



An optimization method of cloud manufacturing service composition based on matching-collaboration degree

Chao Yin^{1,2} · Shanglin Li^{1,2} · Xiaobin Li^{1,2}

Received: 9 February 2023 / Accepted: 21 January 2024 / Published online: 1 February 2024
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2024

Abstract

Traditional service combination methods in the cloud manufacturing paradigm mainly focus on economic targets, such as time and cost and ignore the matching and collaboration effects between cloud services and manufacturing tasks, resulting in the constructed service combination solutions not fully meeting the individual requirements of users. In this paper, the concepts of service matching degree and service collaboration degree are proposed, and an evaluation system of cloud manufacturing service composition is established to measure the quality of cloud services. Then a double-constraint service composition optimization model is designed considering the interests of both manufacturing service requestors and resource providers, which is solved by using the improved ant colony algorithm (IACO). Finally, an automobile bumper cloud manufacturing case is carried out to demonstrate the feasibility and effectiveness of the proposed method.

Keywords Cloud manufacturing · Cloud service composition · Service matching degree · Service collaboration degree · Improved ant colony algorithm

1 Introduction

With the in-depth applications of cloud computing [1], Internet of Things [2], artificial intelligence, and other new-generation information technologies in the manufacturing industry, many manufacturing enterprises are undergoing significant changes in operating models, manufacturing approaches, and service methods. In this context, mass customization and networked collaboration-based manufacturing has become a mainstream trend. Cloud manufacturing

[3], a service-oriented intelligent manufacturing model integrating the above technologies, is gaining widespread attention from various discrete manufacturing enterprises, especially the large number of small and medium-sized enterprises. In the cloud model, some businesses can register their redundant manufacturing resources and service capabilities on the cloud service platform to conveniently provide various specialized manufacturing services, such as machine tool leasing, networked outsourcing processing, and remote equipment maintenance. Other enterprises, that lack manufacturing resources and capabilities, can inexpensively subscribe to these manufacturing services through the cloud platform to solve production challenges and enhance their overall competitiveness [4, 5]. Cloud manufacturing model breaks down the distance limitation and information gap between manufacturing enterprises, and enables them to carry out all kinds of web-based collaboration manufacturing services anytime, anywhere, and on-demand, which can promote the transformation of traditional manufacturing enterprises from production-oriented manufacturing to service-oriented manufacturing, and increase the core value of the entire manufacturing industry chain.

In cloud manufacturing environment, personalized manufacturing requirements from users or enterprises are often characterized by complexity, diversity, and ambiguity, which

Chao Yin and Shanglin Li contributed equally to this work.

✉ Xiaobin Li
xiaobin_lee@cqu.edu.cn

Chao Yin
ych925@cqu.edu.cn

Shanglin Li
657219983@qq.com

¹ College of Mechanical and Vehicle Engineering, Chongqing University, Shazheng Street, Shapingba District, Chongqing 400044, China

² State Key Laboratory of Mechanical Transmissions, Chongqing University, Shazheng Street, Shapingba District, Chongqing 400044, China

