

Research Article

Multiobjective Optimization of Cloud Manufacturing Service Composition with Improved Particle Swarm Optimization Algorithm

Yongxiang Li ^{1,2}, Xifan Yao ², and Min Liu²

¹School of Mechanical Engineering, Guizhou University of Engineering Science, Bijie 551700, Guizhou, China

²School of Mechanical & Automotive Engineering, South China University of Technology, Guangzhou 510640, Guangdong, China

Correspondence should be addressed to Xifan Yao; mexfyao@scut.edu.cn

Received 7 March 2020; Revised 4 August 2020; Accepted 25 September 2020; Published 14 October 2020

Academic Editor: Agathoklis Giaralis

Copyright © 2020 Yongxiang Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at the problems of low search efficiency and inaccurate optimization of existing service composition optimization methods, a new multiobjective optimization model of cloud manufacturing service composition was constructed, which took service matching degree, composition synergy degree, cloud entropy, execution time, and execution cost as optimization objectives, and an improved particle swarm optimization algorithm (IPSOA) was proposed. In the IPSOA, the integer encoding method was used for particle encoding. The inertia coefficient and two acceleration coefficients were improved by introducing the normal cloud model, sine function, and cosine function. The global search ability of IPSOA in the early stage was improved, and its prematurity was restrained to form a more comprehensive solution space. In the later stage, IPSOA focused on the local fine search and improved the optimization precision. Taking automatic guided forklift manufacturing task as an example, the correctness of the proposed multiobjective optimization model of cloud manufacturing service composition and the effectiveness of its solution algorithm were verified. The performance of IPSOA was analyzed and compared with standard genetic algorithm (SGA) and traditional particle swarm optimization (PSO). Under the same conditions, IPSOA had a faster convergence speed than PSO and SGA and had better performance than PSO.

1. Introduction

In today's world, the trend of manufacturing globalization, diversification of consumer demand, and shortening of product marketization cycle has brought great challenges to traditional manufacturing enterprises. It is difficult for a single enterprise to meet all the customers' needs, and it is also difficult for a single enterprise to possess all the manufacturing resources needed for product manufacturing. In order to effectively overcome the shortcomings of the traditional production mode, cloud manufacturing has developed rapidly in the past ten years. Based on the idea of centralized use of decentralized resources and serving decentralized users with centralized resources [1], cloud manufacturing is a new service-oriented, demand-driven, on-demand payment, efficient and low consumption, and knowledge-based networked manufacturing mode [2].

Cloud manufacturing is developed on the basis of cloud computing technology. It integrates service-oriented technology, Internet technology, communication technology, modern logistics technology, Internet of Things technology, high-performance computing, and artificial intelligence technology to virtualize and servitize all kinds of manufacturing resources and manufacturing capabilities of resource providers, so as to achieve unified and centralized intelligent management and operation. Furthermore, it can provide timely, safe, high-quality, and low-cost cloud manufacturing services for resource users [3].

A single manufacturing cloud service is often difficult to meet the user's needs. A complex manufacturing requirement needs to combine multiple fine-grained simple cloud services into coarse-grained complex cloud services. Complex cloud services are implemented in distributed, heterogeneous, and autonomous environments to accomplish

manufacturing tasks, which are highly uncertain and dynamic. Optimal selection of cloud manufacturing services is one of the key technologies of cloud manufacturing and an important part of service management of cloud manufacturing platform [4]. The advantages and disadvantages of cloud manufacturing service composition model and its solution methods affect the rapid and efficient use of manufacturing resources in cloud manufacturing environment. It has become a hot issue in the field of cloud manufacturing research. In this paper, we study the mathematical model of multiple influence factors in cloud manufacturing service composition and the service composition optimization algorithm.

The remaining chapters of this paper are arranged as follows: Section 2 comprehensively analyzes the research work done by domestic and foreign scholars on cloud manufacturing service composition optimization; Section 3 gives the definitions and calculation methods of cloud entropy, service matching degree, composition synergy degree, execution time, and execution cost; Section 4 proposes the IPSOA algorithm; Section 5 analyzes and verifies the performance of the proposed optimization algorithm through application example; and Section 6 summarizes the whole paper and puts forward the future work.

2. Literature Review

In recent years, many scholars have used genetic algorithm, bee colony algorithm, particle swarm optimization algorithm, and other methods to study the modeling and optimization of cloud manufacturing service composition. For example, Thekinen and Panchal [5] regarded resource allocation in cloud environment as a two-way matching problem. Four kinds of two-way matching mechanisms were classified from individual rationality, stability, antistrategy, consistency, monotony, and Pareto efficiency, including deferred acceptance, top trading cycle, Munkres, and first come first service. Through Delphi research on the attributes of cloud service quality, Lang et al. [6] determined that service function, legality, contract, geographical location, and flexibility were the highest service quality evaluation criteria for cloud service selection. Răileanu et al. [7] combined energy consumption with product scheduling and resource allocation and proposed a design method of high availability production management system based on cloud. Helo and Hao [8] proposed a dynamic optimization model of production planning and control based on cloud for sheet metal processing and developed a scheduling prototype system based on the genetic algorithm. Chen and Wang [9] proposed a classified artificial neural network ensemble method to predict the time required to simulate cloud manufacturing tasks. K-means was used to classify the simulated manufacturing cloud tasks. For each task category, an artificial neural network was constructed to predict the time required for manufacturing cloud tasks in the category. Namjoo and Keramati [10] used resource-based theory and Dematel method to study the causality between the dimensions and attributes of composite service elasticity in cloud manufacturing. Souza et al. [11] studied the

distributed service layout strategy in mixed fog-cloud scenarios and proposed a concurrent service execution scheme. Brant and Sundaram [12] carried out the application experiment of manufacturing cloud. Under the condition of manufacturing cloud, the micrometal materials were manufactured by indoor electrochemical deposition technology. The horizontal deposition parameters were optimized based on the deposition resolution, and the manufacturing data were saved in the cloud for users to use on demand. Based on the formal description of cloud manufacturing resource allocation problem, Wang et al. [13] constructed a multiobjective resource allocation model with minimizing cost and time and optimizing quality. The multiobjective optimization problem was transformed into a single-objective optimization problem by the classical weighted summation method and solved by the maximum inheritance method. Zhou and Yao [14] proposed a multipopulation parallel adaptive differential artificial bee colony algorithm to optimize the selection of NP-hard for composite cloud manufacturing services. A number of parallel subpopulations were used. Each subpopulation evolved according to different mutation strategies borrowed from differential evolution. The control parameter of each mutation strategy was adjusted independently to generate disturbed food sources for foraging. Li et al. [15] studied self-governing cloud manufacturing service composition and optimization selection and proposed a fuzzy soft decision method based on volatility analysis. Li and Yao [16] constructed cloud manufacturing service description model, interaction scenario model, and composition process formal model based on process algebra, extended process algebra semantics to describe service quality information, and proposed an intelligent service composition method based on extended process algebra. Tao et al. [17] designed a cloud manufacturing service supply-demand matching simulator based on hypernetwork, which could compare service matching results and scheduling algorithm performance. Zhang et al. [18] studied a fuzzy QoS-aware manufacturing service composition method based on the extended pollination algorithm. Yang et al. [19] proposed a dynamic service selection method within multiple manufacturing cloud systems aiming to apply the Internet of Things, real-time sensors, big data, and event-driven dynamic service selection. Chen et al. [20] proposed a method called QoS-aware web service composition to help cloud demanders for service composition based on a multiobjective model and provided an efficient-dominance multiobjective evolutionary algorithm to fulfill the service composition model. The huge and ever-increasing number of web service providers in the cloud had the same manufacturing functions, yet they possessed different QoS indexes [21]. Therefore, most scholars conducted their research based on quality of service. A number of indexes such as cost, time, and reliability were utilized to form the overall objective functions in order to select the best possible composition for a specific task [22]. Quality of service could be described as a set of key performance index used to assess the quality of services composed in a cloud manufacturing system. Availability, reliability, cost, time, geographical position, and

technological capability were among the key indicators applied by researchers in service composition issues. Aiming at the problem of optimal selection of manufacturing service composition, Que et al. [23] proposed a new manufacturers-to-users model for cloud manufacturing, established a comprehensive mathematical evaluation model with four key service quality perception indicators (i.e., time, cost, reliability, and capability), and solved the model by using information entropy immune genetic algorithm. Huang et al. [24] combined genetic algorithm with particle swarm optimization, proposed a hybrid genetic particle swarm optimization algorithm based on teaching and learning, introduced learning mechanism into genetic algorithm, and enabled the descendants of genetic algorithm to learn the characteristics of elite chromosomes from double memory learning in the evolutionary process. The algorithm was searched for solutions in two subpopulations of genetic algorithm module and particle swarm optimization module and exchanged information simultaneously. Zhao et al. [25] proposed a SPEA2 algorithm based on adaptive selection evolutionary operator to solve multiobjective optimization problem. In the evolutionary process, the simulated binary crossover operator, polynomial mutation operator, and differential evolution operator could be selected adaptively according to the contribution of operators. Zhang et al. [26] proposed an extended teaching and learning optimization algorithm for parallel optimization of distributed manufacturing resource allocation. Xu and Cai [27] proposed an efficient global optimization algorithm based on multidata for automobile body design. The general computing technology based on graphics processing unit and the hybrid parallel computing method was used to improve the solving efficiency.

The above literature mainly considers time and cost as two quality factors, adopts the basic idea of transforming the multiobjective optimization problem of manufacturing cloud service composition optimization into a single-objective optimization problem, and uses traditional mature optimization algorithm to solve it indirectly [28]. However, these traditional solutions have obvious defects or deficiencies, which are mainly reflected as follows: firstly, the abovementioned studies mainly focus on the optimization of service quality parameters such as execution cost and time, but the impact of composition complexity, collaboration, matching property, and other nonfunctional service quality parameters in service composition is less considered; secondly, the selection of weight coefficients has strong subjectivity, and the optimization results are greatly influenced by the subjective factors. In the cloud manufacturing environment, each cloud service executing agent is in a certain social relationship, not an idealized “rigid body”. In the manufacturing process, different cloud manufacturing services carry out different data exchange, information transmission, and material transportation. They are constrained by each other in the manufacturing life cycle process and in which cooperation and competition coexist. The relationship between cloud manufacturing services and manufacturing tasks and the relationship between services directly affect the efficiency of service composition in performing

manufacturing tasks. In cloud manufacturing environment, customized product manufacturing to meet individual needs is a common process, which often requires collaborative participation of customers and service providers. Cloud manufacturing service composition not only needs to meet the requirements of traditional product delivery period and manufacturing costs, but also the matching degree between manufacturing tasks and cloud services, the synergy degree between cloud manufacturing services in the manufacturing process, and the service composition complexity. All of them have significant impacts on the completion of customized product manufacturing tasks. Therefore, it is necessary to improve the traditional optimization algorithm, explore new efficient cloud service composition optimization methods, and take service matching degree, composition synergy degree, and service composition complexity as optimization factors to study cloud manufacturing service composition optimization.

