Multi-task service composition model considering competitive resource constraint in cloud manufacturing

Wang Zhongning, Wang Yankai*, Chen Ronghua, Huang Hui, Luo Bin
College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing 400044, China.
*Corresponding author’s e-mail: 20112996@cqu.edu.cn.

Abstract. Service composition optimization is one of the core issues of cloud manufacturing research. At present, relevant researches mostly focus on single-task service composition, and there are few reports on the problem of multi-task service composition considering resource competition. However, in the real cloud manufacturing environment, multi-task service composition is common, so how to formulate a robust multi-task service composition model is a meaningful research direction. This paper first proposes a multi-task service composition optimization model considering resource competition constraints (MTSCOM-RCC) in the cloud manufacturing environment to fill the above-mentioned gap. The MTSCOM-RCC model considers the conflicts of interest and time between parallel subtasks and proposes an important service evaluation indicator. Secondly, a two-layer coding method, including task layer and sub-task, is proposed based on studying the MTSCOM-RCC’s competition mechanism and characteristics. An improved hybrid genetic artificial bee colony algorithm model is proposed to solve the multi-task model. The model is combined with the global optimization method and task queue optimization method. Experimental results prove that the two-layer task optimization method model is more feasible and effective than the other two solution models.

1. Introduction

Cloud manufacturing (CMfg) [1] is a new service-oriented model, high-efficiency and low-consumption knowledge networked intelligent manufacturing. It provides high-quality, safe, and reliable services for the entire life cycle through the network, improve resource utilization, and reduces manufacturing costs [2]. Cloud service composition and optimization (CSCO) is one of the core problems of CMfg research. It has the characteristics of multiple optimization goals, large solution space, complexity and dynamic solution process, etc. It is a typical NP-hard problem.

At present, the single-task service composition optimization problem has been extensively studied [3-7]. A few domestic and foreign scholars have researched multi-task problems [8-10]. At present, the research method of cloud manufacturing multi-task optimization problem still adopts the mode of selecting and optimizing sub-tasks in the candidate cloud service resources, similar to the single-task research method. In contrast, current methods ignore the limited resources or urgent needs in multi-task service composition optimization. The multi-task service composition optimization problem considering resource competition constraints among multi-task in CMfg is a practical problem that needs to be solved urgently.

This paper proposes a multi-task service composition optimization model considering resource competition constraints (MTSCOM-RCC) in response to the above problems. This model aims to solve the conflict problem when multiple tasks compete for single or multiple cloud services simultaneously.
by combining the scheduling method. In the model, a two-layer coding method including task layer and sub-task layer is proposed. Simultaneously, a hybrid genetic artificial bee colony algorithm that acts on both the task layer and the sub-task layer is proposed. Besides, to solve MTSCOM-RCC efficiently, this paper proposes a two-layer task optimization method by first optimizing the task layer’s solution, searching the sub-task layer locally, and finally realizing the optimization solution of the model.

2. Multi-task service composition optimization model considering resource competition constraints

2.1. Problem statement
Cloud manufacturing has the characteristics and advantages of the centralized use of decentralized resources and the decentralized services of centralized resources. In the actual operation and management process, there is a problem of competition caused by multiple tasks requiring a certain service at the same time. How to reasonably allocate competitive services to multiple cloud manufacturing tasks and obtain the optimal manufacturing process is the key point of MTSCOM-RCC. This paper will focus on how to model and solve the problem.

Based on the traditional cloud manufacturing service composition optimization process, the multi-task service composition optimization model considering resource competition constraints (MTSCOM-RCC) is mainly composed of three steps: task decomposition, service and task matching, and service combination selection and optimization [11]. The optimization process of MTSCOM-RCC is shown in Fig. 1. As we can see from Fig.1, multiple subtasks may require the same cloud manufacturing resource in the service matching process.