are often difficult to be met by a single, simple cloud service. Therefore, for the complex customized manufacturing tasks submitted by service requestors, the cloud platform needs to invoke decomposition tools to separate tasks into a series of fine-grained sub-tasks according to corresponding manufacturing processes, and select optimal cloud manufacturing resources and services to execute these sub-tasks to meet corresponding manufacturing requirements [6]. As one of key enabling technologies to improve the utilization and added value of manufacturing resources in cloud, manufacturing service combination plays an important role in the cloud manufacturing tasks implementation. In recent years, much research on manufacturing service combination methodologies has been carried out by domestic and international academics. In terms of the evaluation for the manufacturing service portfolio optimization, in addition to traditional metrics such as time, cost, and quality [7–10], Zhou et al. [11] considered combinability among manufacturing services and the reputation of service providers. Li and Yang et al. [12, 13] enriched the indicator system with delivery time, delay time, and energy consumption of manufacturing services from the perspective of sustainable manufacturing perspective, making the chosen services more relevant to real manufacturing situations. Although most of the above findings focus on a multidimensional evaluation of the cloud services supported by manufacturing resources, and calculate the overall operational quality of each manufacturing service through constructing quantitative mathematical models. However, the matching degree between cloud services and manufacturing tasks, as well as the collaboration degree among cloud services have not been considered, which makes it difficult for the selected cloud service combination to complete manufacturing tasks efficiently in cloud manufacturing environment. In addition, the evaluation models constructed above take into account the interests of manufacturing service requestors, but do not weigh their interests from the perspective of manufacturing resource providers. As a result, the cooperation between the supply and demand sides is not sustainable, which restricts the effective operation and long-term development of the cloud service platform. In light of the aforementioned, a matching-collaboration degree-based optimization method of cloud manufacturing service composition is proposed to address the above issues in this paper, and the main contributions of this paper are as follows:

1. The concepts of service matching degree (S_m) and service collaboration degree (CS) are proposed, and an evaluation metric system of cloud manufacturing service composition, including service time (St), service cost (Sc), service reliability (St), S_m , and CS , is established.
2. A double-constraint composition optimization model for cloud manufacturing services is designed in which the

interests of both manufacturing service requestors and manufacturing resource providers are considered comprehensively.

3. An improved ant colony algorithm (IACO) for cloud manufacturing service composition is proposed, and an application example is given to verify the effectiveness of the proposed method.

The remainder of this paper is organized as follows: The related work on cloud manufacturing service composition is introduced in Section 2. Section 3 presents the evaluation metric system of cloud service composition including service matching degree and composition synergy degree, and a double-constraint composition optimization model is constructed. An improved ant colony algorithm is designed in Section 4. Section 5 discusses an application case to verify the feasibility and effectiveness of the proposed method. The conclusion section includes the research summary of this paper and future works.

2 Related work

The purpose of service composition optimization is to assess the efficiency and quality of collaboration among multiple services by constructing a comprehensive evaluation system, so as to ensure that manufacturing requirements are fully met. In cloud manufacturing, kinds of massive, geographically dispersed, and heterogeneous manufacturing resources and capabilities are registered as web services with different quality properties. Therefore, it is challenging to perform complex manufacturing tasks in a multi-service collaborative manner, which makes service composition evaluation and selection extremely significant and has been extensively studied by multinational researchers.

2.1 Cloud service composition

Service composition in cloud manufacturing is an NP-hard problem with multiple decision-making objectives, which is solved simplistically by using some heuristic algorithms, such as genetic algorithm [14–16], artificial bee colony algorithm [17], particle swarm optimization algorithm [18], ant colony optimization algorithm [19], and chaos algorithm [20]. Many academics have also proposed hybrid algorithms to improve the problem solution by exploiting the advantages of the abovementioned algorithms. Wang et al. [21] proposed a novel simplex-based bee-colony hybrid algorithm, using a simplex method along with a chaotic global optimal guidance strategy, to solve the composition constraint model of manufacturing services. To achieve the quality-aware cloud manufacturing service composition, Jin et al. [22] proposed a new hybrid teaching-based optimization

method, which combined the advantages of uniform mutation, adaptive flower pollination, and a teaching-based optimization algorithm. Gavvala et al. [23] proposed a whale optimization algorithm with eagle strategy (ESWOA) to ensure the proper balance among cloud services with differing quality features in collaborative operations. Zhu et al. [24] constructed a multi-batch subtasks parallel composition scheme of manufacturing cloud services, and designed an improved hybrid differential evolution and teaching-based optimization method to enable the combined manufacturing service stable, reliable, and efficient. Bouzary et al. [25] proposed a hybrid approach based on a gray wolf optimizer and genetic algorithm to select and composite available manufacturing resources into an integrated cloud service, which could meet individual manufacturing requirements from users while keeping up the optimal quality of service.