In the process of service composition, all manufacturing subtasks decomposed according to customer’s manufacturing needs must be allocated to one or more corresponding cloud services to complete. The optimization objectives of the optimal cloud manufacturing service composition scheme, such as service matching degree, composition synergy degree, cloud entropy, execution time, and cost, should be as close as possible to the ideal values. Based on the basic rules, the cloud manufacturing service composition optimization modeling, particle swarm optimization algorithm improvement, and its application are studied in the following research.

3. Mathematical Modeling of Cloud Manufacturing Service Composition

3.1. Service Composition Problem Description. In cloud manufacturing, a complex manufacturing task needs to be decomposed into several simple manufacturing subtasks to complete in most cases. By searching for simple cloud services matching each manufacturing subtask in the cloud manufacturing service platform for composition and optimization, the complex cloud service with coarse granularity is constructed to achieve the complex manufacturing task requirements. If a complex manufacturing task \mathbf{J} can be decomposed into m manufacturing subtasks, and it can be expressed as $\mathbf{J} = \{J_1, J_2, J_j, \dots, J_{m-1}, J_m\}$, where J_j is the j -th manufacturing subtask of the complex manufacturing task \mathbf{J} , $j = 1, 2, 3, \dots, m$. For each manufacturing subtask, the corresponding candidate manufacturing cloud services are searched in the cloud resource pool to form the candidate manufacturing cloud service set \mathcal{S}_j of the j -th manufacturing subtask. The j -th cloud service set can be expressed as $\mathcal{S}_j = \{S_{1,j}, S_{2,j}, \dots, S_{b_j,j}\}$, where b_j denotes the number of cloud services contained in the j -th cloud service set and $S_{b_j,j}$ denotes the b_j -th cloud service in the j -th cloud service set. The total number of cloud services in n cloud service sets $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_j, \dots, \mathcal{S}_n$ is $N = \sum_{j=1}^n b_j$. The cloud manufacturing service composition process is shown in Figure 1 [29].

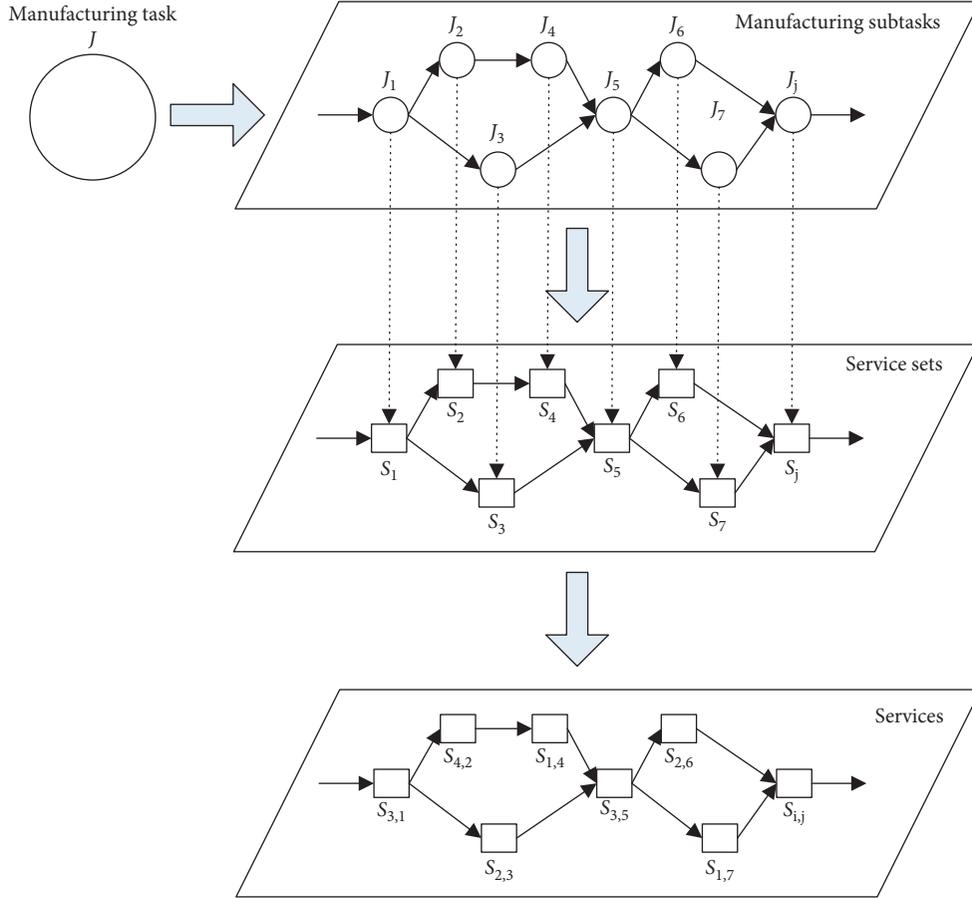


FIGURE 1: Cloud manufacturing service composition process [29].

3.2. Computation of Execution Time and Execution Cost. Minimum time and minimum cost are two basic principles in the operation of a company's business. In the research field of cloud manufacturing service composition, service execution time and execution cost are important indicators for performance evaluation of cloud manufacturing service composition schemes. According to the characteristics of manufacturing resources in cloud manufacturing environment, service execution time and execution cost are defined as follows.

Execution time (ET) is the response time for cloud manufacturing service to manufacturing task. Because cloud manufacturing service involves both online and offline factors, and its execution time is longer than that of ordinary web service. Moreover, the influence of logistics time in heterogeneous cloud manufacturing environment often becomes the bottleneck of the whole manufacturing process. Therefore, it is necessary to take logistics time into consideration of service quality. To sum up, the value of execution time is equal to the sum of the processing time of cloud manufacturing services, the time required for auxiliary work (such as equipment maintenance and workpiece clamping), and the time consumed for material transportation during service execution, namely,

$$ET = T_{\text{processing}} + T_{\text{auxiliary}} + T_{\text{logistics}}, \quad (1)$$

where ET is the execution time, $T_{\text{processing}}$ is the processing time, $T_{\text{auxiliary}}$ is the auxiliary time, and $T_{\text{logistics}}$ is the logistics time.

Execution cost (EC) is the cost that service users pay for using the cloud manufacturing service. The value of execution cost is equal to the sum of manufacturing cloud service charges identified by service providers, payment for material transportation in the service execution, and third-party service fees charged by cloud platform, namely,

$$EC = C_{\text{service}} + C_{\text{logistics}} + C_{\text{platform}}, \quad (2)$$

where EC is the execution cost, C_{service} is the cloud service cost, $C_{\text{logistics}}$ is the logistics cost, and C_{platform} is the platform cost.

The types of cloud manufacturing service composition include sequence composition, parallel composition, choice composition, and cycle composition, as shown in Figure 2 [30]. In the sequence composition, different services are executed sequentially according to their order in the service composition. As shown in Figure 2(a), the two services S_1 and S_2 in the composition constitute an orderly serial chain, and service S_2 can be executed only after service S_1 has been executed. In the parallel composition, the two services S_1 and S_2 are executed concurrently, as shown in Figure 2(b). Fork is the beginning transition of service operation in the parallel

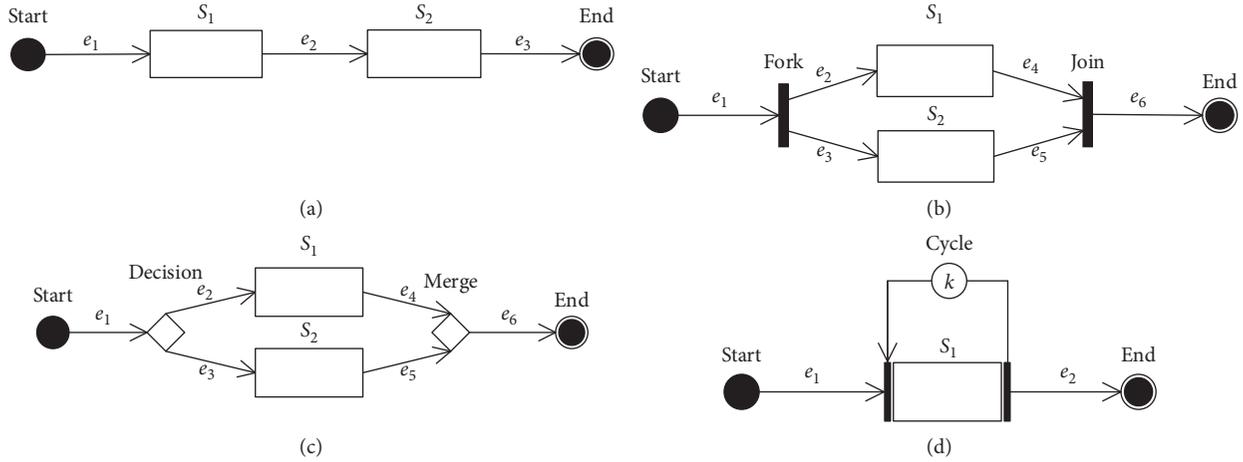


FIGURE 2: Schematic diagrams of four main types of service composition [30]: (a) sequence composition; (b) parallel composition; (c) choice composition; (d) cycle composition.

composition, and join is the end transition. Only when S_1 and S_2 are all executed, can the end transition join be triggered. In the choice composition, as shown in Figure 2(c), one of the services is selected. As long as the selected service is successfully executed, the choice composition is successfully executed. Decision is the beginning transition of service operation of choice composition, and merge is the end transition. Figure 2(d) shows the cycle composition, where service S_1 is repeatedly executed. Cycle is a cyclic operator. $\text{Cycle}(k)\{S_1\}$ indicates that service S_1 is repeatedly executed k times. Under different composition modes, the execution time and cost of cloud manufacturing service composition are different, as shown in Table 1 [30].

Screening candidate services according to service execution time and execution cost is an important work in cloud manufacturing service composition. However, it is not enough to screen candidate services only according to the two indicators. In order to fully reflect the service quality of candidate cloud manufacturing services and service composition, this paper takes execution time, execution cost, service matching degree, composition synergy degree, and cloud entropy as five quality evaluation criteria of service composition and then carries out cloud manufacturing service composition model construction, algorithm improvement, and case analysis.

3.3. Service Matching Degree Modeling. Service matching degree (MD) is a quantitative measure of the matching between cloud manufacturing services and allocated manufacturing tasks. Service matching degree reflects the request-response relationship between cloud services and manufacturing tasks. The main factors affecting service matching degree are the idleness of manufacturing resources, equipment status, comprehensive manufacturing capability, service reputation, cumulative times of cloud manufacturing services performing similar manufacturing tasks, active degree of cloud manufacturing services, and distance between manufacturing resources mapped by cloud

manufacturing services and service objects. According to the characteristics of different factors, the factors affecting service matching can be summarized as technical factor, hunger factor, and distance factor.