![Figure 1 MTSCOM-RCC optimization process](image-url)
2.2. Mathematical model of MTSCOM-RCC

2.2.1. Objective functions. In the MTSCOM-RCC model, the cloud manufacturing task set MTS represents the total manufacturing tasks, which is composed of multiple single manufacturing task sets:

\[ MTS = \{ MT_1, MT_2, \ldots, MT_n, \ldots, MT_{Si} \} \]  

where \( i = 1, 2, \ldots, l \), where \( l \) is the number of tasks. \( MT_i \) is the \( i \)-th task, and each task is decomposed into a series of subtasks \( ST \), which can be expressed as

\[ MTS_{Si} = \{ S_{i1}, S_{i2}, \ldots, S_{ij}, \ldots, S_{in} \} \]  

Among them, \( ST_{ij} \) represents the \( j \)-th subtask into which task \( i \) is decomposed; \( j = 1, 2, \ldots, n \), where \( n \) is the number of subtasks that task \( MT_i \) is decomposed into. To more efficiently select the service combination that meets the task requirements, this paper uses QoS indicator [7], namely completion time (T), product quality (Q), total cost (C), and service level (S), to establish a multi-task competitive resource optimization evaluation index for MTSCOM-RCC.

1) Completion time (T): This indicator includes logistics time \( T_l \) and manufacturing time \( T_e \), waiting time \( T_w \), and processing time \( T_p \). The time required to complete the manufacturing process of a subtask \( ST_{ij} \) can be expressed as:

\[ T = T_l + T_e = l(x_{ij}, x_{ij-1}) + (T_{wj} + T_{mij}) \]  

where \( l(x_{ij}, x_{ij-1}) \) represents the logistics time from the service \( x_{ij} \) for the subtask \( ST_{ij} \) to the service \( x_{ij-1} \) for \( ST_{ij-1} \). \( T_{wj} \) and \( T_{mij} \) respectively represent the waiting time, preparation time, and processing time in the manufacturing process of the subtask \( ST_{ij} \). 

2) Total cost (C): This indicator includes logistics cost \( C_l \) and processing cost \( C_e \). The total cost required for the manufacturing process of a subtask \( ST_{ij} \) can be expressed as:

\[ C = C_l + C_e = l(x_{ij}, x_{ij-1}) + C_{mij} \]  

where \( l(x_{ij}, x_{ij-1}) \) represents the logistics cost from the service \( x_{ij} \) for the subtask \( ST_{ij} \) to the service \( x_{ij-1} \) for \( ST_{ij-1} \). \( C_{mij} \) represents the processing cost of the subtask \( ST_{ij} \) manufacturing process.

3) Product quality (Q): This indicator represents the product qualification rate for completing the task. It is provided by the cloud service provider and can be expressed as:

\[ Q = q(x_{ij}) \]  

4) Service level (S): This indicator means that the cloud service demander scores the cloud service provider's service level after its demand is completed. The data is summarized and provided by the cloud platform operator and can be expressed as:

\[ S = s(x_{ij}) \]

2.2.2. Objective functions. In order to calculate the comprehensive value of each cloud service, each indicator needs to be quantified. The evaluation index is divided into the positive index and negative index. The positivity index means that the larger the index value, the better the comprehensive evaluation. The negative index means that the smaller the value of this index, the better the comprehensive evaluation. Product quality and service level are positive indicators, and completion time and total cost are negative indicators. Considering the generality of tasks, the cloud manufacturing tasks in the MTSCOM-RCC problem in this paper are all sequential structures. The objective function of the MTSCOM-RCC model is:

\[ R = w_T \times (1 - \frac{T_{l}}{T_{max}}) + w_Q \times \frac{Q}{Q_{max}} + w_C \times (1 - \frac{C}{C_{max}}) + w_S \times \frac{S}{S_{max}} \]  

Among them, \( w_T \), \( w_Q \), \( w_C \) and \( w_S \) are the weighting coefficients of \( T \), \( Q \), \( C \) and \( S \) respectively, and the sum of the four weighting coefficients is 1. This paper adopts the weight coefficient \( w_T = 0.1791; w_Q = 0.4647; w_C = 0.2929; w_S = 0.0634 \) in the literature [5] as a reference.
3. Algorithm solution and optimization method

