Bottleneck detection for discrete manufacturing system based on object-oriented colored petri nets and cloud simulation

Yongyang Zhang[1, 2], *En'gao* Peng[1,*], *Zhenkun* Luo[1], *Lianghua* Zeng[1], and *Chun* Li[1]

[1]School of Industrial Automation, Beijing Institute of Technology, 519088 Zhuhai, China
[2]Chongqing Zhangsi Technology Co., LTD, 40014 Chongqing, China

Abstract. To identify the bottleneck unit of discrete manufacturing system dynamically, an intelligent detection method of bottleneck (IDMOB) is proposed by combining object-oriented colored Petri nets (OOCPN) and cloud simulation. With the proposed method, the production cells of manufacturing system are abstracted as several Petri net sub-modules. And the messages are passed through mapping functions and message passing gates among Petri net modules. Then, the cloud simulation is created to simulate the real processing, and the production data and capacity function are plugged into the cloud to detect the bottleneck unit intelligently. Finally, the effectiveness of IDMOB method is applied and validated in a practical case study from the power transformer industry.

Keywords: Intelligent detection method of bottleneck, Object-oriented colored petri nets, Cloud simulation, Discrete manufacturing system.

1 Introduction

With the rapid development of mobile internet, especially affected by COVID-19, people's consumption habits have changed dramatically. Mass-produced products are being replaced by customized, personalized, or digital products [1]. And some key technologies such as the cyber physical systems (CPS), internet of things (IoT), cloud computing, and big data analytics (BDA) are utilized to address the dynamics and fluctuations of the global market [2]. In addition, lean management methods have been used to increase the productivity in value-added processes and to optimize the organization [3].

It is well-known that the productivity of a production system is constrained by the bottlenecks. As a result, it is crucial to detect the bottleneck unit to maximize the benefit of the system [4]. Many recent studies have been carried out on this topic. Mukund Subramaniyan has proposed an active period based data-driven algorithm to predict throughput bottlenecks in the production system [5]. A number of research efforts focussed on identifying the bottleneck position with algorithms, such as simulated annealing

* Corresponding author: 863631723@qq.com

algorithm, deep neural network, long-short term memory (LSTM) neural network, and heuristic algorithm [6-9]. Unfortunately, these approaches are difficult to realize data sharing with complex modelling and computation.

As a powerful tool in cloud computing [10], cloud simulation provides a new idea for the upgrading of manufacturing and plays an essential role in Industrial 4.0. Thus, a method for bottleneck detection in real-time is proposed to shorten the production cycle time and increase the system capacity combined with OOCPN and cloud simulation.

2 Methodology

2.1 Conception

Intelligent detection method of bottleneck (called IDMOB) is a new detecting approach, which combines object-oriented colored petri nets and cloud simulation to abstract the production units as cloud places, and plug the places into the cloud simulation. Alternatively, the processing data and capacity function are saved in the cloud server to detect the bottleneck units intelligently by running the model.

2.2 Definition

Intelligent detection process of bottleneck can be defined as shown below [11]:

$$OOCPN = \left\{ P_i, T_j; IM_i, OM_i, I_i, O_i, C_i \right\} \tag{1}$$

where P_i is a place set, $P_i = \left\{ p_1, p_2, \cdots, p_m \right\}, m > 0$; T_j is a transition set, $T_j = \left\{ T_1, T_2, \cdots, T_n \right\}, n > 0$; $P \bigcup T \neq \varnothing$, $P \bigcap T = \varnothing$. IM_i is a message input place set; OM_i is a message output place set; I_i is the input mapping function from P to T: $C(P) \times C(T) \rightarrow N$ (natural number), $P = P_i \bigcup IM_i$, $T = T_j$, $I(P,T)$ is an incidence matrix; O_i is the output mapping function from P to T: $C(T) \times C(P) \rightarrow N$ (natural number), $P = P_i \bigcup OM_i$, $T = T_j$, $O(P,T)$ is an incidence matrix; C_i is a colored set of places.

2.3 Process

IDMOB method is an intelligent detection for bottleneck units with OOCPN and cloud simulation in discrete manufacturing system. The process of IDMOB is presented as Fig. 1 shows.