3.3.1. Technical Factor. Technical factor (TF) refers to evaluating the technical level of a cloud manufacturing service to accomplish a manufacturing task based on cumulative times of cloud manufacturing services performing similar manufacturing tasks, service reputation, service execution rate, service activity, and equipment performance in the past period of time. It is described by the rank vector $\text{TF}_{ij} = [0, 0.1, 0.2, 0.3, \dots, 0.9, 1]$, in which TF_{ij} represents the technical capability measurement of the i -th service that performs the j -th task, $1 \leq i \leq N$, $1 \leq j \leq m$. The technical capability of N cloud services that perform m manufacturing tasks can be described by $N \times m$ technical matrix $\text{TF} = (\text{TF}_{ij})_{N \times m}$.

3.3.2. Hunger Factor. One of the outstanding contributions of cloud manufacturing is to activate idle manufacturing resources. The idle rate of manufacturing resources refers to the ratio of the number of available manufacturing resources unused for manufacturing tasks to the total number of manufacturing resources. According to the idle rate of manufacturing resources mapped by cloud manufacturing services, the desire level of the cloud manufacturing services to undertake and complete manufacturing tasks can be evaluated, which is described as hunger factor HF_{ij} . HF_{ij} represents the hunger degree of the i -th service to undertake and complete the j -th task, $1 \leq i \leq N$, $1 \leq j \leq m$. The range of HF_{ij} is $[0, 1]$. The higher the idle rate of manufacturing resources mapped by cloud manufacturing services, the more hungry and thirsty for accepting manufacturing tasks, and the bigger the hunger factor; on the contrary, the smaller the hunger factor. The value of hunger factor can be obtained by calculating the idle rate of manufacturing resources, and the hunger factor matrix is $\text{HF} = (\text{HF}_{ij})_{N \times m}$.

TABLE 1: Calculation of execution time and cost for different types of service composition [30].

No.	Composition types	Execution time	Execution cost
1	Sequence composition	$ET_{\text{sequence}} = \sum ET(S_i)$	$EC_{\text{sequence}} = \sum EC(S_i)$
2	Parallel composition	$ET_{\text{parallel}} = \text{Max}(ET(S_i))$	$EC_{\text{parallel}} = \sum EC(S_i)$
3	Choice composition	$ET_{\text{choice}} = \sum (\theta_i \cdot ET(S_i))$	$EC_{\text{choice}} = \sum (\theta_i \cdot EC(S_i))$
4	Cycle composition	$ET_{\text{cycle}} = k \cdot ET(S_i)$	$EC_{\text{cycle}} = k \cdot EC(S_i)$

Note. θ_i is the probability that the i -th service in the choice composition is selected, and $\sum \theta_i = 1$. If the i -th service must not be selected, then $\theta_i = 0$. If the i -th service must be selected, then $\theta_i = 1$, and the probability of other services being selected is 0. k denotes the cycle number.

3.3.3. Distance Factor. Cloud manufacturing services are more complicated than ordinary web services due to the influence of offline factors in the execution process. Spatial distance often becomes a constraint in the execution process of cloud manufacturing services. Generally, small spatial distance is beneficial to the execution of cloud manufacturing services, while large spatial distance is not conducive to the execution of cloud manufacturing services. Distance factor is introduced to describe the impact of the relative distance between manufacturing resources mapped by cloud manufacturing services and service users on the execution of cloud manufacturing services. It is expressed as DF_{ij} , $1 \leq i \leq N$, $1 \leq j \leq m$. DF_{ij} represents the distance factor between the manufacturing resources mapped by the i -th cloud manufacturing service and the j -th service users. The range of DF_{ij} is $[0, 1]$. The smaller the relative distance, the larger the distance factor; on the contrary, the smaller the distance factor. For example, if the distance between cloud manufacturing resources and cloud service users is 300 km, then $DF_{ij} = 0.4$; if the distance is 50 km, then $DF_{ij} = 0.8$; if the distance is less than 1 km, then $DF_{ij} = 1$; similarly, other corresponding distance factors can be obtained.

In summary, service matching degree matrix can be calculated as follows:

$$MD = \begin{pmatrix} MD_{11} & MD_{12} & \cdots & MD_{1,m-1} & MD_{1m} \\ MD_{21} & \cdots & \cdots & \cdots & MD_{2,m} \\ \cdots & \cdots & MD_{ij} & \cdots & \cdots \\ MD_{N-1,1} & \cdots & \cdots & \cdots & MD_{N-1,m} \\ MD_{N,1} & MD_{N,2} & \cdots & MD_{N,m-1} & MD_{N,m} \end{pmatrix}, \quad (3)$$

where MD_{ij} represents the matching degree of the i -th service to the j -th task; $MD_{ij} = \alpha \times TF_{ij} + \beta \times HF_{ij} + \gamma \times DF_{ij}$, $1 \leq i \leq N$, $1 \leq j \leq m$; α , β , and γ are the weight coefficients. Every influencing factor has different importance to service matching degree, and the three influencing factors can be given different weight coefficients. The sum of the weight coefficients of the three factors is equal to 1, i.e., $\alpha + \beta + \gamma = 1$, which is helpful for calculating service matching degree and evaluating the importance of the three factors.

3.4. Composition Synergy Degree Modeling. Composition synergy degree (SD) represents the collaboration level among cloud manufacturing services that are composed to perform a complicated manufacturing task. Composition synergy degree reflects the collaborative relationship between two cloud manufacturing services in service composition. In service composition, the more convenient the information exchange between cloud manufacturing services and the smoother the material transportation, the shorter the time for them to cooperate to complete manufacturing tasks, and the higher the composition synergy degree between cloud manufacturing services; on the contrary, the lower the composition synergy degree [31]. With low composition synergy degree, there are serious obstacles to information exchange and material transportation between cloud manufacturing services, which can delay product delivery and increase execution cost. The composition synergy degree of cloud manufacturing services is directly reflected in execution time. The composition synergy degree can be evaluated by calculating the time taken by cloud manufacturing services to complete manufacturing tasks. For example, the composition synergy degree between services S_i and S_j used to complete tasks J_i and J_j is calculated as follows:

$$SD_{ij} = \frac{T_i + T_j}{T_{ij}}, \quad (4)$$

where T_i is the time taken by service S_i to complete task J_i independently, T_j is the time taken by S_j to complete J_j independently, and T_{ij} is the total time taken by two services S_i and S_j to synergically perform two tasks J_i and J_j . The relationship structure between tasks affects the calculation of execution time. There are mainly parallel, sequential, and interactive coupling relationships among manufacturing tasks. It is relatively easy to calculate the total execution time of complex manufacturing tasks composed of independent manufacturing subtasks. However, the calculation of the total execution time of complex manufacturing tasks with interactive coupling relationship is rather complicated. The calculation for T_{ij} in different cases is listed in the following equation:

$$T_{ij} = \begin{cases} \max[T_i, T_j], & \text{if manufacturing tasks } J_i \text{ and } J_j \text{ are parallel and independent with each other,} \\ T_i + T_j, & \text{if manufacturing tasks } J_i \text{ and } J_j \text{ are sequential and independent with each other,} \\ T_i + T_j + 2\zeta_{ij}\sqrt{T_i \cdot T_j}, & \text{if there is an interactive coupling relationship between tasks } J_i \text{ and } J_j, \end{cases} \quad (5)$$

where ζ_{ij} is the interactive coupling coefficient, and its range is $[-1, 1]$. The higher the previous cooperation frequency and the better the smoothness of service interaction and material transportation, the smaller the ζ_{ij} value; on the contrary, the bigger the ζ_{ij} value. $T_{ij} = \max[T_i, T_j]$ is the total completion time calculation formula when manufacturing tasks J_i and J_j are independent and in the parallel relationship. $T_{ij} = T_i + T_j$ is the total completion time calculation formula when manufacturing tasks J_i and J_j are independent and in the sequence relationship. $T_{ij} = T_i + T_j + 2\zeta_{ij}\sqrt{T_i \cdot T_j}$ is a formula for calculating the total completion time of manufacturing tasks J_i and J_j when they are in the interactive coupling relationship. It is based on the electromagnetic coupling principle in physics. Because of the interactive coupling relationship between manufacturing tasks J_i and J_j , the completion process of two subtasks depends on each other and influences each other. The interactive coupling coefficient ζ_{ij} transforms the qualitative relationship of interdependence and mutual influence into a quantitative one, which is helpful for a clearer analysis of the characteristics and rules of the relationship [32].

Thus, the composition synergy degree matrix of cloud manufacturing service can be constructed as follows:

$$SD = \begin{pmatrix} SD_{11} & SD_{12} & SD_{13} & \cdots & SD_{1,N-1} & SD_{1,N} \\ SD_{21} & SD_{22} & \cdots & \cdots & \cdots & SD_{2,N} \\ SD_{31} & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & SD_{ij} & \cdots & \cdots \\ SD_{N-1,1} & \cdots & \cdots & \cdots & \cdots & SD_{N-1,N} \\ SD_{N,1} & SD_{N,2} & \cdots & \cdots & SD_{N,N-1} & SD_{N,N} \end{pmatrix}, \quad (6)$$

where SD_{ij} represents the composition synergy degree of the i -th cloud manufacturing service to the j -th cloud manufacturing service in service composition.

3.5. Cloud Entropy Modeling. Simple and orderly service composition structure has greater certainty to complete manufacturing tasks successfully. Complex and disordered cloud manufacturing service composition structure is prone to failure, and the probability of successfully completing manufacturing tasks is small. Entropy originated from thermodynamic research and has been widely recognized and accepted by the scientific community since it gained great development in information theory. Different cloud

manufacturing services have different continuous working times, maintenance times, and logistics times when performing manufacturing tasks. The process of cloud manufacturing services completing manufacturing tasks at one time is obviously simpler than that of dividing them into multiple stages to complete. The quantification level of complexity and orderliness of cloud manufacturing service composition can be expressed by cloud entropy (CE). The cloud entropy of the i -th cloud manufacturing service to complete corresponding manufacturing task can be calculated as follows [33]:

$$CE_i = - \sum_{j=1}^{Q_i} \frac{ST_{ij}}{TT_i} \ln \frac{ST_{ij}}{TT_i}, \quad (7)$$

where CE_i is the cloud entropy of the i -th cloud manufacturing service, ST_{ij} is the duration of the i -th cloud manufacturing service in the j -th state, TT_i is the total time of the i -th service for completing the corresponding task, and Q_i is the total state number of the i -th service for completing the task. The cloud entropy of a cloud manufacturing service composition is equal to the sum of cloud entropy of all cloud manufacturing services in the service composition scheme. Cloud entropy is used to measure the composition complexity. Its calculation formula is shown in equation (8) [33]. The smaller the cloud entropy is, the more orderly and simpler the cloud manufacturing service composition is, and the higher the reliability is:

$$CE = - \sum_{i=1}^N \sum_{j=1}^{Q_i} \frac{ST_{ij}}{TT_i} \ln \frac{ST_{ij}}{TT_i}, \quad (8)$$

where N is the total service number in the service composition and CE is the cloud entropy of cloud manufacturing service composition.