3.1. Original Artificial Bee Colony Algorithm

The artificial bee colony algorithm [3] is a typical population-based intelligent algorithm that simulates a natural bee colony’s foraging and communication mechanism. The colony is divided into employed bees, following bees, and scout bees. In the MTSCOM-RCC problem, each nectar source represents a candidate solution, and the quality of the nectar source represents the solution’s fitness. The algorithm process is as follows. The first is the stage of employed bees, where bees fly randomly to different places to find nectar sources $X = \{x_{i1}, x_{i2}, ..., x_{in}\}$, the search formula is defined as:

$$x_{ij} = x_{ij}^{\text{min}} + r(x_{ij}^{\text{max}} - x_{ij}^{\text{min}})$$  \hspace{1cm} (8)

where $i = 1, 2, ..., N_D$, $N_D$ is the number of nectar sources (employed bees), $j = 1, 2, ..., n$, $n$ is the individual dimensions of the population; $r$ is a random number from 0 to 1; $X_{ij}^{\text{max}}$ and $X_{ij}^{\text{min}}$ is the upper and lower bounds of the value of the $j$-th element. Then in the following bee stage, the probability expression for the following bee to select a food source is:

$$p_i = f_i / \sum_{k=1}^{N} f_k$$  \hspace{1cm} (9)

where $f_i$ is the fitness of the $i$-th food source, and $p_i$ is the probability of the $i$-th food source being selected by the bee. After choosing a new food source as the direction, following bee's search to generate a new nectar source $V_i$:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$  \hspace{1cm} (10)

where $i$ is the previous food source, $k$ is the selected food source, and $\phi_{ij}$ is a random value from -1 to 1. The employed bees also search for areas near the nectar source according to equation (10). Each search will update the nectar location through greedy selection. If a nectar source is not optimized after a certain number of iterations limit, the scout bee discards the nectar source and uses the formula (8) to find a new food source.

3.2. Proposed hybrid genetic artificial bee colony algorithm

3.2.1. Coding method.

The presented MTSCOM-RCC is a typical discrete problem. In order to solve this problem, a two-layer coding method based on tasks and services is proposed to realize the MTSCOM-RCC coding $X_M = (X_{\text{task}}, X_{\text{service}})$. The mathematical expression is as follows:

$$X_M = (X_{\text{task}}, X_{\text{service}}) = (x_1, x_2, ..., x_{N}; x_{N+1}, x_{N+2}, ..., x_{2N})$$  \hspace{1cm} (11)

Where $X_M$ is the encoding result, $X_{\text{task}}$ is the task code, $X_{\text{service}}$ is the service code. $N$ is the total number of subtasks. $X_{\text{task}}$ represents each element’s sequence in corresponds to the manufacturing sequence between subtasks; each element in $X_{\text{service}}$ represents the corresponding subtask selected service. As shown in Table 1, this example has 3 tasks ($MT_1, MT_2, MT_3$), each task has 2 subtasks ($ST_{11}, ST_{12}, ST_{21}, ST_{22}, ST_{31}, ST_{32}$), a total of 6 subtasks, namely $N=6$. The corresponding code is $X_M = (1,1,2,3,2,3; 1,2,3,3,1,3)$.

| ST_{11} | ST_{12} | ST_{21} | ST_{22} | ST_{31} | ST_{32} |
|---------|---------|---------|---------|---------|---------|
| 1       | 1       | 2       | 3       | 2       | 3       |
| 1       | 2       | 3       | 3       | 1       | 3       |

Figure 2 A coding example for MTSCOM-RCC

3.2.2. Evolution pattern.

The artificial bee colony algorithm is used to solve continuous optimization problems. In contrast, the MTSCOM-RCC problem is a typical discrete problem, and the optimization Eq. (8-10) of the original artificial bee colony algorithm cannot be used directly. Aiming at the characteristics of the conflicts in
the MTSCOM-RCC problem subtasks, combining the coding method and the genetic algorithm's mutation method, a hybrid genetic artificial bee colony algorithm is proposed. The example code is used as an explanation. Due to the different constraints of task coding and service coding, the total number of each element in $X_{task}$ will not change during the solution process of task coding; each element in $X_{service}$ during the solution process of service coding is within a reasonable range of the corresponding optional candidate resource set.

For task coding, four evolutionary methods are proposed: constant single point, two-point crossover, single-point insertion, and a segment translation.

1) "constant single point" means that the positions of all subtasks of one task in the two task coding remain unchanged. The remaining tasks the order of the subtask positions is arranged by order of each other's task code.

2) "two-point crossover" refers to the code exchange of two random positions in a task code.

3) "single-point insertion" means that a certain code in a task code is randomly inserted into the code another position in the code;

4) "a segment translation" means that a certain code segment in a task code is randomly moved to another position in the code.

