internet of Things (IIoT), cyber physical system (CPS) [1] and cloud computing, the traditional manufacturing factories are turned into smart ones. In a smart factory, the status of manufacturing resources can be captured timely, production decisions can be made optimally using smart algorithms, and production instructions can be transmitted in time and executed accurately [2]. Thus, many advanced production decision-making strategies are enabled, such as real-time scheduling [3] and production logistics synchronization [4]. Based on the application of these real-time production management strategies, the production system can be more fluent and efficient.

Manufacturing resources assignment (MRA) plays an important role in improving productivity and responsiveness. A well-operated production system can increase machine utilization and reduce costs so as to increase profit gain. Thus, in recent years, MRA has gained great attention [5], including deep reinforcement learning based automated guided vehicles real-time scheduling [3], cloud-based CPS for adaptive shop-floor scheduling and condition-based maintenance [6], etc.

Despite the progress, there are several unsolved issues in MRA due to the increasing process complexity, unpredictable exceptions, etc. These issues are summarized as follows:

1) In the MRA stage, tasks are assigned to machines according to historical resources status. The real-time statuses of the resources are seldom considered. However, the status of manufacturing resources changes with time. The tasks may not be optimally assigned among the machines, which often leads to adverse effects on normal production operation.
2) With the wide application of IIoT and CPS technologies, amounts of real-time and multi-source manufacturing data is collected. However, only key production performance indicators (KPPIs) are useful for making production decisions. There is less research on extracting and predicting KPPIs from raw manufacturing data.
3) Most of existing MRA methods use passive and afterwards strategies. The production rescheduling methods are often

triggered after production performance decreases or exceptions occur. Since the abnormal production status takes a period of time to be eliminated, production system will operate in an abnormal status for some time and the production capacity may be constrained.

To tackle these problems, this paper proposed a proactive manufacturing resources assignment (PMRA) method based on production performance prediction. In general, five aspects are important to assess the performance of manufacturing systems, i.e., cost, quality, time, inventory and maintenance. The KPPIs are used to give quantitative values for production performance. This paper mainly focuses on the time and inventory performance. Firstly, the advanced IIoT and CPS technologies are applied to create a smart factory, where the resources are made smart with distributed control capacity, and cloud center and edge resources can collaborate dynamically. Then, the real-time KPPIs are extracted and future KPPIs are predicted using a real-time colored Petri net (RCPN) model. Lastly, the non-dominant sorting genetic algorithm II (NSGA-II) is used to predictively allocate the manufacturing resources. It includes two sides: On the one hand, it predicts the production exceptions in advance and takes proactive measures to reduce their adverse impacts; On the other hand, it assigns the manufacturing resources based on their real-time status. At last, a case study from a typical manufacturer for computer numerical control (CNC) machine tools in North China is used to validate the feasibility and effectiveness of the proposed method. The designed method will provide a new paradigm for manufacturing enterprises to implement more efficient production management.

The rest of paper is organized as follows. Section II reviews the relevant literature under three categories. Section III presents the overall architecture of the proposed method and briefly introduces its key components. Section IV discusses the RCPN-enabled KPPIs analysis and prediction method in detail. Section V presents the self-adaptive PMRA method in detail. Section VI uses a case study to illustrate the implementation of the proposed method. Finally, conclusions and future works are summarized in Section VII.

II. Literature review

Three streams of the literature are relevant to this research. They are IIoT, CPS and their application in manufacturing, Petri Nets (PNs) and their application in manufacturing, and dynamic manufacturing resources assignment.

A. IIoT, CPS and their application in manufacturing

IIoT technologies, including RFID, WLAN, embedded object logic, etc., have been widely developed in manufacturing applications for its autonomy, flexibility, reconfigurability, and scalability [7], including intelligent perception and access of manufacturing resources [8], prognostics and health management [9], etc. The term CPS provides a theoretical framework for mapping the manufacturing-related things to the computing space, so that the modelling of manufacturing systems can be easily achieved, and the optimal production decisions can be simulated and evaluated without wasting physical assets. As to the implementation of CPS, Monostori et al. discussed the expectations, research and development challenges of CPSs in manufacturing [10]. Lee et al. discusses the CPSs for predictive production systems [11]. Álvares et al. developed a cyber-physical framework for monitoring and interaction control of a CNC lathe by combining CPS and IIoT technologies [12]. Leng et al. proposed a digital twin-driven manufacturing CPS for parallel controlling of smart workshop [13]. Cheng et al. proposed a digital twin enhanced industrial Internet reference framework for smart manufacturing [14].

B. Petri Nets and their application in manufacturing

PNs are known to be powerful for process modelling (both graphically and mathematically) and formal verification. They are widely used for modelling, analysis, and control of large-scale automated manufacturing systems [15] and flexible manufacturing systems [16]. Recently, some works discussed the application of PN technology in modelling the dynamic behavior of smart manufacturing system. Wu and Zhou proposed an intelligent token PN for modelling and controlling manufacturing systems, where smart tokens representing job instances carried real-time knowledge about system status just like smart cards in real-life shop-floor [17].

In the PNs, places are used to represent the status of resources, transitions are used to represent activities, and the state is described by tokens in places. For a practical system, the scale of PN models can be very large when considering the variability of products. The CPN model is proposed to reduce the model size with a high-level programing language (colored tokens) and hierarchical model structure. Lv et al. developed a RFID-based CPN model whose colored tokens are used to describe the evolution of dynamic production status in real-life [18]. Zhang et al. proposed a hierarchical timed CPN based real-time production performance analysis model [19].

C. Dynamic manufacturing resources assignment

In real-world manufacturing systems, the working environment always changes dynamically due to the unpredictable uncertain events, such as machines break down suddenly, new orders arrive randomly. Therefore, more and more approaches have been proposed in existing literature. Li et al. proposed an optimization method for task assignment for industrial manufacturing organizations [20]. Ramandi et al. presented a stochastic economic predictive control model for real-time scheduling of balance responsible parties [21]. Wang et al. proposed an adaptive job shop scheduling strategy based on weighted Q-learning algorithm [22]. Leng et al. proposed a contextual self-organizing cyber-physical-social system for mass individualization [23]. Zhang et al. proposed an integrated framework for active discovery and optimal allocation of smart manufacturing services [24]. Wang et al. discussed a reinforcement learning method for real-time pricing and scheduling control in E-V charging stations [25]. Nikolakis et al. proposed a containerized approach for the dynamic planning and control of cyber-physical production system [26].

D. Knowledge gaps and motivations

From this review, although significant progress has been

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2021.3073404, IEEE Transactions on Industrial Informatics

made in the three dimensions mentioned above, there are still some gaps that need to be fulfilled.