2.4 Abstract model construction

Production processes can be abstracted as a number of closed Petri net modules with OOCPN, and IM_i is the input from the previous process, OM_i is the output, P_i is the place (manufacturing place), T_k is the transition. Then the abstract model is established as shown in Fig. 2. The specific operations of each production units are encapsulated in a rounded rectangular box. Furthermore, the modules communicate with each other through input/ output mapping function and message gate to achieve messages passing. And the message gate is a special transition between two modules, which represents the “event”.

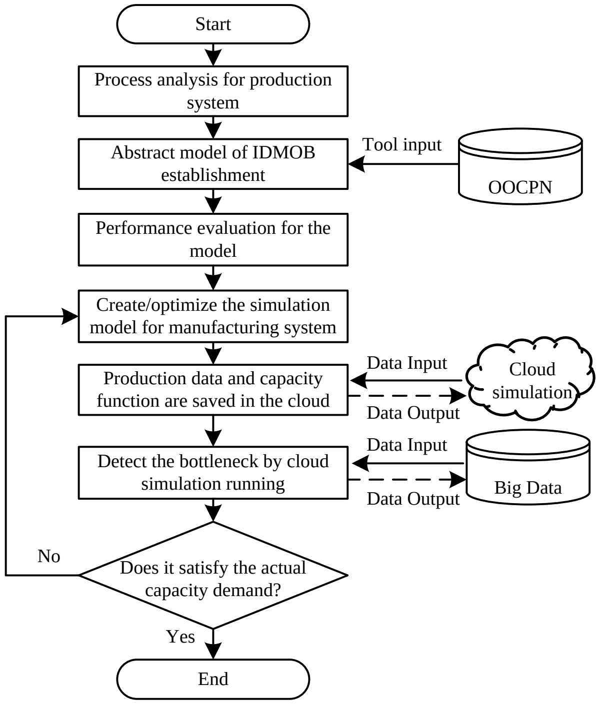

Fig. 1. Bottleneck detecting process based on IDMOB.

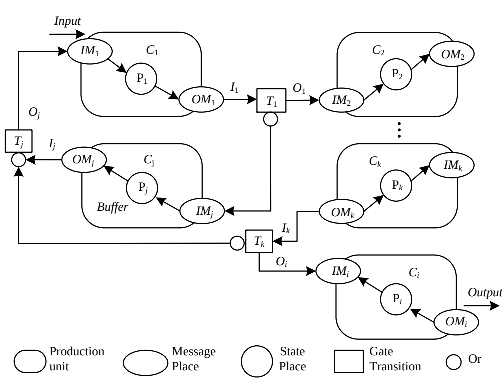

Fig. 2. Abstract model of IDMOB with OOCPN.

2.5 Performance evaluation

2.5.1 Deadlock analysis

According to the invariant theory, deadlocks can be detected by constructing object communication net among different categories [12]. Assuming every transitions and gates of Petri net modules can be fired through initial marking, there is no deadlock in the modules. Similarly, if all modules can be fired, there is no deadlock in the system.

2.5.2 Conflict analysis

There are two types of conflicts: input conflict and output conflict.

(1) Input conflict: It will emerge when more than one transition share the same input place or one transition has more than one input place, and they are connected to the transition with “or” relationship (Fig.3). In the actual production system, input conflict will take place when several production schedules require one machine to process at the same time.

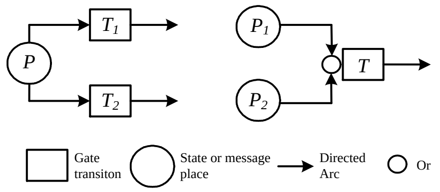

Fig. 3. Petri nets with input conflict.

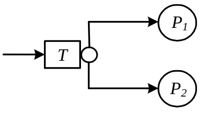

Fig. 4. Petri nets with output conflict.

(2) Output conflict: It will take place when one transition has more than one output places, and these places are connected to the transition with “or” relationship (Fig.4). In a manufacturing system, when the current procedure is accomplished, multiple machines are available for the next process, output conflict will occur.