The diagrams of the four evolution patterns are as follows:

![Evolution patterns for task code: a) constant single point b) two point crossover c) single-point insertion d) a segment translation](image)

For service coding, three evolutionary methods are proposed: population crossover, one segment translation, and single point mutation. "Population crossover" means that a certain segment of two service codes crosses into the other's code. "One segment of translation" refers to a certain segment of one service code. The code is randomly inserted into another position in the code. "Single point mutation" means that the code at a certain point in the service code becomes another reasonable coding situation. The diagrams of the three evolution patterns are as follows:

![Evolution patterns for service code: a) population crossover b) one segment translation c) single point mutation](image)
3.3. Two-level task optimization method for MTSCOM-RCC

The global optimization method [12] is the most commonly used in solving scheduling problems. The algorithm is directly applied to the two-layer coding, which requires high quality of the initial solution; the task queue optimization method uses the "first in first out" queue rule, which only applies to the task code. It traverses all optional services for each task being scheduled and selects the service with the highest current total service quality.

The global optimization method and the task queue optimization method use the same algorithm to solve the problem. The difference is that the global optimization method uses a two-layer coding method, and the task queue optimization way uses a single-layer coding way. The two-layer task optimization method combines the advantages of the global subtask optimization method and the task queue optimization method: first process the task queue, and after the task queue is solved, the task code obtained is expanded into a two-layer code including task code and service code, and then perform a partial search for service coding optimization.

The solution steps of the two-layer task optimization method are as follows.

a) Initialize algorithm parameters. According to formula (8), the task layer code $X_{\text{task}}$ is randomly generated.

Employed bee stage:

b) For each $X_{\text{task}}$, the corresponding service layer code $X_{\text{service}}$ is generated according to the "first in, first out" queue rule. Combine the task layer’s coding and the service layer to $X_M = (X_{\text{task}}; X_{\text{service}})$.

c) The domain search is performed, and according to formula (9) and (10), the population crossover method of population is used to optimize the service layer coding.

d) Use a segment of translation and single point mutation evolution patterns to optimize the service layer coding, the execution probability is $p_{t1}$ and $p_{t2}$ respectively.

Following bee stage:

e) According to formula (9), the single-point constant evolution pattern is used to optimize the task-level coding.

f) Optimize the task layer coding according to the three evolutionary modes of two-point crossover, single-point insertion, and a segment translation, and the execution probabilities are $p_{s1}$, $p_{s2}$, and $p_{s3}$, respectively.

Scouting bee stage:

h) Check whether each population is still not optimized after limit updates. If yes, discard the current population, and generate a new population according to formula (8).

i) Repeat steps b)~g) until the maximum number of iterations is reached, then go to step i).

4. Experimental results and analysis

4.1. Experimental data

In order to verify the effectiveness of the MTSCOM-RCC model, experiments of different scales are designed. The number of tasks, the number of subtasks, and the number of services have three scales, including 3, 6, and 9. There are a total of 27 kinds of tests. Each experiment is named in the form of task-subtask-service. For example, test 333 represents an experiment with 3 tasks, 3 subtasks, and 3 optional services. Taking the test 333 example as an example, Table 1 shows the processing time and preparation of each subtask corresponding service. Table 2 shows the processing cost and product quality of each subtask corresponding service, where NA means that this service is not optional. Table 3 shows the logistics time, logistics cost, and service quality between services.

| Task | ST11 | ST12 | ST13 | ST21 | ST22 | ST23 | ST31 | ST32 | ST33 |
|------|------|------|------|------|------|------|------|------|------|
| S1   | 4/2  | NA   | 3/1  | 4/2  | 2/1  | NA   | 2/1  | 4/2  | NA   |
| S2   | 2/1  | 3/2  | NA   | NA   | 2/1  | 4/1  | 3/1  | 4/1  | 4/2  |
| S3   | NA   | 2/1  | 2/1  | 2/1  | 3/1  | 4/2  | 2/1  | NA   | 3/2  |
Table. 2  \( C_e \) (Kilo yuan)/\( q \)

|     | \( ST_{11} \) | \( ST_{12} \) | \( ST_{13} \) | \( ST_{21} \) | \( ST_{22} \) | \( ST_{23} \) | \( ST_{31} \) | \( ST_{32} \) | \( ST_{33} \) |
|-----|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| \( S1 \) | 267/9.9       | NA            | 221/9.2       | 462/9.2       | 144/9.4       | NA            | 136/9.5       | 130/9.3       | NA            |
| \( S2 \) | 278/9.3       | 120/9.5       | NA            | NA            | 117/1         | 405/9.4       | 115/10        | 175/9.8       | 421/9.5       |
| \( S3 \) | NA            | 155/9.6       | 261/9.7       | 448/9.8       | 147/9.3       | 428/9.9       | 136/9.4       | NA            | 422/10        |