2.2 Cloud service optimization selection

It has been a considerable challenge to select the optimal combination of services from numerous candidate manufacturing services in cloud manufacturing. To address the multi-objective manufacturing service optimal-selection problem, Tao et al. [26] described the correlations among manufacturing services, and proposed an approach based on the principles of particle swarm optimization (PSO). Yi et al. [8] proposed an improved fast non-dominated ranking genetic algorithm with an elite strategy, which used the optimal decision method combining hierarchical analysis and entropy value theory to comprehensively evaluate all manufacturing service compositions. Chattopadhyay et al. [27] studied the multi-constrained QoS-aware cloud service selection problem, which was addressed by the proposed optimal approach based on Pareto front construction. Yin et al. [28] proposed a non-dominated sorting genetic algorithm (NSGA) III-based optimization method for composition selection of cloud manufacturing services, and the conducted experiment showed that NSGA-III achieved the optimal solution more efficiently compared to other methods. Bi et al. [29] proposed a preference-based NSGA-III method to solve the service optimization selection problem in cloud manufacturing. Compared with the traditional preference-based multi-objective algorithms, the proposed method directly distributed the preference reference points on the NSGA-III hyperplane according to the preference weights of QoS properties provided by users and facilitated the search for optimal solutions. In addition, Yuan et al. [10] designed an evaluation method based on gray correlation analysis for the optimization selection of various manufacturing services. Liang et al. [30] proposed a deep reinforcement learning algorithm for the cloud service optimization problem, and experimental results showed that the presented method outperformed the deep reinforcement network and Q-learning algorithm. Liu et al. [31] discussed an adaptive service combination optimization

problem, which was solved by the combined reinforcement learning and deep learning approach.

Nevertheless, the above findings pay little attention to the matching degree between cloud services and manufacturing tasks, as well as the synergy degree among different cloud services. Meanwhile, most studies regard only the interests of manufacturing service requestors, but neglect the interests of manufacturing resource providers, which reduces the operational effectiveness of cloud services. For this reason, this paper establishes a double-constraint composition optimization model that is solved by using the improved ant colony algorithm (IACO).