2.5.3 Conservation analysis

According to the invariant theory, if there is an invariant *P* which is $n\times 1$ non-negative integer vector *x*, so that $x^T \times C = 0$ (*C* is an incidence matrix), thus the OOCPN is strictly constrained [13]. Therefore, the OOCPN of system is conservative and bounded, and the system is safe without overflow.

$$x^T \times C = 0 \tag{2}$$

2.6 Function model establishment

In manufacturing enterprises, capacity is calculated based on each processing procedure or production unit [14]. The formula of production capacity is stated as follows:

$$M = \frac{S \times F_e}{t} \tag{3}$$

$$t = \frac{1}{k}\sum_{i=1}^{k} t_i \cdot \theta_i \tag{4}$$

Where *M* is capacity of a production unit within the planning period, *S* are machines of one production unit, *Fe* is the effective processing time of one machine, *t* are machine hours for one product, t_i are machine hours for i^{th} product, θ_i is the ratio of i^{th} product in the planned volume, and k are varieties of the factory.

Moreover, the effective processing time of one machine is described in formula (5).

$$F_e = (1-p)\times 8\mu\times(1-\eta) \tag{5}$$

where p is the proportion of downtime (including machine general check, maintenance, etc.), η is the allowance (including shift time, stretch time, etc.), μ are shifts per day, and $\mu\in\{1,2,3\}$.

Based on formula (3), (4) and (5),the objective function of capacity is expressed in formula (6).

$$M_d = \frac{S\times(1\text{-}p)\times 8\mu\times(1\text{-}\eta)}{\frac{1}{k}\sum_{i=1}^{k} t_i\cdot\theta_i} \tag{6}$$

$$S.T.\begin{cases}\mu\in\{1,2,3\}\\ 0<\mathrm{p}<1\\ 0<\eta<1\\ 0<\theta_i\le 1, k>0\end{cases}$$

2.7 Cloud simulation and bottleneck detection

Cloud computing refers to the technology that makes cloud work. It lets you ‘plug into' infrastructure via the internet, and use computing resources without installing and maintaining them on-premises [15]. For this reason, the production data, capacity function and simulation model can be stored in the cloud saver to detect the bottleneck units.

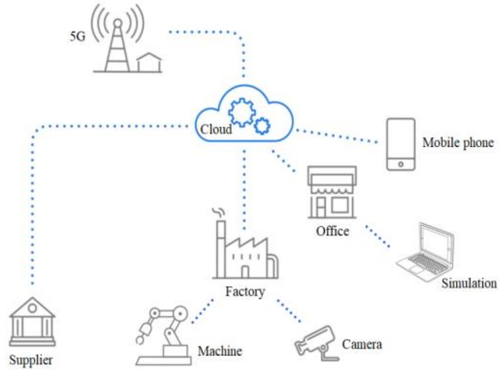

Fig.5. Operation process of cloud simulation.

3 Experiments

3.1 Case description

Z plant is the third power transformer production site, which is established in China by a Fortune 500 company. Its operation data is saved in the group cloud server, and can be shared globally. As Z plant has received some orders from the State Grid, its actual demand increased from 12800MVA to 15360MVA per year. As a result, it is urgent to detect the bottleneck in order to improve the production capacity.

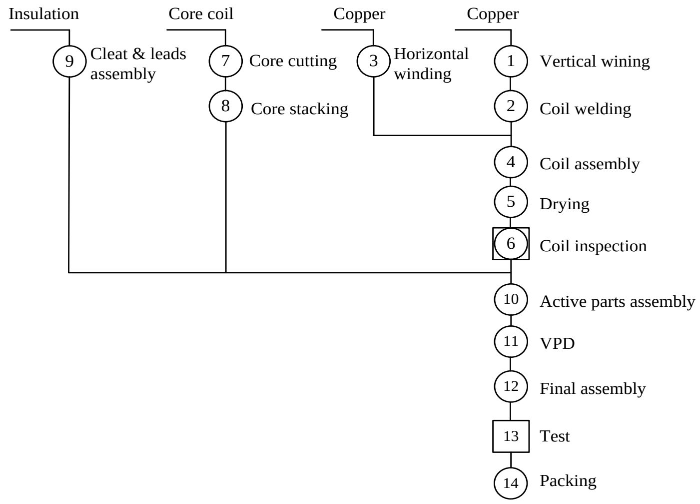

Fig.6. Process flow of transformer.

