

SCHEDULING OPTIMIZATION OF CLOUD RESOURCE SUPPLY CHAIN THROUGH MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION

Jiang, H. & Liu, C. Y.[#]

International Business College, South China Normal University, Nanhai District, Foshan 510631, Guangdong Province, China

E-Mail: lcycaiyn0119@sina.com ([#] Corresponding author)

Abstract

The cloud manufacturing, a new service-oriented manufacturing mode, has become a hotspot in manufacturing industry. Considering supply chain scheduling optimization as a multi-objective optimization problem, this paper combines the fuzzy correlation entropy (FCE) and particle swarm optimization (PSO) into an FCE-based multi-objective optimization approach, in which the fitness of the FCE coefficient is taken as that of the multi-objective PSO to guide the algorithm evolution. After that, the discrete event simulation of the optimization process was carried out on FlexSim software. The simulation results show that the proposed algorithm outperforms the traditional supply chain scheduling algorithms in stability and effectiveness. The research findings shed new light on the scheduling optimization of cloud resource supply chain.

(Received, processed and accepted by the Chinese Representative Office.)

Key Words: Cloud Manufacturing, Supply Chain, Multi-Objective Particle Swarm Optimization (PSO), Fuzzy Correlation Entropy (FCE), Discrete Event Simulation

1. INTRODUCTION

Cloud manufacturing [1], extended from traditional distributed network manufacturing, refers to the service-oriented integration of networked manufacturing and cloud computing, the Internet of Things (IoT) and computer virtualization. There are three parties in each cloud manufacturing system, namely, resource providers, resource users and manufacturing cloud operators. Such a system operates in the following manner: the resource providers register on the cloud platform to perceive and virtualize resources, and aggregate them into cloud services; the resource users submit manufacturing task requests to the cloud platform, and rent high-quality services from the cloud platform on demand; the cloud platform automatically decomposes each task into several subtasks according to the rules of production process and the requests of resource users, and searches for the suitable manufacturing services that matches each subtask.

The key to the construction of cloud manufacturing system lies in the scheduling optimization of cloud resources, which are the basic components of the system [2-4]. For cloud resources, the scheduling optimization is similar to that of distributed network resources, except for one major difference. The scheduling optimization of distributed network resources is static in nature. Each resource-demanding request needs to be divided into many small parts, and assigned to lots of resource providers for further processing. The scheduling optimization of cloud resources, however, is highly dynamical, as the number and type of the resources are constantly changing during the scheduling process. To depict the dynamicity and fault-tolerance of cloud manufacturing, it is necessary to probe into the dynamic resource scheduling optimization.

2. LITERATURE REVIEW

The multi-layer architecture and modularization have often been adopted to build cloud manufacturing systems. Xu [5] presented a four-layer cloud manufacturing system, which consists of an application layer, a global service layer, a virtual service layer and a resource layer. Li et al. [6] developed a ten-layer cloud manufacturing architecture, including but not limited to a resource layer, a perception layer, a virtual resource layer and a core cloud service layer. Yan et al. [7] divided the cloud manufacturing service management and control platform into seven functional modules, namely, production management module, system management module, technology management module, financial management module, integrated information query module, production evaluation module, plan management module and inventory management module.

The scheduling optimization of cloud resources is the cornerstone of cloud manufacturing. The related model and solution mechanism directly bears on the accuracy and velocity of cloud resource sharing. Patel [8] proposed a three-objective scheduling optimization model for networked resources, and solved it by genetic algorithm based on objective weight. Laili et al. [9] applied the objective weight-based maximum inheritance method to the scheduling optimization of three-objective cloud resources in cloud manufacturing environment.

The huge amount of resources in cloud manufacturing system adds to the time and cost of scheduling optimization. Moreover, there may be multiple objectives of the optimization, such as product quality, credibility and reliability. In essence, the scheduling optimization of cloud resources is a high-dimensional multi-objective optimization problem. However, the existing models [10-12] are all incomplete open-loop models, failing to consider resource errors in scheduling optimization. None of them can fully illustrate the real-time dynamicity and fault-tolerance of cloud manufacturing.

From the angle of algorithm optimization, the scheduling optimization of resources can be abstracted as a multi-objective optimization problem. The existing multi-objective optimization algorithms falls into two categories by the fitness allocation mechanism: Pareto priority method and target weight selection. Neither type of algorithms can solve high-dimensional multi-objective problems well [13-15]. Blasco et al. [16] suggested the Pareto priority method be replaced by a new fitness evaluation and selection strategy or combined with a new technology, such as to deal with multi-objective optimization problems. In light of the above, this paper aims to find an effective method for multi-objective optimization of the scheduling of a cloud resource supply chain.

3. MODELLING FOR SCHEDULING OPTIMIZATION OF CLOUD RESOURCE SUPPLY CHAIN

Through cloud manufacturing, the idle resources distributed in various locations are integrated into high-quality cloud manufacturing services, fulfilling the requests from manufacturing enterprises. This paper explores the scheduling optimization of cloud resource supply chain with dynamic and complex resource requests and resource distribution, aiming to optimize the scheduling of cloud manufacturing tasks with resources from different locations.