1) In the terms of IIoT, CPS and their application in manufacturing, most research focused on the high-level frameworks for advanced manufacturing system management. Limited effort was made on the specific algorithms for extraction, prediction and optimization of production performance.
2) In the terms of PNs and their application in manufacturing, most studies focused on the modelling and analysis of production systems. Few works considered the synchronous operation between CPNs and real-life production. However, the real-life production status changes dynamically, the desynchrony may lead to inaccurate analysis of production systems, which may further result in improper decisions.
3) In the terms of dynamic MRA, most works made rescheduling or maintenance decisions after exceptions happened, and the historic outdated information was often used as decision basis. Due to the existence of uncertain factors and time consumption for exception response, it is hard to achieve efficient and optimal production decisions.

III. THE PROPOSED ARCHITECTURE

This research focuses mainly on discrete manufacturing systems. The objective of the proposed method is to change traditional passive MRA mode with the predictive analytics, IIoT, and CPS technologies. The overall architecture of the proposed method is shown in Fig. 1. Mainly it consists of three modules, i.e., configuration of smart factory, RCPN-enabled KPPIs analysis and prediction, and self-adaptive PMRA.

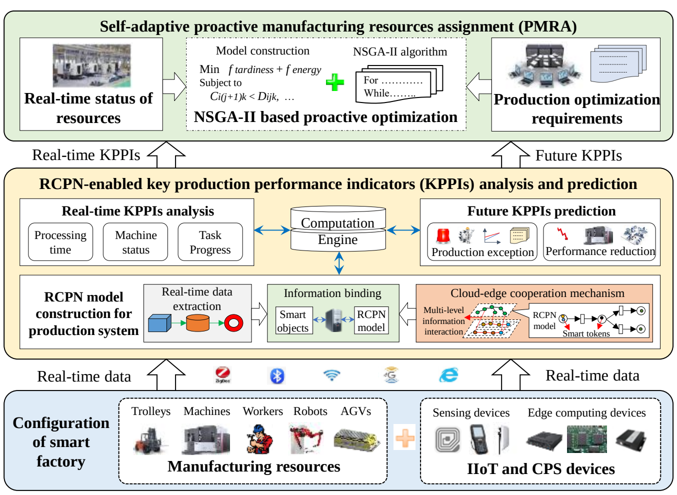


Fig. 1. Overall architecture of PMRA method

A. *Configuration of smart factory*

This module is responsible for constructing a smart factory by adopting IIoT and CPS devices. In a factory, manufacturing resources can be divided into two main sets: production set (including machines, tools, operators, etc.) and logistics set (including trolleys, workers, pallets, etc.). The manufacturing resources are made 'smart' with self-sensing, self-analysis and self-optimization capacity. Firstly, the advanced information communication technologies (e.g., RFID, WLAN) are applied so that the real-time status of resources can be sensed and information can be interacted among different resources and upper-level application systems. Secondly, the embedded edge-computing devices (including hardware and software) are applied to process the data at edge side, transmit high-quality information, make and execute production decisions self-adaptively. Therefore, a cloud-edge cooperation environment can be created, where the distributed control methods can be enabled to reduce the huge complexity issues of centralized techniques, and the coordination between cloud center and edge side are also set to satisfy the production requirements.

B. *RCPN-enabled KPPIs analysis and prediction*

This module is responsible for analyzing the real-time KPPIs and predicting the future KPPIs using the RCPN model. The proposed RCPN model can add the real-time information to traditional CPN model and evolve with real-life production system dynamically. The construction of a high-fidelity RCPN model of real-life system is the basis for further KPPIs analysis and prediction. For a complex production system, the model scale can be very large, which brings computational complexity in model analysis. In general, a discrete production system can be classified into different connected subsystems, and the subsystems at a same level are loosely connected through buffers, the systems from different levels are interacted to fulfil the production tasks. Therefore, a cloud-edge cooperation mechanism is applied when constructing the RCPN model which contains a main RCPN model and loosely connected sub-RCPN models. The sub-RCPN models are distributed analyzed and controlled at the edge resources side, and the high-level RCPN models which consisted of few abstract elements for sub-systems are analyzed at the cloud center. Thus, the computational complexity can be largely reduced and model accuracy can be ensured at the same time. In addition, smart elements (e.g., tokens, places, and transitions) are added in the RCPN model to exchange information with practical system.

Based on the constructed RCPN model, the real-time KPPIs analysis method can be involved to extract the KPPIs which are defined based on decision requirements from different production levels. Monitors will be set on the corresponding places or transitions of RCPN model to provide the KPPIs automatically when production system evolves. Then the RCPN model can be simulated for several rounds to predict future KPPIs, and a possible value or a distribution will be output for future PMRA method.

C. *Self-adaptive PMRA method*

This module is responsible for assigning the manufacturing resources before production exceptions happen. Production exceptions refer to a specific state that deviates from normal or planned operations. This paper focuses on three kinds of exceptions, i.e., tasks exceptions, resources exceptions and system exceptions. Based on the predicted status of KPPIs, the potential production exceptions can be forecasted in advance.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2021.3073404, IEEE Transactions on Industrial Informatics

To eliminate the adverse impact of exceptions, the self-adaptive PMRA strategy can be involved. The real-time status of manufacturing resources will be used when obtaining the optimal production schedule. Then, the production decisions can be released to the smart objects which will execute the production tasks accurately. Since the production decisions are made according to the current status of resources, the PMRA decisions can be more feasible and executable. Many optimization methods can be used to find the optimal assignment decisions, e.g., support vector machine (SVM) and NSGA-II. Among the algorithms, the NSGA-II algorithm can give an optimal result with a low computation complexity, thus this paper selects NSGA-II as the optimization algorithm.

IV. RCPN-ENABLED KPPIS ANALYSIS AND PREDICTION

The overall work logic of RCPN-enabled KPPIs analysis and prediction is shown in Fig. 2, which consists of three main aspects: RCPN model construction, real-time KPPIs analysis, and future KPPIs prediction.