3 Mathematical model of cloud service composition

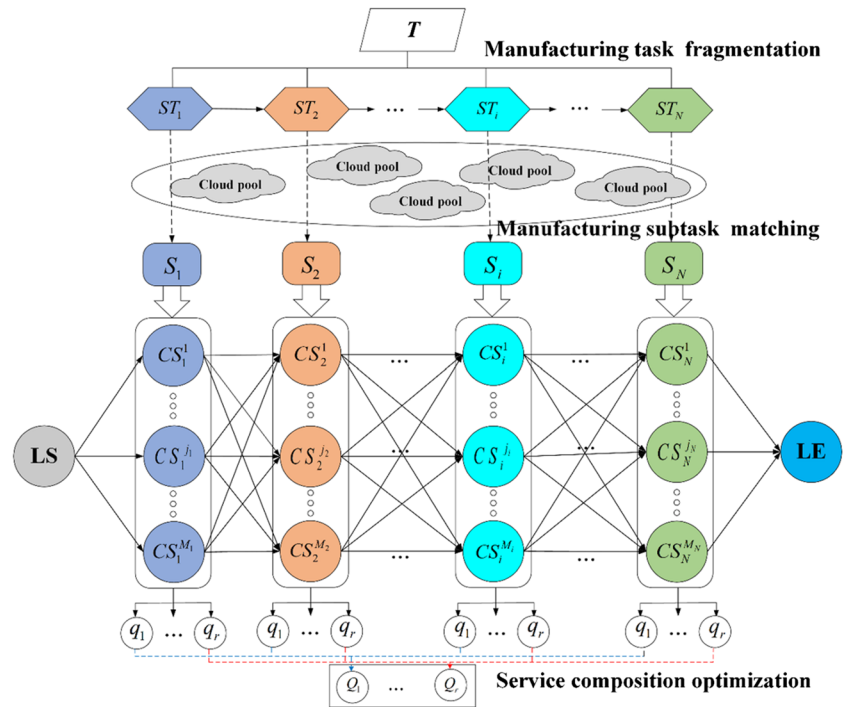
3.1 Problem description

The operation of the submitted complex manufacturing tasks in the cloud platform can be divided into three main phases, including manufacturing task fragmentation, manufacturing subtask matching, and service composition optimization, as shown in Fig. 1. At the manufacturing task fragmentation stage, the cloud service platform first analyzes the manufacturing task T submitted by requestors, and based on the manufacturing process knowledge invokes task resolution tools to divide the task into manufacturing subtasks with process relevance, which can be expressed as $T = \{ST_1, ST_2, \dots, ST_i, \dots, ST_N\}$, where ST_i represents i -th manufacturing subtask. The decomposed subtasks will perform matching operations with various cloud services published by providers in the manufacturing subtask matching phase. For the manufacturing subtask ST_i , a set of candidate services can be matched in the cloud resource pool, denoted as $S_i = \{S_i^1, S_i^2, \dots, S_i^j, \dots, S_i^M\}$, where S_i^j is the j -th candidate service for the manufacturing subtask ST_i and M is the number of candidate services for ST_i . In the service composition optimization stage, the most qualified manufacturing service S_i^j will be selected from each candidate service set S_i to perform the manufacturing subtask ST_i based on the established evaluation criteria, which consist of QoS attributes of each candidate manufacturing service, including service time, service cost and service reliability, noted as $QoS = \{q_1, q_2, \dots, q_r\}$, where r is the number of QoS attributes.

3.2 Evaluation metric system for cloud service composition

Traditional composition evaluation methods of cloud manufacturing services only consider factors that describe the capability and quality of individual services [32, 33], with regard to service time, service cost, reliability of service

Fig. 1 Cloud service composition optimization scheme



operation, and so on, without accounting for matching degree between cloud services and manufacturing tasks, as well as collaboration degree among cloud services. Therefore, this paper establishes a novel cloud service composition evaluation system, which is described as follows:

3.2.1 Service time

In cloud manufacturing, the execution of manufacturing tasks involves the delivery and processing of workpieces. Thereby, service time represents the total time required for the best candidate services which takes the least time to complete the manufacturing subtasks, including service execution time and logistics time. The service time (St) is expressed by the formula (1).

$$St = \sum_{i=1}^N \min [mt(x_{ij}) + lt(x_{(i-1)j}, x_{ij})] \tag{1}$$

where x_{ij} indicates the candidate service CS_i^j from the candidate service set S_i , $mt(x_{ij})$ is the execution time of the j -th candidate service for manufacturing subtask ST_i , and $lt(x_{(i-1)j}, x_{ij})$ means the transport time of workpieces being machined between $x_{(i-1)j}$ and x_{ij} .

3.2.2 Service cost

Service cost is defined as the sum of the minimum manufacturing and logistics costs required to satisfy manufacturing

subtasks for the best manufacturing service selected from the candidate set, which is calculated by the formula (2).

$$Sc = \sum_{i=1}^N \min [mc(x_{ij}) + lc(x_{(i-1)j}, x_{ij})] \tag{2}$$

where $mc(x_{ij})$ denotes the manufacturing cost of the service x_{ij} to complete the subtask ST_i , and $lc(x_{(i-1)j}, x_{ij})$ notes the material delivery cost between the subtask $x_{(i-1)j}$ and x_{ij} when the j -th candidate service is selected.

3.2.3 Service reliability

Service reliability (Sr) means the ability of all best candidate manufacturing services to correctly complete manufacturing task T under specific constraints, which can be obtained by the cloud platform based on historical operation records of the selected services and is expressed as the formula (3).

$$Sr = \prod_{i=1}^N \max r(x_{ij}) \tag{3}$$

where $r(x_{ij})$ is the reliability of the candidate manufacturing service x_{ij} to accomplish the subtask ST_i .