The following hypotheses were put forward to obtain an accurate mathematical description of our problem: (1) Each requested task is automatically split into multiple serial subtasks by the cloud manufacturing platform according to certain process rules; (2) This paper only considers the scheduling optimization of hard resources (i.e. supply chain devices) distributed in various locations, rather than that of these resources in an enterprise with a

single supply chain of resources. The supply chain factors like logistics time, logistics cost and delay cost will be generated in the scheduling optimization process.

3.1 Primary model

The cloud manufacturing platform automatically decomposes the requested task into several subtasks by attribute information, and then searches for the resources matching each subtask from the cloud resource pool. After that, the searched resources are prepared for the optimization process. According to the demand of the resource requester, the scheduling optimization pursues the best product quality at the minimal processing time and production cost. To eliminate resources with long processing time, high production cost and poor product quality, this paper sets up a primary model based on the relative membership function [17] below:

$$u_M(i) = \begin{cases} 1 & f_M(i) \leq y_{M2} \\ \frac{f_M(i)-y_{M2}}{y_{M1}-y_{M2}} & y_{M1} < f_M(i) < y_{M2} \\ 0 & f_M(i) \geq y_{M2} \end{cases} \quad (1)$$

where, $u_M(i)$ is the membership value of the M^{th} index; y_{M1} and y_{M2} are the upper and lower bounds of the M^{th} index, respectively. The following matrix can be derived through the search of the v^{th} resource matching the i^{th} task.

$$\left[\frac{p(iv)-p(i)_{min}}{p(i)_{max}-p(i)_{min}}, \frac{q(iv)-q(i)_{min}}{q(i)_{max}-q(i)_{min}}, \frac{Qa(iv)-Qa1(i)_{min}}{1-Qa1(i)_{min}} \right] \quad (2)$$

where the first dimension of the matrix is the processing time; the second dimension is the production cost; the third dimension is the product quality. When the membership value of each dimension falls in the range of [0, 1], the resources can be retained into the preferred candidate resource set. In addition, $p(iv)$, $q(iv)$ and $Qa(iv)$ denote the processing time, production cost and product quality of the v^{th} resource matching the i^{th} sub-task. For processing time, production cost and product quality, the upper bounds are $p(i)_{max}$, $q(i)_{max}$ and 1, respectively, and the lower bounds are $p(i)_{min}$, $q(i)_{min}$ and $Qa1(i)_{min}$, respectively.

By Eq. (1), the three basic indices (i.e. processing time, production cost and product quality) of resources can be expressed as real numbers between [0, 1]. In this way, the primary model can effectively eliminate the resources with long processing time, high production cost and poor product quality.

3.2 Optimization model

For our problem (the scheduling optimization of cross-location resource supply chain in cloud manufacturing environment), the resources from various locations should be subjected to combinatorial optimization, and many factors need to be considered to optimize the cloud resource supply chain. From the perspective of market demand, the author set up an optimization model of cloud resource supply chain in an attempt to describe the exact resource locations, production, logistics, delay and other factors. The established model involves eight dimensions: production cost, logistics cost, processing time, logistics time, delay cost, mean product quality, mean credibility and mean reliability. In other words, the optimal model contains the information on production cost, supply chain, and resource quality.

(1) Objective function

The objective function of the optimal model can be defined as:

$$F = \min(f1, f2, f3, f4, f5, f6, f7, f8) \quad (3)$$

The processing time function:

$$f1 = \sum_{i=1}^m \sum_{j=1}^{K_i} (x_{ij} \times p(ij)) \quad (4)$$

where $P(ij)$ is the processing time of the first candidate resource of the i^{th} sub-task.

The logistics time function:

$$f2 = \sum_{j=1}^{K_1} (a_{0j} \times l(0j)) + \sum_{i=1}^{m-1} \sum_{j=1}^{K_i} \sum_{k=1}^{K_{i+1}} (b_{ij,(i+1)k} \times l(ij, (i+1)k)) + \sum_{j=1}^{K_m} (a_{j0} \times l(j0)) \quad (5)$$

where $\sum_{j=1}^{K_1} (a_{0j} \times l(0j))$ is the logistics time between the resource requester and the candidate resource of the first subtask; $\sum_{i=1}^{m-1} \sum_{j=1}^{K_i} \sum_{k=1}^{K_{i+1}} (b_{ij,(i+1)k} \times l(ij, (i+1)k))$ is the logistics time between candidate resources of different locations of two adjacent subtasks; $\sum_{j=1}^{K_m} (a_{j0} \times l(j0))$ is the logistics time between the first candidate resource and the resource requester of the previous subtask.

The production cost function:

$$f3 = \sum_{i=1}^m \sum_{j=1}^{K_i} (x_{ij} \times q(ij)) \quad (6)$$

where $q(ij)$ is the production cost of the j^{th} candidate resource of the i^{th} subtask.