Table. 3  \( T_i \) (Day)/ \( C_i \) (Kilo yuan)/\( s \)

|     | \( S1 \) | \( S2 \) | \( S3 \) |
|-----|---------------|---------------|---------------|
| \( S1 \) | 0/0           | 2/2.5         | 2/1.5         |
| \( S2 \) | 2/2.5         | 0/0           | 3/2.75        |
| \( S3 \) | 2/1.5         | 3/2.75        | 0/0           |
| \( s \)  | 0.8687        | 0.8586        | 1             |

Table. 4  The range of parameters of MTSCOM-RCC

| Parameter | \( T_e \) | \( T_i \) | \( C_e \) | \( q \) | \( T_i \) | \( C_i \) | \( s \) |
|-----------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| Range     | \{2,3,4\}     | \{1,2\}       | [100,500]     | [8,10]       | \{1,2,3\}     | [1,3]         | [0.8,1]       |

The parameter ranges of all experiments are shown in Table 4. Besides, the population size is 20, the maximum number of iterations is 300, the coding parameters \( pt_1 \), \( pt_2 \), \( ps_1 \), \( ps_2 \) and \( ps_3 \) of the two-layer task optimization algorithm are 0.5, 0.8, 0.7, 0.4, 0.4, respectively, and the bee colony optimization parameter limit is 40. The test environment uses Matlab R2014a simulation software, intel(R) Core i5-4460, 3.2GHz processor, and 8.00GB of memory. Each experiment carried out 20 groups.

4.2. Result analysis
To evaluate the superiority of the two-layer task optimization method, it is compared with the solution results of the global optimization method and the task queue method. The comparison between the average value of the total service quality solution and the solution time is shown in Table 5.

Table. 5  Results obtained by three methods

| Experiments | Two-layer task optimization method | Global optimization method | Queue task method |
|-------------|-----------------------------------|----------------------------|------------------|
|             | Mean | Time | Mean | Time | Mean | Time |
| 333         | 0.6019 | 5.18 | 0.6019 | 0.55 | 0.6006 | 2.22 |
| 336         | 0.6278 | 5.84 | 0.6207 | 0.55 | 0.6275 | 3.52 |
| 339         | 0.6259 | 6.46 | 0.6091 | 0.56 | 0.6254 | 4.91 |
| 363         | 0.5417 | 9.58 | 0.5088 | 0.99 | 0.5396 | 4.50 |
| 366         | 0.6679 | 10.86 | 0.6167 | 1.00 | 0.6679 | 7.01 |
| 369         | 0.6822 | 12.13 | 0.5852 | 1.01 | 0.6806 | 9.87 |
| 393         | 0.5519 | 13.94 | 0.5053 | 1.43 | 0.5511 | 6.69 |
| 396         | 0.6190 | 15.81 | 0.5307 | 1.45 | 0.6190 | 10.53 |
| 399         | 0.6402 | 17.82 | 0.5247 | 1.46 | 0.6401 | 14.77 |
| 633         | 0.6358 | 9.53 | 0.6114 | 0.98 | 0.6306 | 4.47 |
The result of solving the MTSCOM-RCC model is to consider the comprehensive quality of service (QoS) result, including completion time, product quality, total cost, and service level, not the time-optimal result. For scheduling problems, the Gantt chart can intuitively express the scheduling results. Taking test 333 as an example, the three methods' scheduling results are shown in Fig. 5, and the overall results of the solution are shown in Table 6.