Firstly, the process flow of transformer is drawn as Fig.6 shows:

Secondly, the man-hour quota can be calculated by cloud computing and time study.

Table 1. Man-hour Quota of each process.

No.	Process name	Machine (Sets)	Quota (Hours/sets)	Remark
1	Vertical winding	6	255.59	
2	Coil welding	1	31.95	
3	Horizontal winding	3	95.85	
4	Coil assembly	—	—	No machine
5	Drying	1	54.90	
6	Coil inspection	—	—	No machine
7	Core cutting	2	56.80	
8	Core stacking	2	59.64	
9	Cleat & leads assembly	1	28.40	
10	Active parts assembly	3	117.86	
11	VPD	1	26.51	
12	Final assembly	1	32.19	
13	Test	1	25.56	
14	Packing	—	—	No machine

3.2 Definition of transformer production with OOCPN

According to the definition of OOCPN, the transformer manufacturing system is expressed as shown below:

$$OOCPN\text{-}T = \left\{ P_i, T_j; IM_i, OM_i, I_i, O_i, C_i \right\} \tag{7}$$

where P_i is a place set, $i \in \{1, 2, \cdots, 15\}$; T_j is a transition set, $j \in \{1, 2, \cdots, 12\}$; IM_i is a message input place set; OM_i is a message output place set; I_i is the input mapping function from P to T; O_i is the output mapping function from P to T; C_i is a colored set of places.

3.3 Abstract model of transformer production with OOCPN

By using IDMOB method, 14 processes of transformer manufacturing can be abstracted with $P_1 \sim P_{14}$, and a buffer P_{15} is established to deal with the production scheduling of oven. Then, according to the real process of transformer, the abstract model of transformer manufacturing system is constructed with the 15 modules.

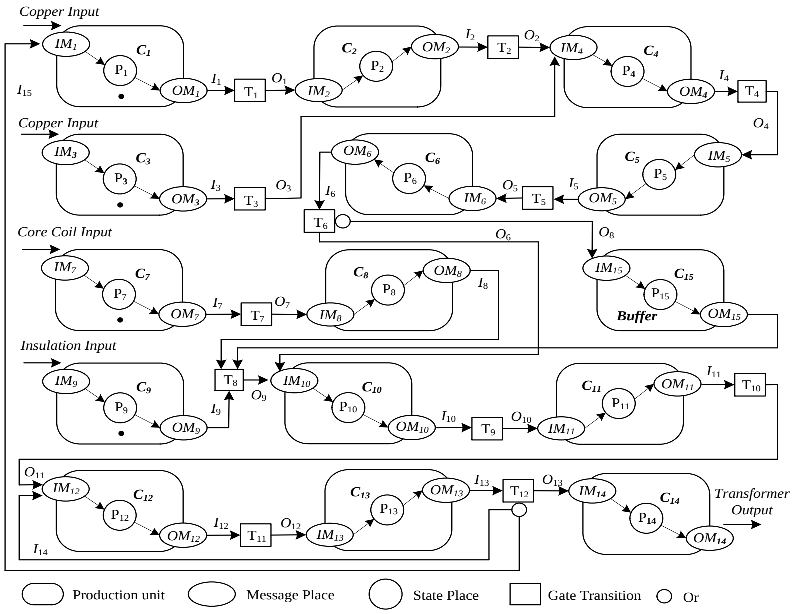

Fig.7. Abstract model of transformer production with OOCPN.

Next, meanings of places and transitions are described as Table 2 shows.

3.4 Performance evaluation for abstract model of transformer

3.4.1 Deadlock analysis for abstract model of transformer

In order to facilitate analysis and research, the incidence matrix of the abstract model can be stated as Table 3 shows.