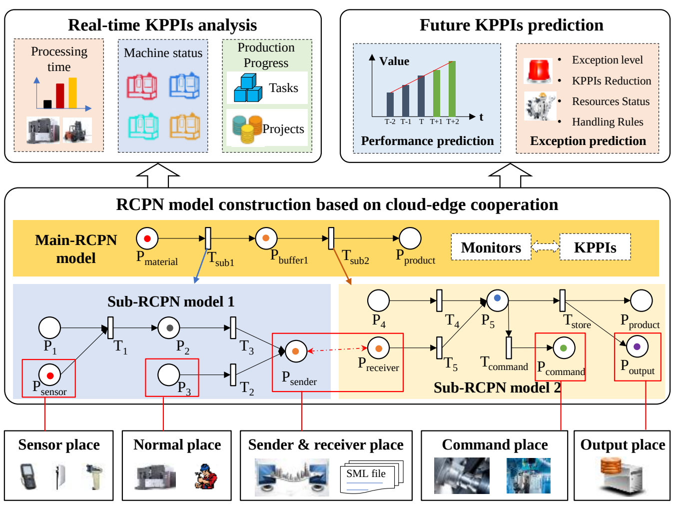


Fig. 2. RCPN-enabled KPPIs analysis and prediction

A. RCPN model construction

In a smart complex factory, the centralized modelling and analysis methods need large computational capability and storage capacity. The distributed control methods can be computationally tractable but may give up performance optimality. Therefore, a cloud-edge cooperation mechanism is proposed to construct the RCPN model. A main RCPN model is created to modelled production system at an abstract level. The sub-production systems are modelled with distributed and connected sub-RCPN models whose performance analysis can be executed on edge-resources side. The main RCPN and sub-RCPN models are interacted and collaborated through few elements. Since the RCPN model is extended from CPN model with dynamic synchronization capacity, it is necessary to give a definition for the CPN model and RCPN model before describing the construction method.

A CPN can be defined as N=<P, T, C, I, O, M>, where, P refers to the places for resources status; T refers to the transitions for activities; C denotes the colored mapping from PxT to W, where W is a finite and non-empty set, and complex manufacturing information can be represented in the token values and inscriptions of the models; I and O denote the forward and backward incidence matrix of PxT respectively, and they are used to describe the connections (arcs) between the places and transitions; M is a marking representing the number of tokens in P, and M_0 denotes the initial marking.

To fulfill the target of RCPN model construction for cloud-edge cooperation, the following measures are made: 1) intelligent interfaces are added to the places in CPN model, including sensor places and smart tokens to accept the real-time data from sensing devices, sender place and receiver place to work as information communication bridges between different RCPN models, command places to transmit production commands to resources, and output places to upload production information to databases or applications; 2) time information can be added to describe the consumed time of production activities, which can be added either in the places or transitions; 3) macro transitions are set to represent integrated status or processes for sub-systems, so that the hierarchical structure of factory can be described in RCPN models, and the sub-RCPNs are loosely connected; 4) Constraint functions are added to model the rules of resources assignment in production system, for example, guard functions can be set on transitions to describe the processing sequence of resources at machines, arc functions can be added to the connection arcs so that the information attached to the resources can be changed following practical processing rules.

Therefore, A RCPN can be defined as RCPN=<P, T, C, I, O, M, G, E, D >, where, places set P=<P_n, P_s, P_c, P_{se}, P_r, P_o> contains six sub-sets: normal place (P_n) as places in CPNs, sensor place (P_s), command place (P_c), sender place (P_{se}), receiver place (P_r), output place (P_o); transitions set T=<T_i, T_t, T_m > contains three sets: immediate transitions (T_i), timed transitions (T_t) and macro transitions (T_m); C, I, O, and M share the same meanings with CPNs; G denotes the guard function and maps each transition T to a Boolean expression; E denotes the arc function binding to each arc; D denotes the time delays consumed in tokens or transitions. The RCPN model can be constructed following five main steps.

- Step 1, outline the overview structure of the production system and transform all the abstract sets into RCPN elements to construct a main RCPN model.
- Step 2, analyze the abstract sets and construct a detailed RCPN model for each sub-system.
- Step 3, repeat the second step until any element is described by a primitive manufacturing event. There are many factors (e.g., machines, tools, materials) participating in a practical production processes, creating a RCPN model that considers all the states or elements in real-life system is intractable and useless. This paper focuses on the KPPIs analysis and prediction, especially the time-related KPPIs, e.g., progress, cycle time, thus only the critical factors that may affect the KPPIs status will be considered in the RCPN model construction.
- Step 4, set connection and communication elements (e.g., buffer places) between different connected RCPN models,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2021.3073404, IEEE Transactions on Industrial Informatics

and bind the smart tokens and places with real-life production resources, so that the RCPN model can evolve with practical system.

- Step 5, set transition priority-based controllers to prevent system deadlocks. For each RCPN model, the reachability graph or siphons can be calculated to find the deadlock markings. Then, the transitions related to each deadlock making can be selected, and different priorities can be designed based on their processing information. The transition with a high priority will be triggered preferentially than the transition with a low priority. As a result, the liveness of the RCPN model can be achieved.

After a RCPN model is constructed, it can be seen as a mirror of real-life production system, and the real-time KPPIs analysis and future KPPIs prediction can be fulfilled based on the simulation of RCPN model.

B. Real-time KPPIs analysis

Different managers in production system often cares different KPPIs which can be classified into four levels:

- Sensing level. The sensing-level events are obtained from the IIoT and CPS devices, which record the unit production status or activities. The KPPIs include the activity or status of targeted resources (e.g., number, location).
- Resource level. Resource-level events contain five kinds: material distribution events, processing and assembly events, work-in-progress (WIP) delivery events, quality detection events, and storage events. The KPPIs at this level include processing time of WIPs, operational condition of resources, and waiting time of tasks, etc.
- Cell level. A cell is a set of machines or a production line that is responsible for manufacturing several products or parts. The KPPIs at this level include average processing time of a WIP that passes this cell, qualified product or WIP rate, average production load, etc.
- Factory level. The factory-level KPPIs reflect the whole status of production system, including total progress of projects, average processing time of projects, etc.

After the requirements of KPPIs analysis are obtained, monitors can be set on the corresponding places and transitions. Three kinds of monitors are important: data collector monitor, break point monitor, and write in file monitor. The data collector monitor is used to record the information changes of places or transitions, such as material number and delivery frequency of buffers. The break point monitor is used to stop the current production events when some conditions are satisfied, e.g., when an electric trolley works for a too long time, the RCPN model can detect it timely, and a charging instruction can be released to the trolley. The write in file monitor can publish tracked information to the files for the usage of other applications.

Since the smart tokens and places in the RCPN model are connected with real-life smart resources, their status can change with real-life resources dynamically. Besides, the position, number and attributes of tokens in RCPN model can be timely changed based on the guard functions and consume rules:

$$\forall P_i \in P: M'(P) = M(P) + O(P,T) - I(P,T) \qquad (1)$$

Where, $M'(P)$ denotes the new state of the place; $M(P)$ denotes the current state of the place and $I(P,T)$ and $O(P,T)$ denote the output function and input function, respectively.