The logistics cost function:

$$f4 = \sum_{j=1}^{K_1} (a_{0j} \times g(0j)) + \sum_{i=1}^{m-1} \sum_{j=1}^{K_i} \sum_{k=1}^{K_{i+1}} (b_{ij,(i+1)k} \times g(ij, (i+1)k)) + \sum_{j=1}^{K_m} (a_{j0} \times g(j0)) \quad (7)$$

where $\sum_{j=1}^{K_1} (a_{0j} \times g(0j))$ is the logistics cost between the resource requester and the candidate resource of the first subtask; $\sum_{i=1}^{m-1} \sum_{j=1}^{K_i} \sum_{k=1}^{K_{i+1}} (b_{ij,(i+1)k} \times g(ij, (i+1)k))$ is the logistics cost between the resources of different locations for two adjacent subtasks; $\sum_{j=1}^{K_m} (a_{j0} \times g(j0))$ is the logistics cost between the resource requester and the candidate resource of the previous subtask.

The delay cost function:

$$f5 = \sum_{i=1}^m \sum_{j=1}^{K_i} (x_{ij} \times \gamma \times \max\{0, D_i - C_{ij}\}) \quad (8)$$

where D_i is the delivery date of the i^{th} subtask; C_{ij} is the completion time of the j^{th} candidate resource the first subtask; γ is the unit delay cost.

The mean product quality function:

$$f6 = (\sum_{i=1}^m \sum_{j=1}^{K_i} (x_{ij} \times Qa(ij))) / m \quad (9)$$

where $Qa(ij)$ is the product quality of the j^{th} candidate resource of the i^{th} subtask. The value of $Qa(ij)$ is a real number in (0, 1).

The mean reliability function:

$$f7 = (\sum_{i=1}^m \sum_{j=1}^{K_i} (x_{ij} \times Re(ij))) / m \quad (10)$$

where $Re(ij)$ is the reliability of the j^{th} candidate resource of the i^{th} subtask. The value of $Re(ij)$ is a real number in (0, 1).

The mean credibility function:

$$f8 = (\sum_{i=1}^m \sum_{j=1}^{K_i} (x_{ij} \times Rep(ij))) / m \quad (11)$$

where $Rep(ij)$ is the credibility of the j^{th} candidate resource of the i^{th} subtask. The value of $Rep(ij)$ is a real number in (0, 1).

The information on product cost is illustrated in Eqs. (4) and (6). The former specifies the total processing time of all resources in the supply chain, while the latter specifies the total production cost of all resources in the supply chain.

The information on supply chain is provided in Eqs. (5), (7) and (8). Specifically, Eq. (5) describes the logistics time, Eq. (7) specifies the logistics cost and Eq. (8) represents the

total the penalty cost of the resources selected for the subtasks, which arises from the production delay. Note that the penalty cost is proportional to the delay time.

The information on product quality is shown in Eqs. (9), (10) and (11). The data of these equations are acquired from the cloud manufacturing platform after user evaluation of cloud resources. The minimum required product quality, reliability and credibility of the resources of the selected resource are denoted as Qa_{min} , Re_{min} and Rep_{min} , respectively.

(2) Constraints

The established model is subjected to the following constraints:

1) Only one resource can be selected from the K_1 candidate resources of the first subtask.

$$\sum_{j=1}^{K_1} x_{ij} = 1 \quad (12)$$

2) The resource requester cannot establish logistics connections with more than one resource within the K_1 candidate resources of the first subtask.

$$\sum_{j=1}^{K_1} a_{0j} = 1 \quad (13)$$

3) The j^{th} candidate resource of the i^{th} subtask can only have a logistics connection with the K_{i+1} candidate resources of the $(i + 1)^{\text{th}}$ subtask.

$$\sum_{j=1}^{K_i} \sum_{k=1}^{K_{i+1}} b_{ij,(i+1)k} = 1 \quad (14)$$

4) The resource requester can only have a logistics connection with one of the K_m candidate resources of the m^{th} subtask.

$$\sum_{j=1}^{K_m} a_{j0} = 1 \quad (15)$$

5) The mean product quality of the selected supply chain must reach the pre-set minimum value.

$$f6 > Qa_{min} \quad (16)$$

6) The mean reliability of the selected supply chain must reach the pre-set minimum value.

$$f7 > Re_{min} \quad (17)$$

7) The mean credibility of the selected supply chain must reach the pre-set minimum value.

$$f8 > Rep_{min} \quad (18)$$

4. MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION (PSO)

Focusing on the membership degree of fuzzy sets, the multi-objective optimization method based on the fuzzy correlation entropy (FCE) uses the relative membership function to transform multi-objective Pareto solution and multi-objective ideal solution into fuzzy sets, thereby eliminating the impacts of target magnitude/dimension and the subjectivity of target weight selection. This section combines the FCE and the PSO into an FCE-based multi-objective PSO algorithm, in preparation for scheduling optimization of cloud resource supply chain.