Here, transition T_1 of winding process is taken as an example, the incidence matrix between place and transition is expressed as follows:

$$C = [c_{ij}]_{15\times 12} \tag{8}$$

where $c_{ij} = c_{ij}^{+} - c_{ij}^{-}$, $i \in \{1, 2, \cdots, 15\}$, $j \in \{1, 2, \cdots, 12\}$

$$c_{ij}^{+}=\begin{cases}1 & (t_j,p_i)\in F \quad i\in\{1,2,\cdots,15\},\ j\in\{1,2,\cdots,12\}\\ 0 & \text{otherwise}\end{cases}$$

$$c_{ij}^{-}=\begin{cases}1 & (p_i,t_j)\in F \quad i\in\{1,2,\cdots,15\},\ j\in\{1,2,\cdots,12\}\\ 0 & \text{otherwise}\end{cases}$$

Therefore, the incidence matrix of T_1 is described as follows:

$$C=\begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1\\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1\\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0
\end{bmatrix}$$

According to the formula (2), $x^T m = x^T m_0$ (x is the vector of $n\times 1$, m_0 is the initial marking, all m is reachable by m_0), and $m(P_i)$ are tokens of P_i .Consequently, $m(P_1)=1$, $m(P_1)+m(P_2)+\cdots m(P_{14})+m(IM_1)-m(OM_2)+\cdots+m(IM_{14})-m(OM_{14})=0(X)$. Therefore, T_1 can be fired.

Similarly, other transitions of abstract model can be fired under the initial marking m_0, so there is no deadlock in the abstract model of transformer production.

3.4.2 Conflict analysis for abstract model of transformer

As can be seen from Fig.7, transition T_{12} has more than one output places, and these places are connected to the transition with "or" relationship, so output conflicts will take place at T_{12}. In this case, the "fewest number of remaining operations" should be available to give priority to P_{12}, which should be sent back to the final assembly area to repair.

3.4.3 Conservation analysis for abstract model of transformer

Assuming $x^T\times C=0$, then by solving the equation, $x^T=\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 6 & 1 & 2 & 1 & 1 & 1 & 2\end{bmatrix}$. Therefore, there are non-zero positive vectors that make the equation $x^T\times C=0$ is true. As can be seen, the abstract model of transformer is strictly constrained without overflow.

3.5 Cloud Simulation and analysis for transformer production

3.5.1 Initial cloud simulation establishment

Hardware: 10th generation Intel i5-9500, 4GB (DDR4/2666), 1TB (SATA), integrated graphics card. Software: Plant simulation; Cloud server: Huawei cloud.

 https://doi.org/10.1051/itmconf/20224501015

Table 2. Meanings of places and transitions.

Colored set	Meanings	Places	Meanings	Transitions	Meanings
C_1	Vertical winding	P_1	Vertical winding	T_1	High/medium/low voltage coil winding is accomplished
C_2	Coil welding	P_2	Welding for high voltage coil	T_2	Welding for high voltage coil is finished
C_3	Horizontal winding	P_3	Horizontal winding	T_3	Medium voltage Ⅱ coil winding is accomplished
C_4	Coil assembly	P_4	Size adjustment	T_4	Get ready to dry with oven
C_5	Drying	P_5	Drying with oven	T_5	Coils are dried and prepare for assembly
C_6	Coil inspection	P_6	Coil inspection	T_6	Coils are checked and ready to be transported to the active parts assembly area
C_7	Core cutting	P_7	Core cutting	T_7	Core sheets are moved to stacking area by forklift
C_8	Core stacking	S	Core stacking	T_8	Preparations for active parts assembly are finished
C_9	Cleat & leads assembly	P_9	Cleat & leads assembly	T_9	Active part is ready to dry
C_{10}	Active parts assembly	P_{10}	Active parts assembly	T_{10}	Active part is out of Oven, and ready for final assembly
C_{11}	VPD	P_{11}	VPD	T_{11}	Final assembly is finished, prepare for test
C_{12}	Final assembly	P_{12}	Final assembly	T_{12}	If the transformer passes the test, it shall be packaged and delivered, otherwise, it shall be sent back to reassembly or rewinding
C_{13}	Test	P_{13}	Performance test		
C_{14}	Packing	P_{14}	Packing		
C_{15}	Buffer	P_{15}	Buffer for drying		

Table 3. Matrix for abstract model of transformer.