Based on the evolution of RCPN models, the real-time KPPIs can be traced and obtained dynamically. Firstly, the dynamic markings of tokens can be tracked to record real-time status of resources. Secondly, the firing operation of transitions can be tracked to obtain the throughput, busy time, processing time, deviation from planned time, etc. Thirdly, the performance analysis of the RCPN model can be taken to obtain the cycle time, the average tasks queue length, production cycle time, etc.

C. Future KPPIs prediction

After a RCPN model is constructed, the uncertain factors can be further added, and the future KPPIs can be predicted by the simulation of RCPN model. Since there are many uncertainties in real-life production system, it is hard and impossible to give a specific value. This paper presents a prediction method for a possible value or distribution of the future KPPIs.

Firstly, the status of manufacturing resources that participate in the production activities is calculated. There are many uncertain factors in real-life production which should be considered in the prediction process. For example, new orders often arrive randomly, machines broken down occasionally, different operators may take various time to complete a same process, machines perform differently under different conditions, etc. It is necessary to consider the resources difference in the RCPN model.

Secondly, simulate the model for several times to give predictions for the future KPPIs. The status of RCPN models is updated with real-life status of manufacturing resources. The difference of resources capacity and potential uncertain factors are considered in the RCPN model. As a result, the real-life production status can be described in the RCPN model to a great extent, and the simulation of RCPN model can give a better prediction for the KPPIs.

Thirdly, extract the distributions based on the predicted data set of KPPIs and output. The production distribution of KPPIs can be calculated based on the following steps: 1) Divide the dataset into *r* ranges equally. 2) Sort the variable, and the first *k* values less than or equal to the upper limit of the first bin are counted as the frequency of the first bin. 3) Repeat the counting step 2 for each bin until all the data is classified. 4) Draw the bars to represent the counts. 5) Find the probability distribution based on the chart of bar distribution.

V. Self-adaptive PMRA method

The self-adaptive PMRA method aims to find an optimal combination between manufacturing tasks and smart resources according to the real-time and future KPPIs. The overall architecture of the PMRA method is shown in Fig. 3. It includes three main parts, i.e., production demands analysis, real-time production status extraction and self-adaptive PMRA.

The production demands analysis forecasts the production exceptions and releases the related management requirements.

There are three kinds of exceptions, i.e., tasks exceptions, resources exceptions and system exceptions. The tasks exceptions contain two parts: urgent tasks and tasks delivery delays. The urgent tasks can be predicted according to the frequency distribution analysis of historical tasks. The tasks delivery delays and resources exceptions can be predicted by comparing the future KPPIs with the planned values. For example, the tasks may not be completed in time, a machine lacks of tools, the material inventory is lower than threshold value. The system exceptions can be forecasted by comparing two KPPIs at adjacent time. After the exceptions are predicted, the related exception handling requirements will be released to the decision center. When two or more requirements arrive at a same time, the requirements will be evaluated and ranked based on the historical similar cases and expert knowledge. The requirement that will bring more serious influence will be given a higher priority, and will be tackled firstly.

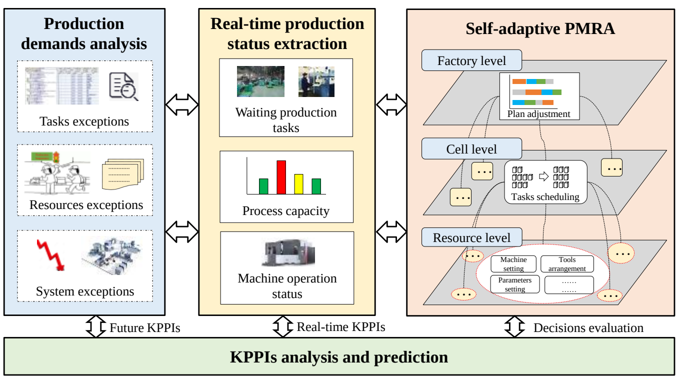


Fig. 3. Self-adaptive PMRA method

In order to meet the requirements, the real-time status of related production resources can be extracted from the real-time KPPIs information set. Three kinds of real-time production status are considered from different management levels: machine operation status at resources level, production capacity at cell level and waiting production tasks at factory level.

The self-adaptive PMRA aims to make more practical and optimal production decisions before production exceptions happen based on the real-time information. Under the cloud-edge cooperation environment, the decision-making processes use a self-adaptive mechanism which follows a down-to-top rule. Once an exception occurrence is predicted, the smart resources take measures firstly. For example, a machine maintenance measure can be taken to improve its efficiency. Then the possible improved machine performance and the required maintenance time will be updated to the RCPN model. Based on the simulation result of the new RCPN model, the future KPPIs can be predicted to evaluate whether the exception can be solved. If the problem cannot be solved, the cell-level production adjustment will be involved, such as production queue changes, and the effectiveness of the adjustment measures can be evaluated by the RCPN model. If the problem still cannot be solved, the factory-level production tasks plan needs to be adjusted.

When an exception needs factory-level adjustment, the NSGA-II based PMRA method will be involved. A new production task set will be created by combining the unfinished production tasks set (T) with the new tasks set (T'), which can be denoted as TN=<T, T'>=<T_1, T_2, …, T_n>. Before making PMRA decisions, it is necessary to give a mathematical model of the problem, which is listed as follows.

Objective function 1:

$$\min\ f_1 = C_m = \sum_{i=1}^{n} \beta_i \bullet \max(0, C_{ij} - D_i), i \in [1,n], j \in [1, n_i] \tag{2}$$

Objective function 2:

$$\min\ f_2 = E = \sum_{i=1}^{n} \sum_{j=1}^{K} \sum_{k=1}^{m_j} (P_{jk} t_{ijk} x_{ijk}) \tag{3}$$

Subject to:

$$C_{ij} - C_{ij-1} \geq t_{ijk} \bullet x_{ijk},\ j \in [2, K] \tag{4}$$

$$C_{i1} > r_i, i \in [1, n] \tag{5}$$

$$\sum_{k \in M(O_{ij})} x_{ijk} = 1, \forall i, j \tag{6}$$

For objectives in equations (2) and (3), f_1 denotes the total tardiness, f_2 denotes the total energy consumption. Constraint (4) ensures the precedence relationships. Constraint (5) ensures that the starting time of job i must begin after its release time. Constraint (6) ensures that each job is processed only on one machine. The responding notations is listed in Table I.