3.6. Multiobjective Optimization Model for Service Composition. The ultimate goal of cloud manufacturing service composition is to select the best cloud services to complete all of the users' manufacturing tasks. The resulting service composition scheme should meet the constraints of execution time and execution cost and make the cloud manufacturing service composition have the highest composition synergy degree, the biggest composition matching degree, the smallest cloud entropy, the smallest execution time, and the smallest execution cost. Cloud manufacturing service composition is a multiobjective optimization problem, and its mathematical model is constructed as follows:

$$\max Y_1 = \max \sum_{i=1}^N \sum_{j=1}^m (\text{MD}_{ij} \cdot \xi_{ij}), \quad (9)$$

$$\max Y_2 = \max \sum_{j=1}^m \sum_{k=1}^j \sum_{p=1}^N \sum_{q=1}^N (\text{SD}_{pq} \cdot \xi_{pj} \cdot \xi_{qk}), \quad (10)$$

$$\min Y_3 = \min \text{CE} = \min - \left(\sum_{i=1}^N \sum_{j=1}^{Q_i} \frac{\text{ST}_{ij}}{\text{TT}_i} \ln \frac{\text{ST}_{ij}}{\text{TT}_i} \right), \quad (11)$$

$$\min Y_4 = \min \text{ET} = \min (\max (T_1, T_2, T_3, \dots, T_{m-1}, T_m)), \quad (12)$$

$$\min Y_5 = \min \text{EC} = \min \sum_{j=1}^m \text{EC}_j = \min \sum_{j=1}^m \sum_{i=1}^N (T_i \cdot w_i \cdot \xi_{ij}), \quad (13)$$

$$\text{s.t. } \max (T_1, T_2, T_3, \dots, T_{m-1}, T_m) \leq \text{ET}_0 \quad (14)$$

$$\sum_{j=1}^m \text{EC}_j = \sum_{j=1}^m \sum_{i=1}^N (T_i \cdot w_i \cdot \xi_{ij}) \leq \text{EC}_0, \quad (15)$$

$$\sum_{i=1}^N \xi_{ij} \geq 1, \quad (16)$$

$$\xi_{ij} = \begin{cases} 1, & \text{if the } j\text{-th manufacturing task is allocated to the } i\text{-th cloud manufacturing service,} \\ 0, & \text{otherwise,} \end{cases} \quad (17)$$

where w_i represents the unit time cost and EC_j represents the execution cost of the j -th manufacturing task. Equations (9)–(13) are objective functions, in which equation (9) maximizes the total service matching degree, equation (10) maximizes the total composition synergy degree, and equations (11)–(13) represent the minimum values of the total cloud entropy, execution time, and execution cost of service composition, respectively. Equations (14)–(16) are constraints, in which equation (14) stipulates that the maximum execution time of m manufacturing tasks cannot exceed the threshold time ET_0 . Equation (15) stipulates that the maximum execution cost of m manufacturing tasks cannot exceed the threshold cost EC_0 . Equation (16) stipulates that every task must be allocated to one or more cloud services for execution.

4. Improved Particle Swarm Optimization Algorithm

The cloud manufacturing service composition problem is a NP-hard problem of multiobjective collaborative optimization. The traditional particle swarm optimization algorithm is easy to premature and fall into local extremum, and it is difficult to adapt to complex nonlinear optimization problem. Therefore, its optimization quality needs to be improved. This paper introduces the normal cloud model [34] to improve the inertia coefficient and

acceleration coefficient of the traditional particle swarm optimization algorithm, balances the global exploration and local development ability of the algorithm, and designs an improved particle swarm optimization algorithm (IPSOA) to solve the cloud manufacturing service composition mathematical model. The algorithm process is shown in Figure 3.

4.1. Particle Encoding Method. The mapping relationship between particle position vector and service composition scheme is established by the integer encoding method. In cloud manufacturing environment, if a multifunctional requirement task is decomposed into m subtasks, and each subtask is allocated to a candidate cloud service to perform; then, the service composition used by the complex manufacturing task is composed of m single resource cloud services. A cloud service composition is represented as an m -dimensional particle. The particle dimension is $\text{ParDim} = m$. The j -th dimension of the particle represents the j -th subtask J_j of the complex manufacturing task, and the task J_j corresponds to a cloud service set S_j , which contains multiple candidate cloud services for the task J_j . The value of the j -th dimension represents the candidate cloud service number selected by the task J_j . Assuming that the particle swarm consists of R particles, the position and velocity of the i -th particle can be represented by an m -dimensional vector as follows:

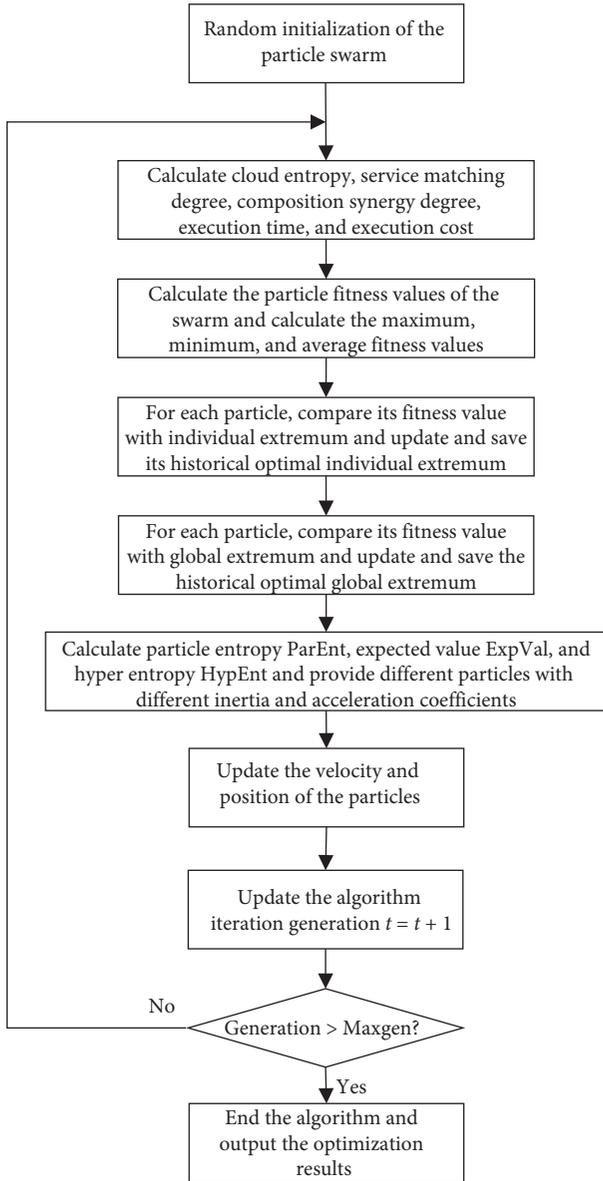


FIGURE 3: Flow chart of IPSOA.

$$x_i = (x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,m}), \quad i = 1, 2, 3, \dots, R, \quad (18)$$

$$v_i = (v_{i,1}, v_{i,2}, v_{i,3}, \dots, v_{i,m}), \quad i = 1, 2, 3, \dots, R \quad (19)$$

where $x_{i,j}$ and $v_{i,j}$ represent the position and velocity of the j -th dimension of the i -th particle, respectively. The position $x_{i,j}$ of the particle represents the candidate cloud service number selected by the task J_j in the process of service composition. Figure 4 shows the corresponding relationship between particle and service composition.

BP_i and BG_i represent the historical optimal position of the i -th particle (i.e., individual extremum) and the global optimum position of the whole particle swarm (i.e., global extremum), respectively. Both of them can be represented by an m -dimensional vector as follows:

Particle	3	4	2	1	3	...	1
Tasks	J_1	J_2	J_3	J_4	J_5	...	J_m
Service sets	S_1	S_2	S_3	S_4	S_5	...	S_m
Services	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,4}$	$S_{1,5}$...	$S_{1,m}$
	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$	$S_{2,4}$	$S_{2,5}$...	$S_{2,m}$
	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$	$S_{3,4}$	$S_{3,5}$...	$S_{3,m}$
	$S_{4,1}$	$S_{4,2}$	$S_{4,3}$	$S_{4,4}$	$S_{4,5}$...	$S_{4,m}$
	$S_{5,1}$	$S_{5,2}$	$S_{5,3}$	$S_{5,4}$	$S_{5,5}$...	$S_{5,m}$

FIGURE 4: The corresponding relationship between particle and service composition.

$$BP_i = (BP_{i,1}, BP_{i,2}, BP_{i,3}, \dots, BP_{i,m}), \quad i = 1, 2, 3, \dots, R, \quad (20)$$

$$BG = (BG_1, BG_2, BG_3, \dots, BG_m), \quad (21)$$

where $BP_{i,j}$ represents the optimal position of the i -th particle in the j -th dimension and BG represents the optimal service composition selected by the manufacturing task set $J = (J_1, J_2, J_3, \dots, J_m)$.

In the first iteration of the algorithm, the particle P_i in the particle swarm is randomly generated and assigned to BP_i , and the particle with the optimal fitness value is assigned to BG . In each subsequent iteration, BP_i and BG are updated according to particle fitness values. When the maximum iteration number is reached, the algorithm is terminated, and cloud manufacturing service composition with the optimal comprehensive service quality is obtained.

4.2. Velocity and Position Updating Method. The formulas for particle velocity and position updating are as follows:

$$v_{i,j}^{(t+1)} = \psi \cdot v_{i,j}^{(t)} + c_1 \cdot r_1^{(t)} \cdot (BP_{i,j}^{(t)} - x_{i,j}^{(t)}) + c_2 \cdot r_2^{(t)} \cdot (BG_j^{(t)} - x_{i,j}^{(t)}), \quad (22)$$

$$x_{i,j}^{(t+1)} = x_{i,j}^{(t)} + v_{i,j}^{(t+1)}, \quad (23)$$

where ψ represents the inertia coefficient, t represents the current number of iterations, i represents the particle number, j represents the j -th dimension of the particle, c_1 and c_2 represent the acceleration coefficient, and $r_1^{(t)}$ and $r_2^{(t)}$ are random numbers distributed in $[0, 1]$. The right side of equation (22) consists of three parts: $\psi \cdot v_{i,j}^{(t)}$ is the momentum part of the particle, which reflects the inertia of the particle motion, and indicates that the particle has the tendency to maintain its previous velocity; $c_1 \cdot r_1^{(t)} \cdot (BP_{i,j}^{(t)} - x_{i,j}^{(t)})$ is the part of the particle's self-recognition, which reflects the memory of the particle's own historical

experience, and indicates that the particle has the tendency to approach its best historical position; $c_2 \cdot r_2^{(t)} \cdot (BG_j^{(t)} - x_{i,j}^{(t)})$ is the social consciousness part of the particle, which reflects the collective historical experience of cooperation and knowledge sharing among particles, and indicates that the particle tends to approach the best historical position of group or neighborhood.