4.1 Basic PSO

The basic PSO [18], proposed by Kennedy, is an intelligent swarm search algorithm inspired by the migration and clustering behaviours of foraging birds. It stands with ant colony algorithm as two famous swarm intelligence algorithms.

In the basic PSO, the particle swarm is self-organized in that each particle can transfer the relevant information in a d -dimensional space by a certain rule and adjust its state with the change of information. The information in the particle swarm comes from the matrix

$P = (P_1, P_2, \dots, P_m)$ containing the individual extremum of each particle. The information extracted from the matrix includes the optimal position of the swarm (the global best position) and that of each particle (the individual best position). The cooperation and information transfer between the particles ensure the good search performance of the PSO.

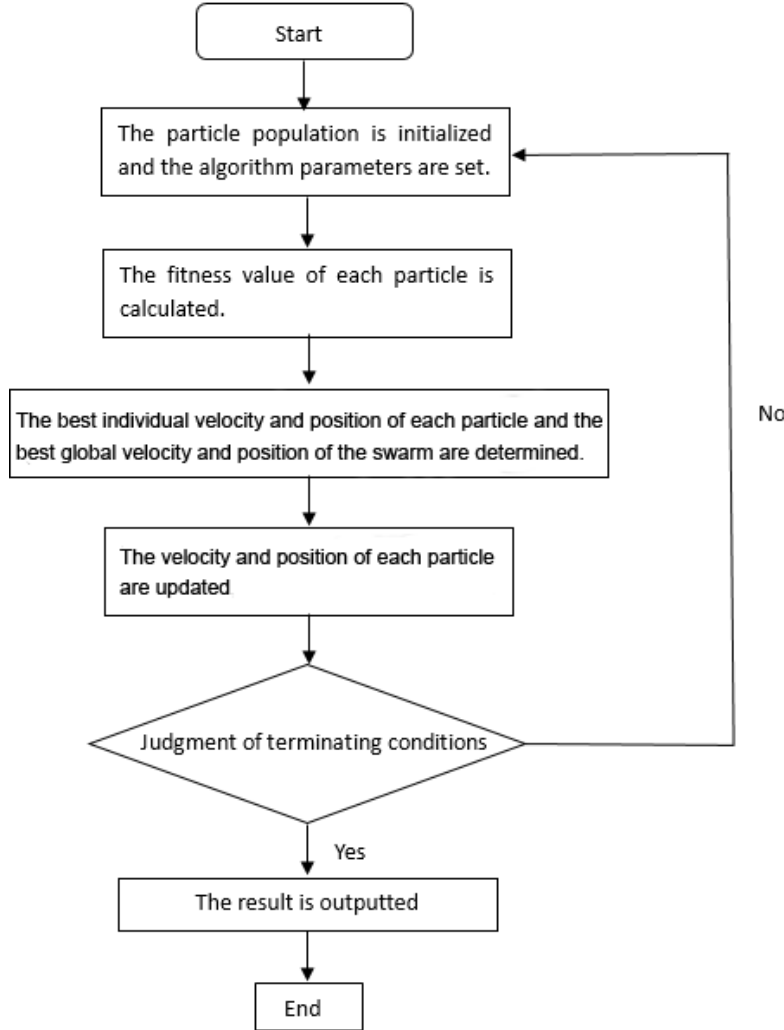


Figure 1: The flow chart of the basic PSO.

Assuming that N particles of a swarm are flying at different velocities in the d -dimensional space, the state properties of particle i at time t can be expressed as follows:

The position of particle i :

$x_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{id}^t)$, where $x_{id}^t \in (L_d, U_d)$ with L_d and U_d being the lower and upper limits of the search space, respectively.

The velocity of particle i :

$v_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{id}^t)$, where $v_{id}^t \in (-v_{max,d}, v_{max,d})$ with $v_{max,d}$ being the maximum velocity of the particle.

The individual best position of particle i :

$P_i^t = (P_{i1}^t, P_{i2}^t, \dots, P_{iD}^t)$, where $1 \leq d \leq D, 1 \leq i \leq n$.

The global best position of the swarm:

$P_g^t = (P_{g1}^t, P_{g2}^t, \dots, P_{gD}^t)$.

At time $t+1$ of the search process, the swarm will update the position and velocity of each particle as follows:

$$v_{id}^{t+1} = v_{id}^t + c_1 \times r_1 (P_{id}^t - x_{id}^t) + c_2 \times r_2 (P_{gd}^t - x_{id}^t) \quad (19)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (20)$$

where x_{id}^t and v_{id}^t are the d -dimensional components of particle i 's position and velocity at time t , respectively; P_{id}^t and P_{gD}^t are the d -dimensional components of the individual optimal position of particle i and the global optimal position of the swarm of the current search; c_1 and c_2 are learning factors, a.k.a. acceleration constants (usually, $c_1 = c_2 = 2$); r_1 and r_2 are random numbers in uniform distribution (0, 1). Fig. 1 is the flow chart of the basic PSO.