![Gantt chart](image)

**a) Global optimization method scheduling results**
b) Task queue method scheduling results

c) Scheduling results of two-level task optimization method

Figure 5 Three methods scheduling results

| Method                        | Optimal solution coding | QoS   | Completion time(day) |
|-------------------------------|-------------------------|-------|----------------------|
| Two-layer task optimization method | 212332113; 133332223 | 0.6019 | 19                   |
| Task queue method             | 213232311               | 0.6006 | 19                   |
| Global optimization method    | 312231123; 133332223    | 0.6019 | 19                   |
Based on the above experimental results, it can be seen that the global optimization method can guarantee the optimization direction of the solution. However, the ability to obtain is poor when the number of tasks and services are large. The task queuing method has a better solving ability. Due to the phased steps' traversal to determine the service, a better time solution can be obtained. However, it does not solve the multi-indicator coupling complexity of the MCSO model, which is not universal. Besides, in many test cases of fewer services and more competition, it is easy to fall into the local optimum, and the solution result is unstable. For example, in test 333, the task queue method takes the same time to complete the solution as the other two methods, where both are 19 days. However, due to the determined steps' traversal, service 1 with poor service level is selected multiple times, resulting in the final low quality of service. The two-layer task optimization method can obtain the highest comprehensive service quality solution within an acceptable time, has good adaptability and effectiveness, and provides theoretical and technical support for the research and development of multi-task competitive resource scheduling problems.

5. Conclusion
The contribution of this paper is summarized as follows.
1) This paper puts forward the problem of multi-task competition resource optimization on the cloud platform, focusing on solving the situation of competition when multiple cloud manufacturing tasks require one or several services simultaneously.
2) The total service quality consisting of time, quality, cost, and service is taken as the optimization target, and a mathematical model of multi-task competitive resource optimization is established.
3) A service composition scheduling model is proposed. Moreover, this paper proposes a two-layer coding method and develops a hybrid genetic artificial bee colony algorithm. Based on the global optimization method and task queue method, a two-layer task optimization method is proposed.
4) Experimental results show that the two-layer task optimization method can effectively solve multi-task competing resource scheduling on the cloud platform. Twenty-seven experiments ranging from small to large demonstrate the effectiveness and feasibility of the model and algorithm in this paper.

Acknowledgments
The presented work was supported by the 2030 Innovation Megaprojects of China (Programme on New Generation Artificial Intelligence) [grant number 2018AAA0101804], Key Project of Technological Innovation and Application Development Plan of Chongqing (Grant No. cstc2019jscx-mbdxX0056).

References
[1] Li B H, Zhang L, WANG S L, et al. Cloud manufacturing: a new service-oriented networked manufacturing model [J]. Computer Integrated Manufacturing Systems, 2010, 16(1):1-7,16.
[2] Li B H, Zhang L, Ren L, et al. Further discussion on cloud manufacturing [J]. Computer Integrated Manufacturing Systems, 2011, 17(3):449-457.
[3] Zhou J J, Yao X F. Multi-objective hybrid artificial bee colony algorithm enhanced with Lévy flight and self-adaption for cloud manufacturing service composition [J]. Applied Intelligence, 2017, 47(3):721-742.
[4] Su K K, Xu WS, Li JY, et al. Manufacturing resource allocation method based on bi-level programming in cloud manufacturing [J]. Computer Integrated Manufacturing Systems, 2015, 21(7):1941-1952.
[5] Cao Y, Wang S L, Kang L, et al. A TQCS-based services election and scheduling strategy in cloud manufacturing [J]. International Journal of Advanced Manufacturing Technology, 2016, 82(1-4):235-251.
[6] Zhou J, Yao X. Hybrid teaching-learning-based optimization of correlation-aware service composition in cloud Manufacturing [J]. International Journal of Advanced Manufacturing Technology, 2017: 1-19.
[7] Yang Y F, Yang B, Wang S L, et al. An enhanced multi-objective grey wolf optimizer for service composition in cloud manufacturing [J]. Applied Soft Computing, 87.

[8] Liu WN, Liu B, Sun L H. Multi-task oriented service composition in cloud manufacturing [J]. Computer Integrated Manufacturing Systems, 2013, 19(1):199-209.

[9] Han Y L. Research on cloud manufacturing service combination optimization based on harmony search algorithm[D]. Kaifeng: Henan University, 2018.

[10] Shu X, Wang S L, Kang L, et al. Multi-task scheduling game with limited resources for cloud manufacturing[J]. Journal of Chongqing University, 2020(3).

[11] Tao F, Zhao D M, Hu Y F, et al. Resource service composition and its optimal-selection based on particle swarm optimization in manufacturing grid system[J]. Industrial Informatics IEEE Transactions on, 2008.

[12] Li Y L, Li X Y, Gao L. Review of Hybrid Flow Shop Scheduling Problems[J] Chinese Journal of Mechanical Engineering, 2020, 31(23):2798-2813+2828.