4.3. Inertia Coefficient Setting. Inertia coefficient is one of the most important parameters in the particle swarm optimization algorithm. A bigger inertia coefficient can improve the global search ability of the algorithm and avoid premature convergence because of falling into local extremum. A smaller inertia coefficient can help to achieve accurate search in an area and improve convergence accuracy. In order to avoid falling into local extremum and improve the diversity of particles, the normal cloud model is introduced to improve the inertia and acceleration coefficients of the traditional particle swarm optimization algorithm, and a nonlinear inertia coefficient is set up to balance the global exploration and local development ability of the particle swarm optimization algorithm. Normal cloud model is a set of random numbers, which follow the normal distribution law and has a stable tendency. It is described by four main parameters: expected value ExpVal, particle entropy ParEnt, standard deviation StaDev, and hyperentropy HypEnt. The improved inertia coefficient is calculated as follows [34]:

$$\psi = \begin{cases} \eta_1 e^{-(f - \text{ExpVal})^2 / 2(\text{StaDev})^2}, & f \geq \bar{f}, \\ \eta_2 \left[\psi_{\max} - \frac{f - f_{\min}}{f_{\max} - f_{\min}} (\psi_{\max} - \psi_{\min}) \right], & f < \bar{f}, \end{cases} \quad (24)$$

where $\text{ExpVal} = \bar{f}$, in which \bar{f} is the average fitness value of particles in the particle swarm; $\text{StaDev} = r_3 \times \text{HypEnt} + \text{CloEnt}$, in which r_3 is a random number distributed in $[0, 1]$; $\text{ParEnt} = f_{\max} - \bar{f} / \tau_1$, in which τ_1 is the control coefficient of particle entropy; $\text{HypEnt} = \text{CloEnt} / \tau_2$, in which τ_2 is the control coefficient of hyperentropy; ψ_{\max} is the maximum inertia coefficient of the particle swarm; ψ_{\min} is the minimum inertia coefficient of the particle swarm; f_{\max} is the maximum fitness value of the particle swarm; f_{\min} is the minimum fitness value of the particle swarm; f is the fitness value of the particle; and η_1 and η_2 are the constants in $[0, 1]$ and can be set $\eta_1 = 0.4$ and $\eta_2 = 0.8$. From the analysis of equation (24), the inertia coefficient ψ has a larger value in the initial stage. With the increase in iterations, the inertia coefficient gradually decreases, which makes the algorithm change from global search in the initial stage to local fine search in the later stage.

The verticality of normal cloud model curve is controlled by particle entropy ParEnt. The average value of the normal cloud model is reflected in expected value ExpVal. The cloud particles in the model fluctuate around ExpVal, which shows the discreteness of cloud particles. The discreteness of cloud

particles is mainly determined by the hyperentropy HypEnt. The randomness of normal cloud model increases with the increase in HypEnt, and its stability increases with the decrease in HypEnt. With the help of the normal cloud model, the IPSOA algorithm improves the global search ability in the initial stage, restrains the premature convergence to form a more comprehensive solution space, and focuses on local fine search in the later stage of the algorithm.

4.4. Acceleration Coefficients Setting. In the iteration process of IPSOA algorithm, the acceleration coefficients c_1 and c_2 determine the influence of particle self-cognition and social cognition on particle trajectory, which reflects the information exchange degree between particles in swarms, and represent the acceleration weights of particles advancing towards their own extremum and global extremum, respectively. The formulas for calculating the acceleration coefficients are as follows [35]:

$$c_1 = c_{1,\text{start}} + (c_{1,\text{end}} - c_{1,\text{start}}) \sin(\psi), \quad (25)$$

$$c_2 = c_{2,\text{start}} - (c_{2,\text{start}} - c_{2,\text{end}}) \cos(\psi), \quad (26)$$

where c_1 is the self-acceleration coefficient, c_2 is the global acceleration coefficient, $c_{1,\text{start}}$ and $c_{2,\text{start}}$ are the initial values of acceleration coefficients c_1 and c_2 , and $c_{1,\text{end}}$ and $c_{2,\text{end}}$ are their termination values, respectively. Acceleration coefficients c_1 and c_2 are set according to equations (25) and (26). Setting larger global acceleration coefficient and smaller self-acceleration coefficient in the initial stage of the IPSOA algorithm, the social learning ability of the particle is stronger and the self-learning ability is weaker, which is beneficial to strengthening the global search ability; setting smaller global acceleration coefficient and larger self-acceleration coefficient at the later stage of the IPSOA algorithm, the self-learning ability of the particle is stronger and the social learning ability is weaker, which is beneficial to local fine search and converges to the global optimal solution with high precision.

4.5. Fitness Function Establishment. The multiple objectives of cloud manufacturing service composition interact with each other and are difficult to solve directly by general mathematical methods. However, cloud service users often have a clear understanding of the objectives and expectations of cloud manufacturing service composition. The purpose of cloud manufacturing service composition can be expressed as finding a cloud manufacturing service composition scheme which is as close as possible to the expectations of cloud service users under limited time, limited cost, and other conditions. The expectations of cloud service users are defined as the ideal point of the objective functions of cloud manufacturing service composition. IPSOA algorithm uses the ideal point method to design fitness function. The criterion for evaluating the effect of service composition scheme is the distance between ideal point and objective function values, i.e., deviation. Its calculation formula is as follows:

$$\text{Dev} = \sqrt{(Y_1^i - Y_1^*)^2 + (Y_2^i - Y_2^*)^2 + (Y_3^i - Y_3^*)^2 + (Y_4^i - Y_4^*)^2 + (Y_5^i - Y_5^*)^2}, \quad (27)$$

where $Y_1^i, Y_2^i, Y_3^i, Y_4^i,$ and Y_5^i are the five objective function values of cloud manufacturing service composition scheme corresponding to the i -th particle; $Y_1^*, Y_2^*, Y_3^*, Y_4^*,$ and Y_5^* are the optimal values of five objective functions, respectively, which constitute the ideal point; and Dev is the deviation.

Dimensionless treatment is done for five objective functions of cloud manufacturing service composition, so the deviation formula is changed to

$$\text{Dev}' = \sqrt{\left(\frac{Y_1^i - Y_1^*}{Y_1^*}\right)^2 + \left(\frac{Y_2^i - Y_2^*}{Y_2^*}\right)^2 + \left(\frac{Y_3^i - Y_3^*}{Y_3^*}\right)^2 + \left(\frac{Y_4^i - Y_4^*}{Y_4^*}\right)^2 + \left(\frac{Y_5^i - Y_5^*}{Y_5^*}\right)^2}, \quad (28)$$

where Dev' is called the relative deviation.

The smaller the deviation, the better the service composition scheme; on the contrary, the worse the service composition scheme. There are two main methods to determine the ideal point: one is to calculate the optimal values through single-objective function optimization algorithm to

form the ideal point, and another is to compose the ideal point by all objective function expected values specified by cloud service users. Weight coefficients are given according to the different importance of five objective functions. Thus, the fitness function of IPSOA algorithm can be designed as follows:

$$f(i) = \Gamma - \sqrt{\delta_1 \left(1 \frac{Y_1^i - Y_1^*}{Y_1^*}\right)^2 + \delta_2 \left(\frac{Y_2^i - Y_2^*}{Y_2^*}\right)^2 + \delta_3 \left(\frac{Y_3^i - Y_3^*}{Y_3^*}\right)^2 + \delta_4 \left(\frac{Y_4^i - Y_4^*}{Y_4^*}\right)^2 + \delta_5 \left(\frac{Y_5^i - Y_5^*}{Y_5^*}\right)^2}, \quad (29)$$

where $f(i)$ is the fitness function value of the i -th particle; Γ is a sufficiently large positive number; $\delta_1, \delta_2, \delta_3, \delta_4,$ and δ_5 are the weight coefficients of the five objective functions respectively; and $\sum_{k=1}^5 \delta_k = 1$.

4.6. IPSOA Algorithm Steps. The main steps of IPSOA algorithm are as follows:

Step 1: initialize particle swarm, including particle swarm size ParSiz , position x_i , and velocity v_i of each particle; set the initial and end values of acceleration coefficients c_1 and c_2 , maximum evolutionary generation MaxGen , and randomly initialize the initial position and initial velocity of each particle in the definition domain. The initial position of the particle is set as the initial individual optimal position, and the position of the particle with the best fitness in the particle swarm is set as the initial global optimal position.

Step 2: calculate service matching degree, composition synergy degree, cloud entropy, execution time, and execution cost.

Step 3: calculate the fitness value of all the particles in the particle swarm.

Step 4: for each particle, compare its fitness value $f(i)$ with individual extremum $f(\text{BP}_i)$. If $f(i) > f(\text{BP}_i)$, then replace $f(\text{BP}_i)$ with $f(i)$ and BP_i with P_i .

Step 5: for each particle, compare its fitness value $f(i)$ with global extremum $f(\text{BG})$. If $f(i) > f(\text{BG})$, then replace $f(\text{BG})$ with $f(i)$ and BG with P_i .

Step 6: update the inertia coefficient of particles according to equation (24); update the self-acceleration coefficient and global acceleration coefficient of particles according to equations (25) and (26); update the velocity and position of particles according to equations (22) and (23).

Step 7: check the termination condition of the algorithm. If the evolutionary generation reaches the set maximum value or other algorithm end conditions are met, stop the iterative operation of the algorithm and output the results; otherwise, return to step 2.

IPSOA is a parallel algorithm. All the particles can be divided into several groups [36]. Each group solves the optimization problem separately, and each group is computed by one core of the multicore CPU, i.e., parallel computing, to achieve concurrent solutions.

5. Application Example

Taking producing automatic guided forklift (AGF) by using cloud manufacturing services as an example, the proposed optimization algorithm for cloud manufacturing service composition is applied.

The AGF manufacturing task can be divided into seven subtasks: J_1 , body production; J_2 , driving device production; J_3 , working device production; J_4 , power supply system production; J_5 , auxiliary control system production; J_6 , main control system production; and J_7 , painting and packaging. According to the proposed IPSOA algorithm, the above manufacturing tasks are matched with cloud services, as shown in Table 2. The cloud service sets for manufacturing tasks $J_1, J_2, J_3, J_4, J_5, J_6$, and J_7 are $S_1, S_2, S_3, S_4, S_5, S_6$, and S_7 , respectively. The number of cloud services included in each set is 3, 2, 2, 3, 2, 4, and 2, respectively. TF, HF, and DF represent technical factor, hunger factor, and distance factor. T_{exe} , T_{con} , and T_{rep} are used for execution time (hour), maximum continuous working time (hour), and repair time (hour), respectively. w denotes the unit time cost (dollar/hour). Taking $\alpha=0.4$, $\beta=0.3$, and $\gamma=0.3$, the service matching degree MD can be calculated according to equation (3). The cloud entropy CE can be calculated according to equation (8).