4.2 FCE-based multi-objective PSO

The PSO differs from the traditional multi-objective evolutionary algorithm in that the particles can share information with each other and the swarm can be updated by particle position and velocity without complex evolutionary operations like replication, crossover and mutation. With simple principle, few parameters, easy coding and convenient implementation, the PSO has been applied to various cases of multi-objective optimization in recent years. Nevertheless, this algorithm faces the same defects in fitness allocation mechanism as traditional multi-objective evolutionary algorithms like the genetic algorithm.

Acquired by the FCE, the FCE coefficient serves as the criterion to judge the advantages and disadvantages of the multi-objective optimization solution. It provides a new fitness allocation strategy to guide the evolution of multi-objective optimization algorithm. Following this strategy, the multi-objective Pareto solution and multi-objective ideal solution can be transformed into a fuzzy set sequence. In this sequence, the FCE coefficients of every two Pareto multi-objective fuzzy sets should be computed, and the multi-objective Pareto solution with FCE coefficient must be closer to the multi-objective ideal solution, and should be identified as the multi-objective optimal solution.

Since the FCE method can overcome the defects of the traditional multi-objective optimization algorithm in fitness allocation strategy (e.g. target weight selection and Pareto priority ordering), the FCE was integrated with the multi-objective PSO algorithm into an FCE-based multi-objective PSO algorithm. Fig. 2 presents the flow chart of the proposed algorithm.

5. SCHEDULING OPTIMIZATION MODEL OF CLOUD RESOURCE SUPPLY CHAIN

5.1 Scheduling optimization of cloud resource supply chain

The scheduling optimization of cloud resource supply chain can be split into several processes: task decomposition, resource search and matching, primary resource selection, and resource optimization. The detailed process is as follows:

Step 1. Task decomposition: the cloud manufacturing platform decomposes each manufacturing task of the resource requester into several sub-tasks according to process requirements.

Step 2. Resource search and matching: For each subtask, a certain number of supply chain resource sets matching the basic attribute conditions are searched for from the cloud resource pool. The cloud manufacturing platform monitors the real-time resource status in each supply chain, and discovers the incorrect resources in time.

Step 3. Primary resource selection: The supply chain resources matching each subtask are searched for. Among them, the resources with substandard processing cost, processing time and product quality are eliminated by the relative membership function, leaving the qualified resources.

Step 4. Resource optimization: The qualified resources are optimized by the proposed algorithm, and then the optimal resources in cloud resource supply chain are selected by the objective function method.

Step 5: If there is no unqualified resource in Steps 2-4, go to Step 6. Otherwise, go to Step 2.

Step 6: Output the global best cloud resource supply chain.