TABLE I
NOTATIONS

Notations	Description
$J_i (i = 1, \ldots, n)$	Set of jobs, n denotes the total number of jobs
D_i	The due date of job J_i
K	Total number of processes of job J_i
O_{ij}	The j^{th} process of job J_i
$M_{jk} (k = 1, \ldots, m_j)$	The set of machines at jth process, m_j denotes total number of machines at jth process
C_{ij}	The completion time of O_{ij}
E	The total energy consumption of machines
x_{ijk}	1, if M_{jk} is selected for O_{ij}; 0, otherwise
P_{jk}	The processing power of M_{jk}
t_{ijk}	The processing time of O_{ij} on M_{jk}

Then, the NSGA-II algorithm can be used to make optimal PMRA decisions. The key steps of NSGA-II algorithm for PMRA method are briefly described in Fig. 4. Firstly, construct initial parent population, and each individual (X) of population can be seen as a chromosome. Secondly, use the crossover or the mutation operator to create a new population. Then, combine the new and initial population and construct an optimal population using the crowding distance sorting method. Based on this, a round of iteration is ended, and the following iteration can be executed until the termination condition is met. Lastly, a pareto optimal solution set can be achieved, all the individuals in the set are the optimal schedules, and the managers can select one to output according to their demands.

//**Algorithm for the NSGA-II based PMRA method**
//**Input**: Iteration number M; Population size N; Mutation rate P_m; Due date of tasks d; $t_{i,j,k}$; E
//**Output**: Production task assignment schedule X
(1) **Initialization**. farm{ }=cell(1, N) → farm{ }=X; // Since the case factory has a same number of processes for each job (denoted as K), thus a two-dimensional matrix-based method is used to form the chromosomes. Chromosome X is defined as a n x K matrix:

$$X = \begin{bmatrix} M_{11} & M_{12} & \dots & M_{1K} \\ M_{21} & M_{22} & \dots & M_{2K} \\ \dots & \dots & M_{ij} & \dots \\ M_{n1} & M_{n2} & \dots & M_{nK} \end{bmatrix}$$

where, n represents the total number of jobs. The element M_{ij} in the matrix denotes the allocated machine of job J_i in kth process, and it consists of an integer g ($1 \leqslant g \leqslant m_j$).
(2) **While** counter<M
(3) **For** i=1:2:N
(4) **If** P_m<rand(1) //Random create a number, if the number exceeds P_m, use the crossover operator
(5) $X_1 \times X_2 \rightarrow x1$, $x2$; //Random choose two chromosomes, i.e., X_1 and X_2, exchange the genes to create two new offspring, i.e. x_1 and x_2. Two crossover directions can be used for the proposed chromosomes: vertical and horizontal. After the crossover direction is selected, two crossover points are chosen to form two crossover sections randomly, and the genes in crossover sections are exchanged to generated new offspring.
(6) **Else** // If the number is smaller than P_m, use the mutation operator
(7) $X_1 \rightarrow x_1$; //Create x_1 from X_1 using the mutation method
(8) **End; End**
(9) newfarm{ }=[X, x]; //Combine the parent and offspring population
(10) [fitness1, fitness2]=Objective(farm); //
(11) **For** i=1:2N
(12) front(X)=sort_mod(fitness1, fitness2); //Rank the populations
(13) distance(X); **End**
(14) farm=choose(newfarm); **End** //Choose top N individuals
(15) **For** i=1:N
(16) **If** front(X)==1
(17) Farm{i}=X;
(18) **End; End**
(19) **Output** Pareto optimal solution set; **End**

Fig. 4. Solution procedure

VI. CASE STUDY

A. Case description

An industrial case from a collaborate company is used to demonstrate the proposed PMRA method. The company is a typical manufacturer for CNC machine tools in North China. The selected factory mainly manufactures spindle boxes. With a 4-week investigation at its factory, it is found that the manufacturing information does not accurately and promptly reflect the real-time situation. As a result, the production abnormalities cannot be found and tackled timely. Besides, the production system uses the passive MRA method which configures the machines after the occurrence of production exceptions. Under this situation, a long waiting time is often consumed for tasks and much unnecessary energy consumption of machines is wasted. Therefore, the company is truly in need of the proposed PMRA method.

To demonstrate the advantages of the proposed PMRA method, the advanced IIoT and CPS technologies are applied in the factory to provide a cloud-edge cooperation environment, where the proposed RCPN model-based KPPIs analysis and prediction, self-adaptive PMRA method can be enabled. As shown in Fig. 5, the production processes contain 8 main stages, each stage has several parallel machines. The spindle box materials need to be manufactured through all the stages in sequence.

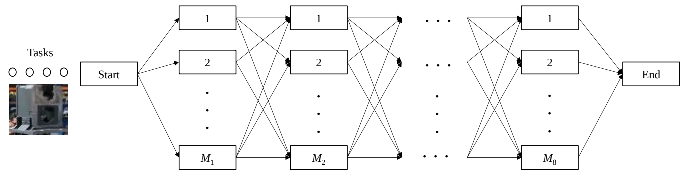
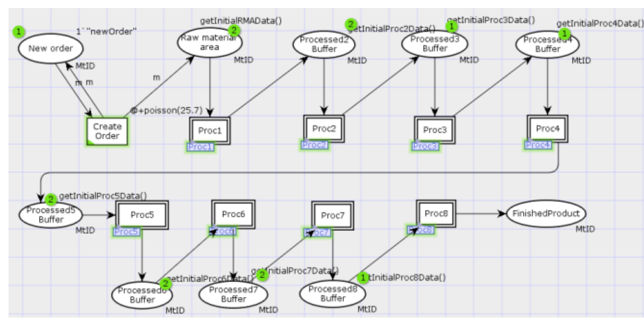


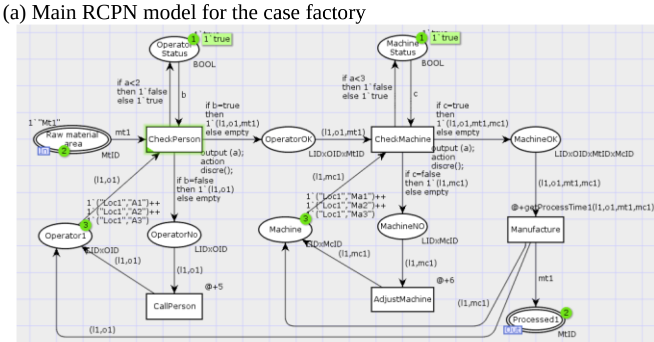
Fig. 5. The processes of spindle box

B. RCPN-enabled KPPIs analysis and prediction

Referring to the production processes in Fig. 5, an RCPN model is built up in CPN tools as seen in Fig. 6. Fig. 6(a) presents one overall RCPN model for the whole production system, and it has eight macro-transitions each of which is linked to a sub-RCPN model for each process, respectively. To simplify the presentation, only the model for process 1 is given in Fig. 6(b).