According to equation (6), the service composition synergy degree matrix of the AGF manufacturing task can be calculated as follows:

$$SD = \begin{pmatrix} SD_{1,1} & SD_{1,2} & \cdots & SD_{1,6} & SD_{1,7} \\ SD_{2,1} & \cdots & \cdots & \cdots & SD_{2,7} \\ \cdots & \cdots & SD_{4,4} & \cdots & \cdots \\ SD_{6,1} & \cdots & \cdots & \cdots & SD_{6,7} \\ SD_{7,1} & SD_{7,2} & \cdots & SD_{7,6} & SD_{7,7} \end{pmatrix}, \quad (30)$$

where the diagonal elements of the matrix SD are all 1 and SD_{12} is a 3×2 matrix, which represents the composition synergy degree of three candidate cloud manufacturing services of the manufacturing task J_1 to two candidate cloud manufacturing services of the manufacturing task J_2 . The meaning of other elements is the same. All elements of matrix SD are calculated, as shown in Table 3.

The improved particle swarm optimization algorithm is programmed by using Matlab R2015a. The deadline constraint ET_0 is 480, and the cost constraint EC_0 is 18000. The particle swarm size is $ParSiz = 30$. The maximum iteration generation is $MaxGen = 120$. Taking $\Gamma = 100$, all the weight coefficients of the five objective functions are set to 0.2, i.e., $\delta_k = 0.2$ ($k = 1, 2, 3, 4, 5$). According to the single-objective function optimization algorithm, the ideal point can be obtained, i.e., (5.15, 19.035, 7.317, 406, 13608). The space boundaries for total execution time ET_{sum} are [0, 490]. The bounds for total execution cost EC_{sum} are [0, 18500]. The bounds for total service matching degree MD_{sum} are [0, 10]. The bounds for total composition synergy degree SD_{sum} are [0, 20]. The bounds for total cloud entropy CE_{sum} are [0, 20]. After 49 iterations, the optimal fitness value of the particle swarm is 99.928, the corresponding point ($MD_{sum}, SD_{sum}, CE_{sum}, ET_{sum}, EC_{sum}$) is (4.47, 18.142, 7.887, 406, 13671), and the particle code of the optimal service composition scheme is 2112132. The relative deviation between the optimal solution and the ideal point is 0.160. As shown in Figure 5, the particle code represents the following meanings: the manufacturing task J_1 is allocated to the 2-nd cloud service

in cloud service set S_1 , task J_2 to the 1-st cloud service in cloud service set S_2 , task J_3 to the 1-st cloud service in cloud service set S_3 , task J_4 to the 2-nd cloud service in cloud service set S_4 , task J_5 to the 1-st cloud service in cloud service set S_5 , task J_6 to the 3-rd cloud service in cloud service set S_6 , and task J_7 to the 2-nd cloud service in cloud service set S_7 . The average running time of the algorithm is 11.56 s, and the iteration curves of its related parameters are shown in Figure 6. Figure 6(a) shows the iteration curve of the optimal particle fitness, Figure 6(b) shows the iteration curve of the service matching degree, Figure 6(c) shows the iteration curve of the composition synergy degree, Figure 6(d) shows the iteration curve of the cloud entropy, Figure 6(e) shows the iteration curve of service composition execution time, Figure 6(f) shows the iteration curve of service composition execution cost, and Figure 6(g) shows the iteration curve of relative deviation. The particle swarm tends to be stable when it iterates to the 49-th generation.

Given the same maximum iteration generation and particle swarm size, IPSOA algorithm, standard genetic algorithms (SGA) [37], and traditional particle swarm optimization algorithm (PSO) [35] are used to solve the same service composition optimization problem, respectively. As shown in Figure 7, IPSOA converges to the optimal solution in the 49-th generation, SGA converges in the 82-nd generation, and PSO converges to the optimal solution in the 54-th generation. The algorithms run on a portable computer with Intel core i3-3110M CPU, 2.4 GHz main frequency, and 4 G memory. IPSOA takes 11.56 s, SGA 19.05 s, and PSO 12.94 s, as shown in Table 4. The relative deviation of IPSOA is 0.160, that of SGA is also 0.160, and that of PSO is 0.183. The above case analysis and experimental results show that IPSOA has faster convergence speed and shorter solution time than PSO and SGA for multiobjective optimization of cloud manufacturing service composition, and IPSOA has better performance than PSO.

In the process of using IPSOA to optimize GAF cloud manufacturing service composition, service matching degree, composition synergy degree, cloud entropy, execution time, and execution cost are used as the five main variables of the optimization model. The influence of technical factor, resources vacancy rate, distance factor, multiple services collaboration level, coupling relationship, complexity, reliability, time, and cost of service composition are fully considered. As shown in Table 5, the service composition scheme obtained by IPSOA has better comprehensive characteristics than the general methods. Maximum matching degree method (MaxMDM) service composition scheme has the best service matching degree, but it has worse cloud entropy, execution time, execution cost, and relative deviation. Maximum synergy degree method (MaxSDM) service composition scheme has the best composition synergy degree, but it has worse cloud entropy, execution time, execution cost, and relative deviation. Minimum cloud entropy method (MinCEM) service composition scheme has minimum cloud entropy, but its service matching degree and composition synergy degree are smaller than that of IPSOA, and it has worse execution time, execution cost, and relative deviation. Minimum execution time method (MinETM) service composition scheme has the minimum execution time, but with the increase in the

TABLE 2: Matching table of cloud services and manufacturing tasks.

J_j	J_1			J_2		J_3		J_4			J_5		J_6			J_7		
S_j	S_1			S_2		S_3		S_4			S_5		S_6			S_7		
S_{ij}	$S_{1,1}$	$S_{2,1}$	$S_{3,1}$	$S_{1,2}$	$S_{2,2}$	$S_{1,3}$	$S_{2,3}$	$S_{1,4}$	$S_{2,4}$	$S_{3,4}$	$S_{1,5}$	$S_{2,5}$	$S_{1,6}$	$S_{2,6}$	$S_{3,6}$	$S_{4,6}$	$S_{1,7}$	$S_{2,7}$
TF	0.8	0.2	0.2	0.6	0.2	0.4	0.2	0.8	0.2	0.4	0.6	0.6	0.2	0.4	0.8	0.4	0.6	0.8
HF	0.9	0.5	0.8	0.7	0.4	0.6	0.3	0.5	0.4	0.8	0.3	0.9	0.6	0.7	0.5	0.9	0.9	0.5
DF	0.8	1	0.4	1	0.8	0.8	1	0.4	0.8	1	1	0.8	0.4	0.8	1	0.4	0.8	1
T_{exe}	79	46	52	48	58	49	60	75	63	72	55	62	77	68	62	76	90	83
T_{con}	30	24	24	30	40	20	19	24	20	16	30	24	30	24	16	20	24	30
T_{rep}	3	4	1	2	1	4	2	3	2	1	1	2	2	2	3	1	2	1
w	49	40	45	31	36	31	39	42	51	50	35	40	35	35	30	32	25	22
MD	0.83	0.53	0.44	0.75	0.44	0.58	0.47	0.59	0.44	0.7	0.63	0.75	0.38	0.61	0.77	0.55	0.75	0.77
CE	1.281	0.919	0.991	0.792	0.689	1.220	1.316	1.343	1.309	1.735	0.765	1.251	1.224	1.283	1.691	1.527	1.580	1.190

TABLE 3: Computation results of composition synergy degree.

J_j	J_1			J_2		J_3		J_4			J_5		J_6			J_7		
S_j	S_1			S_2		S_3		S_4			S_5		S_6			S_7		
SD	$S_{1,1}$	$S_{2,1}$	$S_{3,1}$	$S_{1,2}$	$S_{2,2}$	$S_{1,3}$	$S_{2,3}$	$S_{1,4}$	$S_{2,4}$	$S_{3,4}$	$S_{1,5}$	$S_{2,5}$	$S_{1,6}$	$S_{2,6}$	$S_{3,6}$	$S_{4,6}$	$S_{1,7}$	$S_{2,7}$
$S_{1,1}$	1.000	1.000	1.000	0.838	0.771	0.774	0.835	0.588	0.770	0.833	0.836	0.771	0.588	0.770	0.834	0.588	0.770	0.833
$S_{2,1}$	1.000	1.000	1.000	0.909	0.834	0.833	0.910	0.534	0.835	0.911	0.909	0.835	0.534	0.836	0.910	0.534	0.841	0.913
$S_{3,1}$	1.000	1.000	1.000	0.527	0.589	0.588	0.527	0.670	0.589	0.530	0.526	0.589	0.671	0.590	0.527	0.671	0.597	0.533
$S_{1,2}$	0.838	0.909	0.527	1.000	1.000	0.833	0.910	0.532	0.835	0.911	0.909	0.834	0.533	0.835	0.834	0.533	0.840	0.912
$S_{2,2}$	0.771	0.834	0.589	1.000	1.000	0.770	0.833	0.590	0.769	0.834	0.833	0.769	0.591	0.770	0.769	0.590	0.773	0.836
$S_{1,3}$	0.774	0.833	0.588	0.833	0.770	1.000	1.000	0.594	0.771	0.836	0.834	0.770	0.594	0.772	0.834	0.594	0.777	0.838
$S_{2,3}$	0.835	0.910	0.527	0.910	0.833	1.000	1.000	0.528	0.833	0.909	0.909	0.833	0.528	0.834	0.909	0.528	0.836	0.910
$S_{1,4}$	0.588	0.534	0.670	0.532	0.590	0.594	0.528	1.000	1.000	1.000	0.529	0.589	0.667	0.589	0.527	0.667	0.589	0.527
$S_{2,4}$	0.770	0.835	0.589	0.835	0.769	0.771	0.833	1.000	1.000	1.000	0.834	0.769	0.589	0.769	0.833	0.589	0.772	0.835
$S_{3,4}$	0.833	0.911	0.530	0.911	0.834	0.836	0.909	1.000	1.000	1.000	0.910	0.834	0.526	0.833	0.909	0.526	0.834	0.909
$S_{1,5}$	0.836	0.909	0.526	0.909	0.833	0.834	0.909	0.529	0.834	0.910	1.000	1.000	0.530	0.834	0.909	0.530	0.837	0.911
$S_{2,5}$	0.771	0.835	0.589	0.834	0.769	0.770	0.833	0.589	0.769	0.834	1.000	1.000	0.590	0.769	0.833	0.589	0.772	0.835
$S_{1,6}$	0.588	0.534	0.671	0.533	0.591	0.594	0.528	0.667	0.589	0.526	0.530	0.590	1.000	1.000	1.000	1.000	0.589	0.526
$S_{2,6}$	0.770	0.836	0.590	0.835	0.770	0.772	0.834	0.589	0.769	0.833	0.834	0.769	1.000	1.000	1.000	1.000	0.771	0.834
$S_{3,6}$	0.834	0.910	0.527	0.834	0.769	0.834	0.909	0.527	0.833	0.909	0.909	0.833	1.000	1.000	1.000	1.000	0.836	0.910
$S_{4,6}$	0.588	0.534	0.671	0.533	0.590	0.594	0.528	0.667	0.589	0.526	0.530	0.589	1.000	1.000	1.000	1.000	0.589	0.527
$S_{1,7}$	0.770	0.841	0.597	0.840	0.773	0.777	0.836	0.589	0.772	0.834	0.837	0.772	0.589	0.771	0.836	0.589	1.000	1.000
$S_{2,7}$	0.833	0.913	0.533	0.912	0.836	0.838	0.910	0.527	0.835	0.909	0.911	0.835	0.526	0.834	0.910	0.527	1.000	1.000