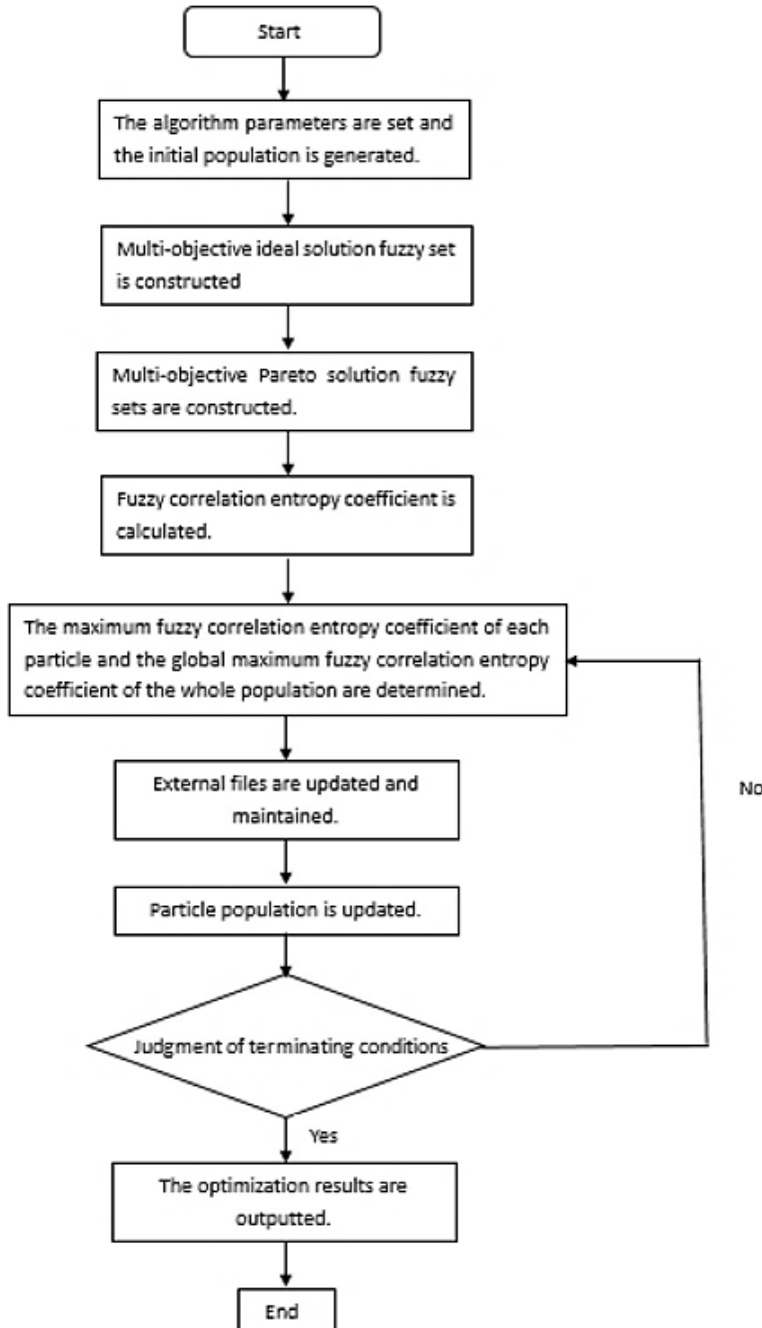


Figure 2: The flow chart of the proposed algorithm.

5.2 Discrete event simulation of the optimization process

The scheduling optimization model of cloud resource supply chain without unqualified resources is shown in Fig. 3 below, where the required task is decomposed into five serial subtasks and each subtask has eight matching cloud resources.

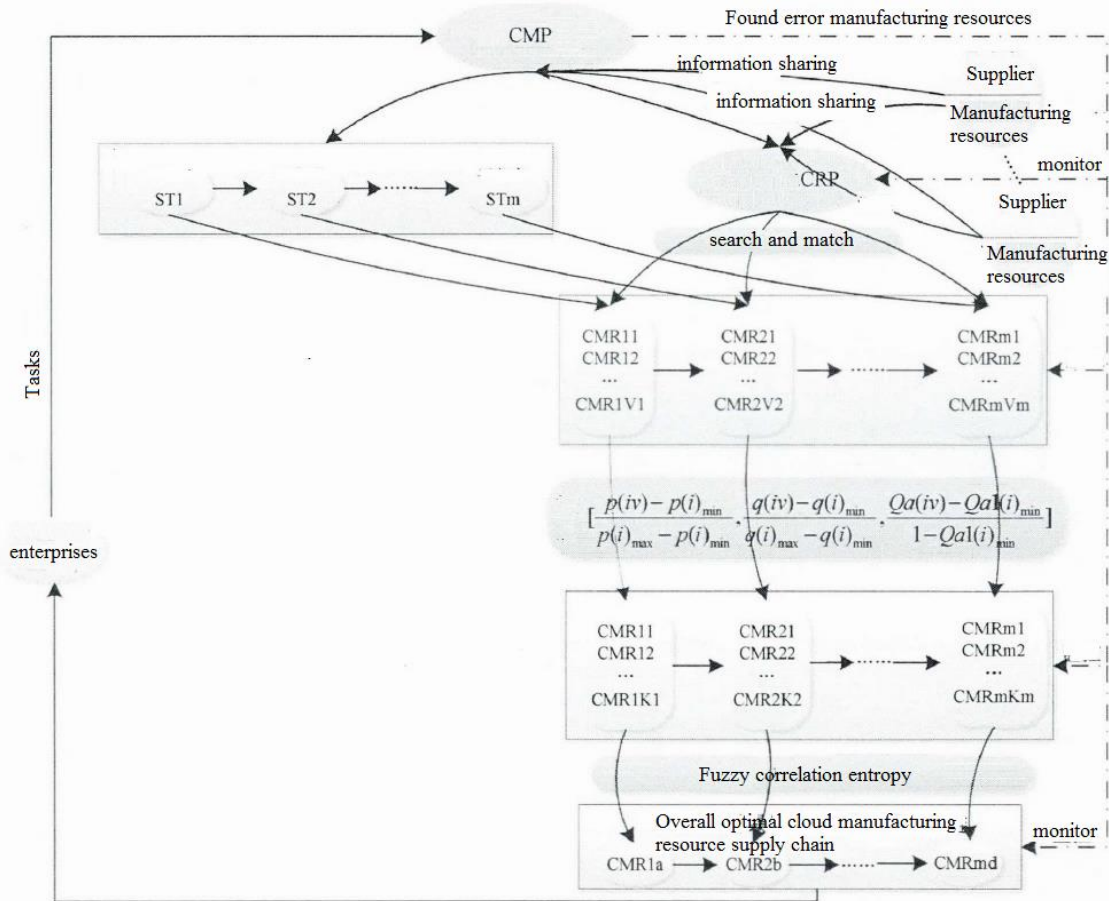


Figure 3: The scheduling optimization model without unqualified resources.

Table I is the list of cloud resources matching each subtask.

Table I: Cloud resources matching each subtask.