(a) Main RCPN model for the case factory



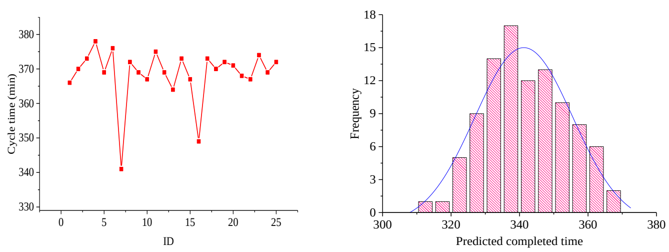
(b) Sub-RCPN model for process 1
Fig. 6. RCPN model for the case factory

Table II presents the global color declaration and initialization for the RCPN model. In the model, the color set includes two main kinds: simple color sets and compound color sets. The simple color sets carry only one kind of information, i.e., Boolean, integer, string. For example, the *MtID* refers to the material ID. The compound color sets carry two or more kinds of information, for example, the *LID*x*OID*x*MtID* refers to the element contains three attributes, i.e., location ID, operator ID, and material ID. The initialization of RCPN model is used to update the real-time information to the constructed RCPN model. Since the CPN tools can only read the 'sml' file, the real-time production information is stored in a 'sml' file (e.g., InitialStatus.sml), and the places or transitions in the model can obtain the real-time information through the instructs 'getInitialRMAData()', 'getInitialProc2Data()', etc.

TABLE II
THE GLOBAL COLOR SET DECLARATION AND INITIALIZATION

The global color set declaration and initialization for RCPN model
Colset MtID=string timed; Colset Bool=bool; ……
Colset LIDxOIDxMtID=product LID*OID*MtID timed;……
fun getInitialRMAData ( )= ["*MtA*10", "*MtB*10"]; fun getInitialProc2Data ( ) = ["*MtA*6", "*MtA*4"];fun getInitialProc3Data ( ) = ["*MtB*3"]; fun getInitialProc4Data ( ) = ["*MtA*5"];fun getInitialProc5Data ( ) = ["*MtA*3", "*MtA*6"]; fun getInitialProc6Data ( ) = ["*MtB*5", "*MtC*4"];……
fun getProcssTime1(*l*1:LID, *o*1:OID, *mt*1:MtID, *mc*1:McID)=Case ("*L*1", "*A*1", "*MtA*2", "*Ma*2") =>70| ……

Based on the status of RCPN model, the enabled transitions will be triggered and the real-time KPPIs can be obtained. Once new real-time manufacturing data is obtained, the related information in the 'sml' file will be updated, and the RCPN model can be updated accordingly. As a result, the extracted real-time KPPIs can be kept as practical as possible. Fig. 7(a) takes the cycle time of spindle boxes as an example, which records the changes of different spindle boxes.



(a) cycle time analysis (b) completed time prediction

Fig. 7. The result of KPPIs analysis and prediction

In order to predict the KPPIs, the uncertain production factors should be considered in the RCPN model, for example, new production orders often arrive randomly. In Fig. 6(a), a 'new order' place is added in the RCPN model to describe the potential order arrivals, which can be extracted from historical production data or defined by the production planning managers. After the RCPN model is obtained, it can be simulated many times to forecast the future status of production system. For example, if a material is the predicted object, a mark can be attached to its virtual corresponding elements in RCPN model when it enters the production system. Then the potential entering and leaving time of the material in different processes can be predicted based on the simulation result. Fig. 7(b) shows a prediction set for the completed time of a material. It can be shown that the possible distribution is a Gaussian distribution. Then, the specific values for the distribution can be extracted from the predicted dataset, i.e., Gaussian (341, 12). If a period value is required, then a section value based on its average (μ) and the standard deviation (σ) can be given, e.g., $[\mu-2\sigma, \mu+2\sigma]$. For the case, the period [317, 365] can be seen as the prediction and output to the upper-level managers. On the other side, if a specific value is needed, the average value (341) can be given as the prediction value.

C. *Self-adaptive PMRA method*

After the KPPIs are predicted, the potential production exceptions can be forecasted and the self-adaptive PMRA method can be enabled. To demonstrate the proposed PMRA method, a production case is extracted from real-life production processes. The experimental tests are carried out utilizing the MATLAB programming language on a personal computer with Intel Pentium (R) with a 4-GB RAM and 2.8-GHz processor.

The factory mainly processes 5 types of normal jobs (P_1-P_5) and one urgent job (P_6). In the real-life production, the MRA process uses the passive mode and takes the processing time as stable values shown in Table III. However, the processing time and energy consumption of jobs often changes due to the uncertainties in real-life production. In the factory, the main kind of consumed energy is the electricity, and the total energy consumption of a job processed on a machine can be obtained based on the multiplication of machine power and processing time. The real-life processing time and machine power is listed in Table IV. Here, the type of job J_1 is P_1, the type of J_2 and J_3 is P_2, the type of J_4- J_6 is P_3, the type of J_7 and J_8 is P_4, the type of J_9 and J_{10} is P_5. The first column presents the machines, M_{ab} refers to the bth machine for ath stage. There are two lines for each machine, the first line presents the real processing time (min) and the second line shows the real machine power (kW). The last line presents the due-date of each job.

TABLE III
THE PLANNED PROCESSING TIME OF JOBS

Type	Processing time (min) for each process							
	1	2	3	4	5	6	7	8
P_1	72	21	65	34	58	45	39	61
P_2	52	47	47	16	50	38	26	45
P_3	37	35	58	31	59	15	47	56
P_4	38	60	53	59	51	47	37	65
P_5	65	11	56	31	40	40	30	30
P_6	15	23	21	47	47	85	67	82

Based on the dataset, the PMRA method can be executed, and a pareto optimal solution set which contains many optimal individuals is obtained. This paper selects the individual with the minimum Euler distance as the default output.

To illustrate the efficiency and feasibility, two comparisons are designed. Firstly, the real-life passive production assignment scheduling is obtained according to the dataset in Table III, where the NSGA-II algorithm is also selected as the optimization algorithm. Table V shows the comparison results between the PMRA method and the real-life production assignment. It can be found that the PMRA method can largely reduce the total tardiness and energy consumption simultaneously.

Secondly, the exception handling efficiency is compared. Three kinds of production exceptions are designed according to real-life production system: 1) an urgent order will arrive at time t_1=200, the processing time is listed at Table III; 2) at time t_2=300, the tool in machine M_{42} need to be replaced, which will take 20 minutes; 3) at time t_3=450, machine M_{82} has to be maintained for 30 minutes. Two exception handling methods are compared. One is the real-time scheduling method which can detect the exceptions timely and give adjustment decisions after the exceptions happen. The second method is the right shift method where the related processes need to wait until the exceptions are handled. For the proposed PMRA method, the three exceptions are assumed to be predicted at time t_0=100, and the proactive decisions are executed at t_0. The comparison results are also shown Table V. From the comparisons, it can be found that the proposed PMRA method performs better both in the total tardiness and the total energy consumption.