C	**T_1**	**T_2**	**T_3**	**T_4**	**T_5**	**T_6**	**T_7**	**T_8**	**T_9**	**T_{10}**	**T_{11}**	**T_{12}**	**m_0**
P_1	-1	0	0	0	0	0	0	0	0	0	0	-1	1
P_2	1	-1	0	0	0	0	0	0	0	0	0	0	0
P_3	0	1	-1	0	0	0	0	0	0	0	0	0	1
P_4	0	0	1	-1	0	0	0	0	0	0	0	0	0
P_5	0	0	0	1	-1	0	0	0	0	0	0	0	0
P_6	0	0	0	0	1	-1	0	0	0	0	0	0	0
P_7	0	0	0	0	0	0	-1	0	0	0	0	0	1
P_8	0	0	0	0	0	0	1	-1	0	0	0	0	0
P_9	0	0	0	0	0	0	0	-1	0	0	0	0	1
P_{10}	0	0	0	0	0	1	0	1	-1	0	0	0	0
P_{11}	0	0	0	0	0	0	0	0	1	-1	0	0	0
P_{12}	0	0	0	0	0	0	0	0	0	1	-1	-1	0
P_{13}	0	0	0	0	0	0	0	0	0	0	1	-1	0
P_{14}	-1	0	0	0	0	0	0	0	0	0	0	1	0
P_{15}	0	0	0	0	0	1	0	-1	0	0	0	0	1

By using the cloud simulation tools, the initial cloud simulation is designed to simulate the transformer manufacturing system.

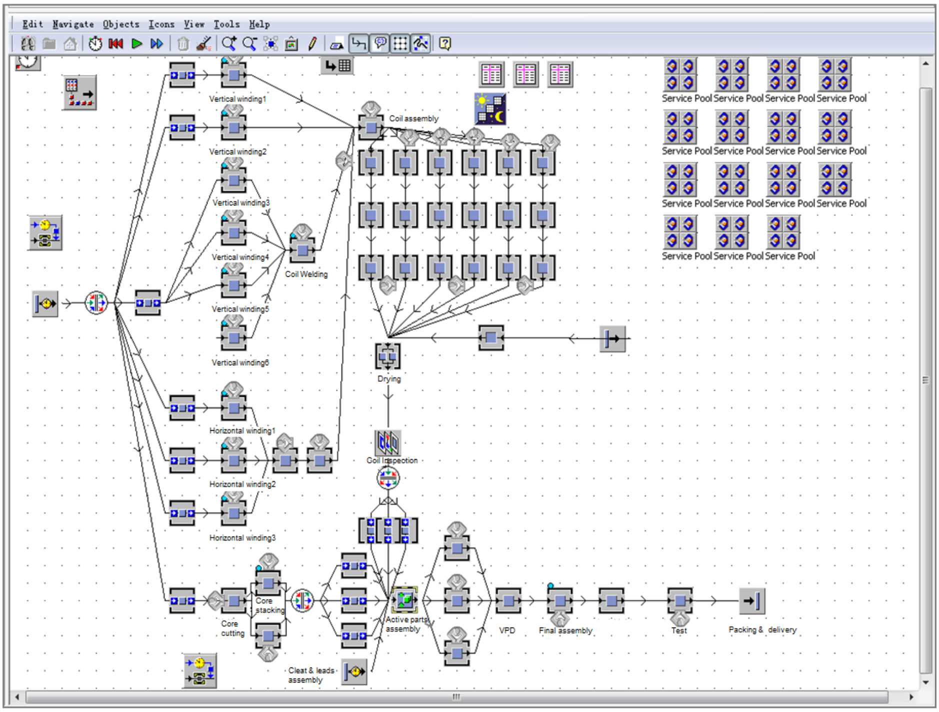

Fig. 8. Initial cloud simulation of transformer production system.

Then, the parameters of cloud simulation should be modified with cloud big data and practical time study, such as 2 shifts per day, 8 hours per shift, and 13.8% downtime.

3.5.2 Cloud simulation running and result output

All actual production data are plugged into the cloud simulation model to get the capacity of each processing unit by cloud computing and simulation running.

Table 4. Capacity of each process.