Subtask	Matched cloud resources
ST1	CMR11, CMR12, CMR13, CMR14, CMR15, CMR16, CMR17, CMR18
ST2	CMR21, CMR22, CMR23, CMR24, CMR25, CMR26, CMR27, CMR28
ST3	CMR31, CMR32, CMR33, CMR34, CMR35, CMR36, CMR37, CMR38
ST4	CMR41, CMR42, CMR43, CMR44, CMR45, CMR46, CMR47, CMR48
ST5	CMR51, CMR52, CMR53, CMR54, CMR55, CMR56, CMR57, CMR58

The primary selection follows the resource searching and matching of all subtasks. The subtask ST1 is cited as an example to illustrate the primary selection. The primary indices of the primary selection of ST1 are shown in Table II.

Table II: Primary indices of subtask ST1.

	Processing time (h)	Production cost	Product quality
CMR11	5	200	0.87
CMR12	6	270	0.75
CMR13	10	220	0.85
CMR14	11	490	0.78
CMR15	8	190	0.80
CMR16	12	610	0.71
CMR17	10	450	0.77
CMR18	2	580	0.65

For the other subtasks, the relative membership function was also called to select the matching resources, yielding the candidate resources for each subtask (Table III).

Table III: Candidate cloud resources for each subtask.

Subtask	Candidate cloud resources
ST1	CMR11, CMR12, CMR15, CMR16
ST2	CMR22, CMR26, CMR28
ST3	CMR33, CMR37
ST4	CMR41, CMR42, CMR44, CMR45, CMR48
ST5	CMR51, CMR53, CMR55

The FCE-based multi-objective PSO was adopted for scheduling optimization under the swarm size of 50 and the maximum number of iterations of 1,000. The processing time, production cost, product quality, credibility and reliability of candidate cloud resources are recorded in Table IV.

Table IV: The processing time, production cost, product quality, credibility and reliability of candidate cloud resources.

Candidate cloud resources	Processing time	Production cost	Product quality	Credibility	Reliability
CMR11	5.5	209	0.886	0.864	0.875
CMR12	7.5	320	0.776	0.832	0.796
CMR15	9	217	0.815	0.785	0.798
CMR16	11	386	0.738	0.769	0.808
CMR22	3.5	115	0.775	0.812	0.826
CMR26	7.5	586	0.832	0.875	0.808
CMR28	8	428	0.765	0.736	0.785
CMR33	10	586	0.605	0.789	0.766
CMR37	7.5	385	0.735	0.756	0.736
CMR41	6	295	0.826	0.802	0.816
CMR42	9	319	0.832	0.795	0.779
CMR44	7	432	0.764	0.755	0.794
CMR45	3	159	0.784	0.805	0.851
CMR48	9	632	0.833	0.902	0.779
CMR51	10	598	0.875	0.865	0.765
CMR53	12.5	452	0.786	0.813	0.828
CMR58	15	483	0.792	0.806	0.785

The eight-objective optimization results of the proposed algorithm are listed in Table V.

Table V: The eight-objective optimization results of the proposed algorithm.

	$f1$	$f2$	$f3$	$f4$	$f5$	$f6$	$f7$	$f8$
Proposed algorithm	32	32.1	1329	872	121	0.842	0.872	0.838
Ideal solution	30	29.8	1277	836	97	0.851	0.866	0.841

The results of discrete event simulation show that the ideal solution and optimal solution of the proposed algorithm were similar to each other, indicating that the combination between the FCE method and the PSO can effectively solve the scheduling optimization of cloud resource supply chain.

6. CONCLUSIONS

Cloud resource scheduling optimization, the key to cloud manufacturing system, is a core technique that reflects resource sharing, real-time dynamicity and fault-tolerance of cloud resources. In this paper, the FCE method is combined with the PSO into an FCE-based multi-objective optimization approach, in which the fitness of the FCE coefficient is taken as that of the multi-objective PSO to guide the algorithm evolution. After that, the discrete event simulation of the optimization process was carried out on FlexSim software. The simulation results show that the primary selection of relative membership function is stable and effective, and the proposed algorithm can obtain good supply chain resources in cloud manufacturing environment.