TABLE IV
THE REAL-LIFE PROCESSING TIME AND MACHINE POWER

M	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}
M_{11}	72	52	54	37	37	37	38	38	65	65
	3.1	2.9	2.9	3.8	3.7	3.7	2.9	2.9	3.2	3.2
M_{12}	70	55	54	39	39	39	34	34	69	70
	3.5	2.7	2.7	3.6	3.6	3.6	3.3	3.2	2.4	2.7
M_{13}	73	50	50	35	35	35	42	42	70	65
	4.1	3.3	3.3	3.9	3.9	3.9	4	4	3.3	3.2
M_{21}	23	47	47	35	35	35	60	60	11	11
	6	8.3	8.3	8.1	8.2	8.2	6.9	6.9	7.2	7.2
M_{22}	21	47	47	37	36	34	60	60	11	13
	6	8.5	8.5	8.5	8.1	8.1	6.3	6.3	8.2	8.2
M_{31}	67	47	46	58	58	57	53	52	56	54
	2	4	4	3.2	3.2	3.1	3	3	2.7	2.7
M_{32}	62	44	45	60	60	60	50	50	62	50
	2.1	4	4	2.8	2.8	2.8	2.7	2.9	2.4	2.3
M_{33}	67	50	50	56	56	55	56	55	60	62
	2.3	4.3	4.3	3	3	3	3	3	2.5	2.5
M_{41}	34	16	16	32	31	33	59	59	31	31
	2.9	3	3	2.8	2.8	2.8	2	2	1.7	1.7
M_{42}	35	16	16	31	30	30	64	62	32	34
	2.6	2.8	2.8	2.8	2.8	2.8	2	2	1.6	1.6
M_{51}	58	50	51	59	59	57	51	50	40	42
	6	8.5	8.5	8.5	8.1	8.1	6.3	6.3	8.2	8.2
M_{52}	55	52	52	59	60	60	48	49	37	37
	5.9	8.4	8.4	8.4	8	8	6.2	6.2	8.1	8.3
M_{53}	60	49	49	58	58	58	53	53	40	42
	6.3	8.4	8.6	8.6	8.4	8.5	6.4	6.3	7.9	7.9
M_{61}	44	38	37	15	15	14	47	46	40	39
	4.4	4.2	4.3	3.3	3.2	3.8	3.8	3.7	4	4
M_{62}	46	36	37	16	15	14	49	49	34	35
	5	4.4	4.3	3.3	3.2	3.5	3.5	3.5	4.3	4.3
M_{71}	38	26	26	47	47	47	37	36	30	31
	2.5	2.3	2.3	2.3	2.4	2.4	2.5	2.4	2.2	2.2
M_{72}	37	25	25	49	46	46	34	35	25	26
	2.4	2.2	2.3	2.4	2.5	2.5	2.4	2.3	2.5	2.3
M_{81}	64	45	45	57	56	56	65	63	30	30
	1.2	1.9	1.9	2.8	2.8	2.8	2.2	2.2	3.4	3.2
M_{82}	60	46	46	56	55	55	64	67	33	32
	1	2	2	3	3	3	2	2	3	3
M_{83}	64	44	44	55	58	54	65	65	27	27
	1.3	2.2	2.2	3.4	3.4	3.4	2.4	2.3	3.5	3.5
D	476	265	532	395	526	651	435	511	316	422

TABLE V
COMPARISON RESULTS

Solutions	Normal production		Exception handling	
	Tardiness (min)	Energy consumption (kWh)	Tardiness (min)	Energy consumption (kWh)
PMRA method	751	159	785	180.4
Real-life production assignment	974	221	/	/
Real-time scheduling	/	/	935	194.2
Right shift method	/	/	1238	202.9

VII. CONCLUSION

Recently, IIoT and CPS technologies have been widely applied in smart factory. The traditional factories are made smart with the capacity of dynamic sensing, real-time communication and self-adaptive control. However, the efficiency of MRA method in factory is often constrained due to the passive and afterwards mode. In this work, a PMRA method based on production performance prediction has been proposed to provide a new paradigm for smart factory to enhance the efficiency of MRA and ensure the optimal operation of production system.

The main contributions of this work include: Firstly, an all-in-one PMRA method based on production performance prediction and its overall architecture is proposed, which can be used to manage the production system through a cloud-edge cooperation mechanism and to provide MRA decisions predictively. Secondly, by establishing a RCPN-based KPPIs analysis and prediction method, the real-time status of manufacturing resources can be dynamically mapped in RCPN models, thus the analysis and prediction capacity of PN model can be used to obtain real-time KPPIs and predict future KPPIs. Thirdly, by using the self-adaptive PMRA strategy, the production tasks can be assigned to the resources before production exceptions happen, and real-time status of manufacturing resources can be used when making production decisions so that the production orders can be more feasible.

There are three main challenges waiting to be solved. The first challenge is how to give a quantitative analysis mechanism for the accuracy and compactness for the RCPN model of a real-life manufacturing system. The second challenge is how to combine the more comprehensive big data analytics with RCPN model to provide a more accurate prediction of KPPIs. In addition, how to easily deploy and apply the key technologies of PMRA method to complex and dynamic manufacturing system is also an important issue.

REFERENCES

[1] Y. Zhang, Z. Zhu, and J. Lv, "CPS-based smart control model for shopfloor material handling," IEEE Trans. Ind. Informatics, vol. 14, no. 4, pp. 1764–1775, 2018.
[2] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, "Intelligent Manufacturing in the Context of Industry 4.0: A Review," Engineering, vol. 3, no. 5, pp. 616–630, 2017.
[3] H. Hu, X. Jia, Q. He, S. Fu, and K. Liu, "Deep reinforcement learning based AGVs real-time scheduling with mixed rule for flexible shop floor in industry 4.0," Comput. Ind. Eng., vol. 149, p. 106749, 2020.
[4] T. Qu, S. P. Lei, Z. Z. Wang, D. X. Nie, X. Chen, and G. Q. Huang, "IoT-based real-time production logistics synchronization system under smart cloud manufacturing," Int. J. Adv. Manuf. Technol., vol. 84, no. 1–4, pp. 147–164, 2016.
[5] P. M. Khodke and A. S. Bhongade, "Real-time scheduling in manufacturing system with machining and assembly operations: a state of art," Int. J. Prod. Res., vol. 51, no. 16, pp. 4966–4978, 2013.
[6] D. Mourtzis and E. Vlachou, "A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance," J. Manuf. Syst., vol. 47, pp. 179–198, 2018.
[7] Y. Lu, C. Liu, K. I. Wang, H. Huang, and X. Xu, "Digital Twin-driven smart manufacturing: connotation, reference model, applications and research issues," Robot. Comput. Integr. Manuf., vol. 61, p. 101837, 2020.
[8] F. Tao, Y. Zuo, L. Da Xu, L. Lv, and L. Zhang, "Internet of things and BOM-based life cycle assessment of energy-saving and emission-reduction of products," IEEE Trans. Ind. Informatics, vol. 10, no.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2021.3073404, IEEE Transactions on Industrial Informatics

2, pp. 1252–1261, 2014.