No.	Process name	Capacity1（sets/d）	Capacity2（sets/a）	Capacity3（MVA /a）
1	Vertical winding	0.2752	68.80	16512
2	Coil welding	0.3669	91.73	22015
3	Horizontal winding	0.3669	91.73	22015
4	Coil assembly	—	—	—
5	**Drying**	**0.2135**	**53.38**	**12811**
6	Coil inspection	—	—	—
7	Core cutting	0.4128	103.20	24768
8	Core stacking	0.3932	98.29	23590
9	Cleat & leads assembly	0.4128	103.20	24768
10	Active parts assembly	0.2984	74.60	17904
11	VPD	0.4423	110.57	26537
12	Final assembly	0.3642	91.06	21854
13	Test	0.4587	114.67	27521
14	Delivery	—	—	—

From Table 4, it can be seen that the coil drying process is the bottleneck unit of the system. To validate the result, the coil drying process is calculated with formula (6):

$$M_d(P_5)=\frac{1\times(1\text{-}13.8\%)\times 8\times 2\times(1-15\%)}{54.90}=0.2135\ sets/d$$

$$M_y(P_5)=0.2135\times 250\times 240=12811\ MVA/\mathrm{a}$$

This suggests that the cloud simulation running result is available.

3.5.3 Cloud simulation Improvement

According to the actual working experience, it is suggested that the coil drying process should be 3 shifts per day, and the cloud simulation need to be improved as Fig. 9 shows:

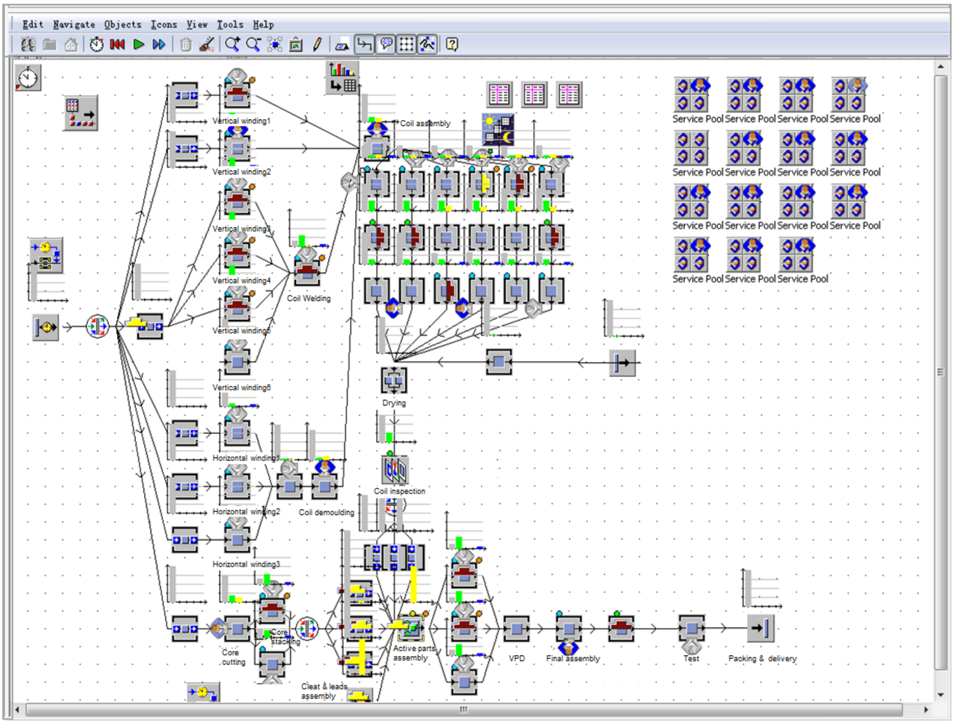

Fig. 9. Improved cloud simulation of transformer production system.

Through running the improved cloud simulation, it can be seen that the capacity of drying process has been improved, and the bottleneck shifts from drying process to vertical winding. However, the capacity of vertical winding process is 16512MVA per year, which is more than the orders.

Table 5. Evaluation of improvement.