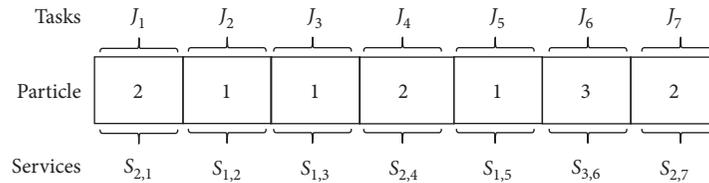


FIGURE 5: The optimal solution of the AGF manufacturing task.

weight coefficients of service matching degree, composition synergy degree, and cloud entropy, the relative deviation value of MinETM will be bigger than that of IPSOA, and its comprehensive service quality will become worse. Minimum execution cost method (MinECM) service composition scheme has the minimum execution cost, but its service matching degree and composition synergy degree are smaller than that of IPSOA, and it has worse execution time and relative deviation. Compared with the five service composition schemes, IPSOA has the smallest relative deviation and the best comprehensive performance, which helps users make

more reasonable decisions. If only the execution cost and execution time are considered in the service composition and the influence of service matching degree, combination coordination degree, and combination entropy are not considered, it may lead to user's wrong decision and bring adverse effects to manufacturing industry, such as the shortcomings of MinETM and MinECM. Because the service matching degree and composition synergy degree of the two service composition schemes are very small, the manufacturing services selected from the two service composition schemes may have problems of poor service quality and low service reliability,

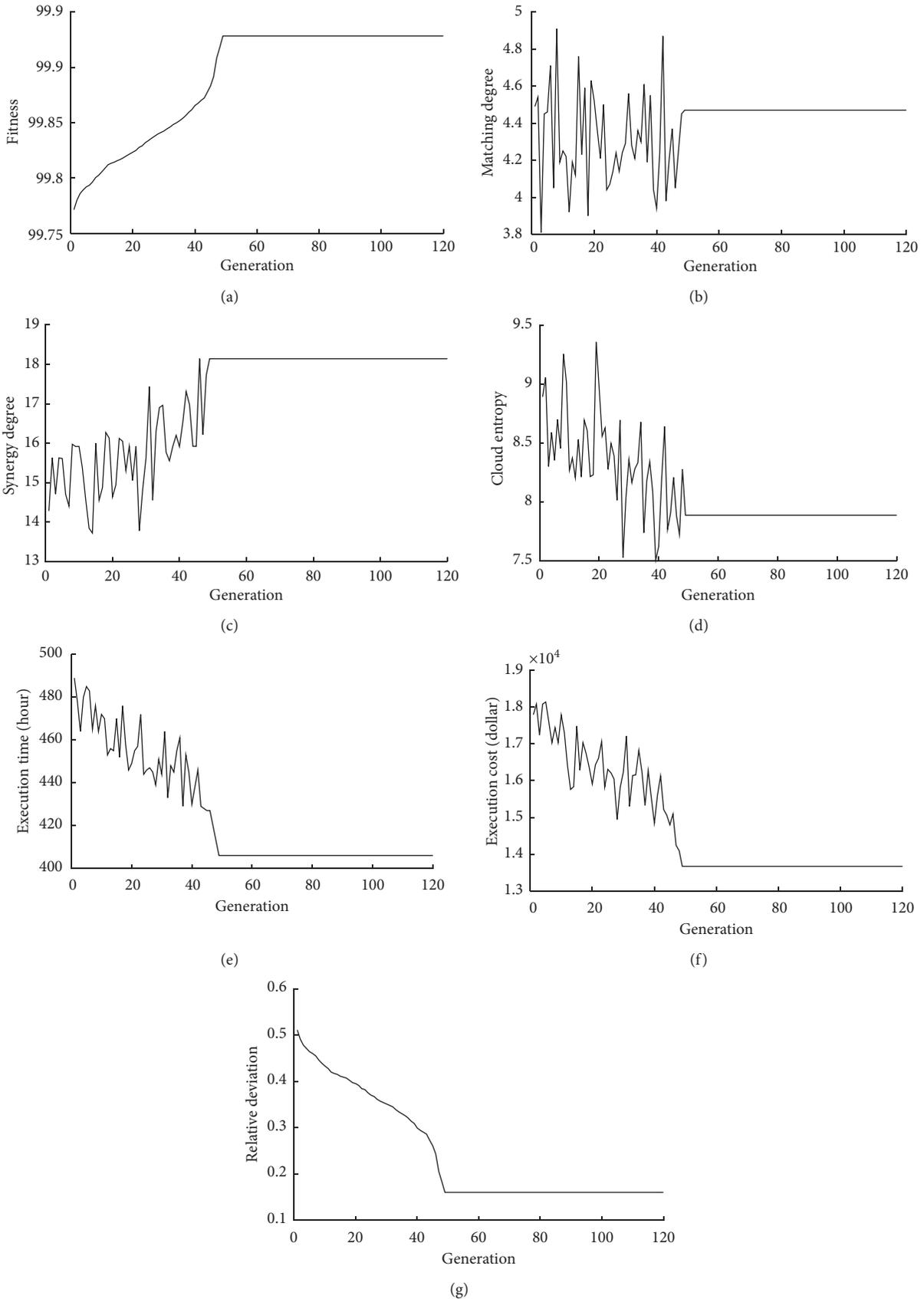


FIGURE 6: IPSOA iteration curves: (a) iteration curve of the optimal particle fitness; (b) iteration curve of service matching degree; (c) iteration curve of composition synergy degree; (d) iteration curve of cloud entropy; (e) iteration curve of execution time; (f) iteration curve of execution cost; (g) iteration curve of relative deviation.

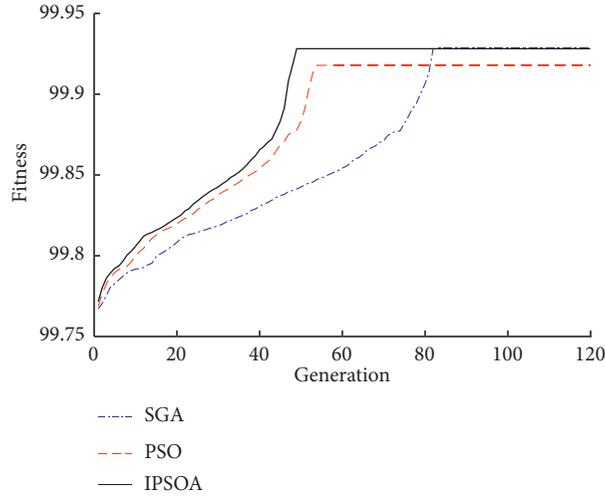


FIGURE 7: Comparison of SGA, PSO, and IPSOA iteration curves.

TABLE 4: Optimization results for the AGF manufacturing task.

Algorithms	Computing time (s)	Convergence generation	Optimization results	Relative deviation
SGA	19.05	82	$\{S_{2,1}, S_{1,2}, S_{1,3}, S_{2,4}, S_{1,5}, S_{3,6}, S_{2,7}\}$	0.160
PSO	12.94	54	$\{S_{2,1}, S_{1,2}, S_{1,3}, S_{2,4}, S_{1,5}, S_{2,6}, S_{2,7}\}$	0.183
IPSOA	11.56	49	$\{S_{2,1}, S_{1,2}, S_{1,3}, S_{2,4}, S_{1,5}, S_{3,6}, S_{2,7}\}$	0.160

TABLE 5: Comparison of different factors' influences on service composition.

Algorithms	Optimization results	MD _{sum}	SD _{sum}	CE _{sum}	ET _{sum}	EC _{sum}	Relative deviation
IPSOA	$\{S_{2,1}, S_{1,2}, S_{1,3}, S_{2,4}, S_{1,5}, S_{3,6}, S_{2,7}\}$	4.47	18.142	7.887	406	13671	0.160
MaxMDM	$\{S_{1,1}, S_{1,2}, S_{1,3}, S_{3,4}, S_{1,5}, S_{3,6}, S_{2,7}\}$	5.15	18.150	8.675	448	16089	0.284
MaxSDM	$\{S_{2,1}, S_{1,2}, S_{2,3}, S_{3,4}, S_{1,5}, S_{3,6}, S_{2,7}\}$	4.62	19.035	8.409	426	14879	0.210
MinCEM	$\{S_{2,1}, S_{2,2}, S_{1,3}, S_{2,4}, S_{1,5}, S_{1,6}, S_{2,7}\}$	3.77	15.919	7.317	431	15106	0.338
MinETM	$\{S_{2,1}, S_{1,2}, S_{1,3}, S_{2,4}, S_{1,5}, S_{3,6}, S_{2,7}\}$	4.47	18.142	7.887	406	13671	0.160
MinECM	$\{S_{2,1}, S_{1,2}, S_{1,3}, S_{1,4}, S_{1,5}, S_{3,6}, S_{2,7}\}$	4.62	16.443	7.921	418	13608	0.192

which will affect the supply chain, product quality, delivery time, and even the survival of manufacturing enterprises. For example, when multinational manufacturing enterprises such as ZTE and Huawei purchase parts globally based on cloud services, they need to consider factors such as service matching degree, composition synergy degree, and cloud entropy, rather than just considering execution time and execution cost. Unstable cooperation and low service reliability may hinder the normal production of products and threaten the survival of enterprises. The impact of COVID-19 has increased the weight of service matching degree, composition synergy degree, and cloud entropy in the service composition optimization model. Enterprises need to pay more attention to these aspects when making decisions from 2020 and make appropriate adjustments to adapt to the rapidly changing global manufacturing environment.