[9] F. Tao, M. Zhang, Y. Liu, and A. Y. C. Nee, “Digital twin driven prognostics and health management for complex equipment,” CIRP Ann. Technol., vol. 67, no. 1, pp. 169–172, 2018.

[10] L. Monostori, B. Ka, T. Bauernhansl, S. Kondoh, and S. Kumara, “Cyber-physical systems in manufacturing,” CIRP Ann. - Manuf. Technol., vol. 65, pp. 621–641, 2016.

[11] J. Lee, C. Jin, and B. Bagheri, “Cyber physical systems for predictive production systems,” Production Engineering Research and Development, vol. 11, no. 2. pp. 155–165, 2017.

[12] A. J. Álvares et al., “Development of a cyber-physical framework for monitoring and teleoperation of a CNC lathe based on MTconnect and OPC protocols,” Int. J. Comput. Integr. Manuf., vol. 31, no. 11, pp. 1049–1066, 2018.

[13] J. Leng, H. Zhang, D. Yan, Q. Liu, X. Chen, and D. Zhang, “Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop,” J. Ambient Intell. Humaniz. Comput., vol. 10, no. 3, pp. 1155–1166, 2019.

[14] J. Cheng, H. Zhang, F. Tao, and C. F. Juang, “DT-II:Digital twin enhanced Industrial Internet reference framework towards smart manufacturing,” Robot. Comput. Integr. Manuf., vol. 62, p. 101881, 2020.

[15] N. Du, H. Hu, and M. Zhou, “A Survey on Robust Deadlock Control Policies for Automated Manufacturing Systems with Unreliable Resources,” IEEE Trans. Autom. Sci. Eng., vol. 17, no. 1, pp. 389–406, 2020.

[16] H. Hu, Y. Liu, and L. Yuan, “Supervisor simplification in FMSs: comparative studies and new results using Petri Nets,” IEEE Trans. Control Syst. Technol., vol. 24, no. 1, pp. 81–95, 2016.

[17] N. Wu and M. Zhou, “Intelligent token petri nets for modelling and control of reconfigurable automated manufacturing systems with dynamcial changes,” Trans. Inst. Meas. Control, vol. 33, no. 1, pp. 9–29, 2011.

[18] Y. Lv, C. K. M. Lee, H. K. Chan, and W. H. Ip, “RFID-based colored Petri net applied for quality monitoring in manufacturing system,” Int. J. Adv. Manuf. Technol., vol. 60, no. 1–4, pp. 225–236, 2012.

[19] Y. Zhang, W. Wang, N. Wu, and C. Qian, “IoT-enabled real-time production performance analysis and exception diagnosis model,” IEEE Trans. Autom. Sci. Eng., vol. 13, no. 9, pp. 1318-1332, 2016.

[20] N. Li, Y. Li, M. Sun, H. Kong, and G. Gong, “An optimization method for task assignment for industrial manufacturing organizations,” Appl. Intell., vol. 47, no. 4, pp. 1144–1156, 2017.

[21] M. Yousefi Ramandi, N. Bigdeli, and K. Afshar, “Stochastic economic model predictive control for real-time scheduling of balance responsible parties,” Int. J. Electr. Power Energy Syst., vol. 118, p. 105800, 2020.

[22] Y. F. Wang, “Adaptive job shop scheduling strategy based on weighted Q-learning algorithm,” J. Intell. Manuf., vol. 31, no. 2, pp. 417–432, 2020.

[23] J. Leng, P. Jiang, C. Liu, and C. Wang, “Contextual self-organizing of manufacturing process for mass individualization: a cyber-physical-social system approach,” Enterp. Inf. Syst., vol. 14, no. 8, pp. 1124–1149, 2020.

[24] G. Zhang, C. H. Chen, P. Zheng, and R. Y. Zhong, “An integrated framework for active discovery and optimal allocation of smart manufacturing services,” J. Clean. Prod., vol. 273, p. 123144, 2020.

[25] S. Wang, S. Bi, and Y. A. Zhang, “Reinforcement learning for real-time pricing and scheduling control in EV charging stations,” IEEE Trans. Ind. Informatics, vol. 17, no. 2, pp. 849–859, 2021.

[26] N. Nikolakis, R. Senington, K. Sipsas, A. Syberfeldt, and S. Makris, “On a containerized approach for the dynamic planning and control of a cyber - physical production system,” Robot. Comput. Integr. Manuf., vol. 64, p. 101919, 2020.

Wenbo Wang received the B.S. degree in mechanical engineering from Huazhong Agricultural University, Wuhan, China, in 2012, the M.S. and Ph.D. degrees in mechanical engineering from Northwestern Polytechnical University, Xi'an, China, in 2015 and 2019 respectively.

He is currently a lecturer with the School of Mechanical Engineering at Jiangsu University, Zhenjiang, China. His research interests include smart manufacturing and predictive production optimization.

Yingfeng Zhang (Member, IEEE) received the B.Eng., M.S., and Ph.D. degrees in mechanical engineering from Xi’an Jiaotong University of China in 1999, 2002, and 2005 respectively.

He is currently a Professor with the Key Laboratory of Industrial Engineering and Intelligent Manufacturing, Ministry of Industry and Information Technology, School of Mechanical Engineering, Northwestern Polytechnical University, Xi’an, China. His research interests include smart manufacturing, digital twin and core models of future intelligent industrial systems.

Jinan Gu received the B.S., M.S., and Ph.D. degrees in mechanical engineering from Jiangsu University, Zhenjiang, China in 1985, 1994, and 2000 respectively.

He is currently a Professor with the School of Mechanical Engineering at Jiangsu University. His research interests include smart manufacturing, machine vision, intelligent robot mechanism design and optimization.

Jin Wang received the B.S. and M.S. degrees in mechanical engineering from Xi’an University of Science and Technology, Xi’an, China, in 2007 and 2010 respectively, the Ph.D. degree in mechanical engineering from Northwestern Polytechnical University, Xi’an, China, in 2019.

He is currently a lecturer with the School of Modern Posts at Xi’an University of Posts & Telecommunications, Xi’an, China. His research interest is real-time data-based production management.