No.	Items	Before	After	Effect
1	Production capacity（MVA /a）	12811	16512	28.89%
2	Increased profits（Million CNY/a）	—	—	17.35

4 Conclusion

In this paper, an intelligent detection method of bottleneck based on object-oriented colored Petri nets and cloud simulation is proposed to improve throughput for discrete manufacturing system. Firstly, the processing units are abstracted as Petri net modules. Then, the cloud simulation is created to simulate the actual production system. Moreover, the practical data and capacity function are plugged into the cloud to detect the bottleneck unit intelligently. Finally, the effectiveness and availability of the IDMOB method are validated by the empirical research. In the future, we are going to add the artificial neural network algorithm into the simulation model to improve the intelligence of detection.

This work is funded by the education and teaching reform project of Beijing Institute of Technology, Zhuhai (No. 2020033JXGG) and Key Disciplines of Guangdong Province: Mechanical Engineering.

References

1. ZHU Wenbohao, LI Xiaofeng, SUN Bo. A Study on the Upgrade of Traditional Industries through Digitalization in the Guangdong-Hong Kong-Macao Greater Bay Area in the Post-COVID-19 Pandemic Era [J]. *Intertrade*, 2021, (3): 52-59.
2. Ray Y. Zhong, Xun Xu, Eberhard Klotz, Stephen T. Newman. Intelligent Manufacturing in the Context of Industry 4.0: A Review [J]. *Engineering*, 2017, 3(5): 616-630.
3. Christopher Prinz, Niklas Kreggenfeld, Bernd Kuhlenkötter. Lean meets Industrie 4.0 – a practical approach to interlink the method world and cyber-physical world [J]. *Procedia Manufacturing*, 2018 (23):21-26.
4. GUAN Da, SHI Lei, KONG Liang, et al. A study on the carry capacity calculation and bottleneck identification of Baotou-Shenhua railway [J]. *Railway Transport and Economy*. 2020, 42(4), 42-47.
5. Mukund Subramaniyan, Anders Skoogh, Hans Salomonsson, et al. A data-driven algorithm to predict throughput bottlenecks in a production system based on active periods of the machines [J]. *Computers & Industrial Engineering*, 2018(125): 533-544.
6. Michael Wedel, Michael von Hacht, Ralf Hieber, et al. Real-time bottleneck detection and prediction to prioritize fault repair in interlinked production lines[J]. *Procedia CIRP*, 2015(37):140-145.
7. Binbin Huang, Wenbo Wang, Shan Ren, et al. A proactive task dispatching method based on future bottleneck prediction for the smart factory[J]. *International Journal of Computer Integrated Manufacturing*, 2019, 32(3):278-293.
8. YANG Haolong, GUO Yu, FANG Weiguang, CUI Shiting. Bottleneck Prediction for Discrete Manufacturing Workshops in Real-time Location Environment[J]. *Modular Machine Tool & Automatic Manufacturing Technique*, 2020, (10): 176-180.
9. WANG Shaowei, JIN Wenxiang. Research of Heuristic Rules Based on Dynamic Bottleneck of Production Process Optimization [J]. *Computer and Digital Engineering*, 2018, 46(9): 1927-1930.
10. J Bhagwan, S Kumar. Independent Task Scheduling in Cloud Computing using Meta-Heuristic HC-CSO Algorithm[J]. *Indian Journal of Science and Technology*, 2020, 13(2):114-126.

11. WANG Jinghua, HAN Jianghong , LIU Zhenyu, et al. Scheduling Modeling in Discrete Manufacturing System Based on Oriented Petri Net [J]. *Journal of System Simulation*, 2008, 20(15): 4159-4162.
12. Saurabh Shanua, Hanumat G.Sastry, Venkatadri Marriboyina. Optimal solution approach on large scale data to avoid deadlocks in resource allocations [J]. *Materials Today Proceedings*, 2021, 47(19): 7162-7166.
13. Liu C M, Wu F C. Using Petri nets to solve FMS problems[J]. *International Journal Computer Integrated Manufacturing*, 1993, 6 (3):175-185.
14. WANG Liya, CHEN Youling, MA Hanwu, et al. Production planning and control[M]. Beijing: Tsinghua University Press, 2007: 135-139.
15. Sai Vennam. Hybrid cloud explained [EB/OL]. https://www.ibm.com/cloud/learn/loud-computing, 18 August 2020.