6. Conclusion

In order to solve the problems of low search efficiency and inaccurate optimization in existing service composition optimization methods, the multiobjective optimization of

cloud manufacturing service composition is discussed, a new improved particle swarm algorithm is proposed, and comprehensive service quality evaluation method is studied. The main work and contributions are summarized as follows:

- (1) The main factors affecting the performance of cloud manufacturing service composition are studied. A new service quality model of cloud manufacturing service composition is constructed by combining the three new attributes of cloud entropy, service matching degree, and composition synergy degree with two traditional attributes of execution time and execution cost, which evaluates the service composition performance more comprehensively.
- (2) The mathematical model of cloud manufacturing service composition optimization is established. Cloud entropy, service matching degree, composition synergy degree, execution time, and execution cost are taken as five objective functions, and particle fitness function is constructed by the ideal point method, which provides a multiobjective

optimization solution for cloud manufacturing service composition optimization problem. It is helpful for enterprises to make more reasonable decisions and adapt to the unstable global manufacturing environment. Especially under the influence of COVID-19, enterprises should consider more unstable manufacturing factors in cloud manufacturing activities, and the weight coefficients of service matching degree, composition synergy degree, and cloud entropy should be given bigger values in the service composition optimization model.

- (3) An improved particle swarm optimization algorithm IPSOA is proposed. The inertia and acceleration coefficients of the algorithm are improved by introducing the normal cloud model, sine function, and cosine function. It improves the global search ability in the initial stage of the algorithm, restrains the premature convergence of the algorithm in order to form a more comprehensive solution space, and makes the algorithm focus on local fine search in the later stage, so as to improve the optimization precision and efficiency.
- (4) Taking the AGF manufacturing task as an example, the correctness of the multiobjective service composition optimization mathematical model and the feasibility and effectiveness of the IPSOA algorithm are verified. Case study shows that, compared with the PSO algorithm and SGA algorithm, the IPSOA algorithm has better performance, faster convergence speed, and shorter solving time for multiobjective optimization problem of cloud manufacturing service composition.

The development of cloud manufacturing and other related technologies impels manufacturing enterprises to develop from traditional large-scale flow line production mode to multibatch customization production, from production-oriented to production-service-oriented, and activates all kinds of idle manufacturing resources around the world. Service quantification, service granularity, cloud entropy, matching degree, and synergy degree affect the future development of cloud manufacturing and will be studied in depth in future work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The project was supported by the National Natural Science Foundation of China (nos. 51375168 and 5181102248), the GUES Scientific Research Foundation for Advanced Talents (no. YKHZ G2018009), and the Science and Technology

Foundation of Guizhou Province (nos. QKHLHZ [2014] 7528 and QJHKYZ [2019]158).

References

- [1] B. H. Li, L. Zhang, S. L. Wang et al., "Cloud manufacturing: a new service-oriented networked manufacturing model," *Computer Integrated Manufacturing Systems*, vol. 1, pp. 1–7, 2010.
- [2] F. V. Omid and H. Mahmoud, "A platform for optimization in distributed manufacturing enterprises based on cloud manufacturing paradigm," *International Journal of Computer Integrated Manufacturing*, vol. 11, pp. 1031–1054, 2014.
- [3] M. Yuan, K. Deng, and W. A. Chaovaitwongse, "Manufacturing resource modeling for cloud manufacturing," *International Journal of Intelligent Systems*, vol. 32, no. 4, pp. 414–436, 2017.
- [4] J. Lartigau, X. Xu, L. Nie, and D. Zhan, "Cloud manufacturing service composition based on QoS with geo-perspective transportation using an improved Artificial Bee Colony optimisation algorithm," *International Journal of Production Research*, vol. 53, no. 14, pp. 4380–4404, 2015.
- [5] J. Thekinen and J. H. Panchal, "Resource allocation in cloud-based design and manufacturing: a mechanism design approach," *Journal of Manufacturing Systems*, vol. 43, pp. 327–338, 2017.
- [6] M. Lang, M. Wiesche, and H. Krcmar, "Criteria for selecting cloud service providers: a delphi study of quality-of-service attributes," *Information & Management*, vol. 55, no. 6, pp. 746–758, 2018.
- [7] S. Răileanu, F. Anton, T. Borangiu et al., "A cloud-based manufacturing control system with data integration from multiple autonomous agents," *Computers in Industry*, vol. 11, pp. 50–61, 2018.
- [8] P. Helo and Y. Hao, "Cloud manufacturing system for sheet metal processing," *Production Planning & Control*, vol. 28, no. 6–8, pp. 524–537, 2017.
- [9] T. Chen and Y.-C. Wang, "Estimating simulation workload in cloud manufacturing using a classifying artificial neural network ensemble approach," *Robotics and Computer-Integrated Manufacturing*, vol. 38, pp. 42–51, 2016.
- [10] M. R. Namjoo and A. Keramati, "Analysing causal dependencies of composite service resilience in cloud manufacturing using resource-based theory and dematel method," *International Journal of Computer Integrated Manufacturing*, vol. 4, pp. 1–19, 2018.
- [11] V. B. Souza, X. Masip-Bruin, E. Marín-Tordera et al., "Towards a proper service placement in combined Fog-to-Cloud (F2C) architectures," *Future Generation Computer Systems*, vol. 87, pp. 1–15, 2018.
- [12] A. Brant and M. M. Sundaram, "A novel system for cloud-based micro additive manufacturing of metal structures," *Journal of Manufacturing Processes*, vol. 20, pp. 478–484, 2015.
- [13] S. Wang, W. Song, L. Kang et al., "Manufacturing resource allocation based on cloud manufacturing," *Computer Integrated Manufacturing Systems*, vol. 7, pp. 1396–1405, 2012.
- [14] J. Zhou and X. Yao, "Multi-population parallel self-adaptive differential artificial bee colony algorithm with application in large-scale service composition for cloud manufacturing," *Applied Soft Computing*, vol. 56, pp. 379–397, 2017.
- [15] C. Li, J. Guan, T. Liu et al., "An autonomy-oriented method for service composition and optimal selection in cloud manufacturing," *International Journal of Advanced Manufacturing Technology*, vol. 3, pp. 1–22, 2018.

- [16] Y. X. Li and X. F. Yao, "Cloud manufacturing service composition and formal verification based on extended process calculus," *Advances in Mechanical Engineering*, vol. 6, pp. 1–16, 2018.
- [17] F. Tao, J. Cheng, Y. Cheng, S. Gu, T. Zheng, and H. Yang, "SDMSim: a manufacturing service supply-demand matching simulator under cloud environment," *Robotics and Computer-Integrated Manufacturing*, vol. 45, pp. 34–46, 2017.
- [18] S. Zhang, Y. Xu, W. Zhang et al., "A new fuzzy QoS-aware manufacture service composition method using extended flower pollination algorithm," *Journal of Intelligent Manufacturing*, vol. 4, pp. 1–15, 2017.
- [19] C. Yang, W. Shen, T. Lin, and X. Wang, "IoT-enabled dynamic service selection across multiple manufacturing clouds," *Manufacturing Letters*, vol. 7, pp. 22–25, 2016.
- [20] F. Chen, R. Dou, M. Li, and H. Wu, "A flexible QoS-aware web service composition method by multi-objective optimization in cloud manufacturing," *Computers & Industrial Engineering*, vol. 99, pp. 423–431, 2016.
- [21] Y. Liu, X. Xu, L. Zhang, L. Wang, and R. Y. Zhong, "Workload-based multi-task scheduling in cloud manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 45, pp. 3–20, 2017.
- [22] H. Bouzary and F. Frank Chen, "Service optimal selection and composition in cloud manufacturing: a comprehensive survey," *The International Journal of Advanced Manufacturing Technology*, vol. 97, no. 1-4, pp. 795–808, 2018.
- [23] Y. Que, W. Zhong, H. Chen et al., "Improved adaptive immune genetic algorithm for optimal QoS-aware service composition selection in cloud manufacturing," *International Journal of Advanced Manufacturing Technology*, vol. 10, pp. 1–11, 2018.
- [24] X. Huang, Z. Guan, and L. Yang, "An effective hybrid algorithm for multi-objective flexible job-shop scheduling problem," *Advances in Mechanical Engineering*, vol. 9, pp. 1–14, 2018.
- [25] F. Zhao, W. Lei, W. Ma, Y. Liu, and C. Zhang, "An improved SPEA2 algorithm with adaptive selection of evolutionary operators scheme for multiobjective optimization problems," *Mathematical Problems in Engineering*, vol. 2016, pp. 1–20, 2016.
- [26] W. Zhang, S. Zhang, S. Guo et al., "Concurrent optimal allocation of distributed manufacturing resources using extended teaching-learning-based optimization," *International Journal of Production Research*, vol. 3, pp. 1–18, 2016.
- [27] B. Xu and Y. Cai, "A multiple-data-based efficient global optimization algorithm and its parallel implementation for automotive body design," *Advances in Mechanical Engineering*, vol. 8, pp. 1–13, 2018.
- [28] Y. Wang, Z. Dai, W. Zhang et al., "Urgent task-aware cloud manufacturing service composition using two-stage biogeography-based optimization," *International Journal of Computer Integrated Manufacturing*, vol. 10, pp. 1–14, 2018.
- [29] W. Liu, Y. Li, and B. Liu, "Service composition in cloud manufacturing based on adaptive mutation particle swarm optimization," *Journal of Computer Applications*, vol. 10, pp. 2869–2874, 2018.
- [30] Y. X. Li, X. F. Yao, C. Xu et al., "Cloud manufacturing service composition modeling and QoS evaluation based on extended process calculus," *Computer Integrated Manufacturing Systems*, vol. 3, pp. 689–700, 2014.
- [31] Y. Li, X. Yao, and M. Liu, "Cloud manufacturing service composition optimization with improved genetic algorithm," *Mathematical Problems in Engineering*, vol. 2019, Article ID 7194258, 19 pages, 2019.
- [32] Y. Li, X. Yao, and J. Zhou, "Multi-objective optimization of cloud manufacturing service composition with cloud-entropy enhanced genetic algorithm," *Strojniški Vestnik-Journal of Mechanical Engineering*, vol. 62, no. 10, pp. 577–590, 2016.
- [33] Y. J. Chen, X. F. Yao, and D. L. Xu, "Job shop scheduling with profit and entropy as performance measures," *Journal of Beijing University of Technology*, vol. 10, pp. 1305–1311, 2010.
- [34] C. H. Dai, Y. F. Zhu, W. R. Chen et al., "Cloud model based genetic algorithm and its application," *Acta Electronic Sinica*, vol. 7, pp. 1419–1424, 2007.
- [35] M. Guo, R. Li, and L. Liu, "Particle swarm optimization algorithm of learning factors and time factor adjusting to weights," *Application Research of Computers*, vol. 11, pp. 3291–3294, 2014.
- [36] N. Ghadimi, M. Afkousi-Paqaleh, and A. Nouri, "PSO based fuzzy stochastic long-term model for deployment of distributed energy resources in distribution systems with several objectives," *IEEE Systems Journal*, vol. 7, no. 4, pp. 786–796, 2013.
- [37] Y. J. Huang, X. F. Yao, D. Y. Ge, and Y. X. Li, "Entropy-enhanced genetic algorithm with tabu search for job shop scheduling problems," *Advanced Materials Research*, vol. 590, pp. 557–562, 2012.