REFERENCES

- [1] Ostrosi, E.; Fougères, A.-J. (2018). Intelligent virtual manufacturing cell formation in cloud-based design and manufacturing, *Engineering Applications of Artificial Intelligence*, Vol. 76, 80-95, doi:[10.1016/j.engappai.2018.08.012](https://doi.org/10.1016/j.engappai.2018.08.012)
- [2] Li, H. Y.; Gui, C.; Xiao, K. (2018). Simulation of multivariate scheduling optimization for open production line based on improved genetic algorithm, *International Journal of Simulation Modelling*, Vol. 17, No. 2, 347-358, doi:[10.2507/IJSIMM17\(2\)CO9](https://doi.org/10.2507/IJSIMM17(2)CO9)
- [3] Wang, S.; Hu, Y. (2018). Binocular visual positioning under inhomogeneous, transforming and fluctuating media, *Traitement du Signal*, Vol. 35, No. 3-4, 253-276. doi:[10.3166/TS.35.253-276](https://doi.org/10.3166/TS.35.253-276)
- [4] Su, J.; Li, C.; Tsai, S.-B.; Lu, H.; Liu, A.; Chen, Q. (2018). A sustainable closed-loop supply chain decision mechanism in the electronic sector, *Sustainability*, Vol. 10, No. 4, Paper 1295, doi:[10.3390/su10041295](https://doi.org/10.3390/su10041295)
- [5] Xu, X. (2012). From cloud computing to cloud manufacturing, *Robotics and Computer-Integrated Manufacturing*, Vol. 28, No. 1, 75-86, doi:[10.1016/j.rcim.2011.07.002](https://doi.org/10.1016/j.rcim.2011.07.002)
- [6] Li, B.-H.; Zhang, L.; Wang, S.-L.; Tao, F.; Cao, J.-W.; Jiang, X.-D.; Song, X.; Chai, X.-D. (2010). Cloud manufacturing: a new service-oriented networked manufacturing model, *Computer Integrated Manufacturing Systems*, Vol. 16, No. 1, 1-7
- [7] Yan, K.; Cheng, Y.; Tao, F. (2015). A trust evaluation model towards cloud manufacturing, *International Journal of Advanced Manufacturing Technology*, Vol. 84, No. 1-4, 133-146, doi:[10.1007/s00170-015-8002-5](https://doi.org/10.1007/s00170-015-8002-5)
- [8] Patel, P. S. (2014). Multi-objective job scheduler using genetic algorithm in grid computing, *International Journal of Computer Applications*, Vol. 92, No. 14, 34-43, doi:[10.5120/16079-5312](https://doi.org/10.5120/16079-5312)
- [9] Laili, Y.; Tao, F.; Zhang, L.; Ren, L. (2011). The optimal allocation model of computing resources in cloud manufacturing system, *Proceedings of the Seventh International Conference on Natural Computation*, 2322-2326, doi:[10.1109/ICNC.2011.6022564](https://doi.org/10.1109/ICNC.2011.6022564)
- [10] Evola, G.; Marletta, L.; Cimino, D. (2018). Weather data morphing to improve building energy modelling in an urban context, *Mathematical Modelling of Engineering Problems*, Vol. 5, No. 3, 211-216, doi:[10.18280/mmep.050312](https://doi.org/10.18280/mmep.050312)
- [11] Bere, P.; Berce, P.; Nemes, O. (2012). Phenomenological fracture model for biaxial fibre reinforced composites, *Composites Part B: Engineering*, Vol. 43, No. 5, 2237-2243, doi:[10.1016/j.compositesb.2012.01.073](https://doi.org/10.1016/j.compositesb.2012.01.073)
- [12] Deese, K.; Geilen, M.; Rieg, F. (2018). A two-step smoothing algorithm for an automated product development process, *International Journal of Simulation Modelling*, Vol. 17, No. 2, 308-317, doi:[10.2507/IJSIMM17\(2\)437](https://doi.org/10.2507/IJSIMM17(2)437)
- [13] Rahdari, V.; Soffianian, A. R.; Pourmanafi, S.; Ghaiumi, M. H.; Mosadeghi, R.; Amiri, F. (2018). A multi-objective approach for land conservation capability evaluation using multi-criterion evaluation models, *Applied Ecology and Environmental Research*, Vol. 16, No. 2, 1353-1367, doi:[10.15666/aeer/1602_13531367](https://doi.org/10.15666/aeer/1602_13531367)

- [14] Shao, Z.; Pi, D.; Shao, W. (2018). A multi-objective discrete invasive weed optimization for multi-objective blocking flow-shop scheduling problem, *Expert Systems with Applications*, Vol. 113, 77-99, doi:[10.1016/j.eswa.2018.06.020](https://doi.org/10.1016/j.eswa.2018.06.020)
- [15] Li, B.; Guo, C.; Ning, T. (2018). An improved bacterial foraging optimization for multi-objective flexible job-shop scheduling problem, *Journal Européen des Systèmes Automatisés*, Vol. 51, No. 4-6, 323-332, doi:[10.3166/JESA.51.323-332](https://doi.org/10.3166/JESA.51.323-332)
- [16] Blasco, X.; Herrero, J. M.; Sanchis, J.; Martínez, M. (2008). A new graphical visualization of n -dimensional Pareto front for decision-making in multi objective optimization, *Information Sciences*, Vol. 178, No. 20, 3908-3924, doi:[10.1016/j.ins.2008.06.010](https://doi.org/10.1016/j.ins.2008.06.010)
- [17] Sun, G.; Bin, S. (2018). A new opinion leaders detecting algorithm in multi-relationship online social networks, *Multimedia Tools and Applications*, Vol. 77, No. 4, 4295-4307, doi:[10.1007/s11042-017-4766-y](https://doi.org/10.1007/s11042-017-4766-y)
- [18] Kennedy, J. (2011). Particle swarm optimization, Sammut, C.; Webb, G. I. (Eds.), *Encyclopedia of Machine Learning*, Springer, New York, 760-766, doi:[10.1007/978-0-387-30164-8_630](https://doi.org/10.1007/978-0-387-30164-8_630)