3.2.4 Service matching degree

Service matching is a measurement of the quality level of the results produced by candidate services carrying out the assigned manufacturing subtasks in cloud manufacturing,

which is related to service activity, service history, and service delivery rate. Service activity refers to the number of times that manufacturing services are selected to satisfy the requirements of manufacturing subtasks over a period. There is a positive correlation between service activity and service matching degree, the higher the activity level, the better the execution ability of the services, and the higher the matching degree between the services and the subtasks. For the i -th subtask, the activity of its j -th candidate service (Sa_{ij}) is expressed as the formula (4).

$$Sa_{ij} = o(x_{ij}) / O(x) \tag{4}$$

where $o(x_{ij})$ represents times the service x_{ij} has been invoked, and $O(x)$ denotes the total number of times all services have been invoked in the same period.

Service history means the track record of the selected service in performing various manufacturing tasks, including service experience, knowledge, and service evaluation from users. Therefore, service history (Sh_{ij}) is calculated by the formula (5).

$$Sh_{ij} = \omega_1 To_{ij} + \omega_2 K_{ij} + \omega_3 E_{ij} \tag{5}$$

where To_{ij} is the effective uptime percentage of the service x_{ij} in cloud, K_{ij} refers to the proportion of process knowledge held by the service x_{ij} to the knowledge of similar services in the whole platform, E_{ij} is a triangular fuzzy number, which denotes the service satisfaction rating graded by users, ω_1 , ω_2 , and ω_3 are their weights, and $\omega_1 + \omega_2 + \omega_3 = 1$. Obviously, the richer the experience and knowledge of the selected service and the higher the service evaluation, the higher the matching degree of the service.

Service delivery (Sd_{ij}) is the ratio of the number of manufacturing tasks successfully performed by the service to the

times the service was invoked, which is expressed by the formula (6).

$$Sd_{ij} = o_d(x_{ij}) / o(x_{ij}) \tag{6}$$

Therefore, the matching degree of j -th candidate service (Sm_{ij}) can be represented as the formula (7). Here λ_1 , λ_2 , and λ_3 are weights, and $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

$$Sm_{ij} = \lambda_1 Sa_{ij} + \lambda_2 Sh_{ij} + \lambda_3 Sd_{ij} \tag{7}$$

3.2.5 Service collaboration degree

Collaboration among services is also a key factor that affects the service quality in the cloud, which represents the ability of multiple cloud manufacturing services to work together to efficiently accomplish a complex manufacturing task. Consequently, Service collaboration is mainly characterized by metrics such as service process execution time and the efficiency of information interaction among services. The faster information interaction and execution among services, the higher the inter-service collaboration level[42]. And then, service collaboration degree can be calculated by the following formula (8).

$$CS_{i(i+1)} = \frac{T_{ij} + T_{(i+1)j}}{T_{i(i+1)j}} \tag{8}$$

where $CS_{i(i+1)}$ is the collaboration degree of best candidate service x_{ij} and $x_{(i+1)j}$ for subtask ST_i and ST_{i+1} , respectively, T_{ij} (or $T_{(i+1)j}$) denotes the time taken by service x_{ij} (or $x_{(i+1)j}$) to complete subtask ST_i (or ST_{i+1}) independently, and $T_{i(i+1)j}$ is the total time spent by two cloud manufacturing services x_{ij} , $x_{(i+1)j}$ cooperating to complete two manufacturing tasks. The calculation of $T_{i(i+1)j}$ is closely related to the interaction between services, which is calculated using the following formula.

$$T_{i(i+1)j} = \begin{cases} T_{ij} + T_{(i+1)j}, & \text{Subtasks } ST_i \text{ and } ST_{i+1} \text{ are executed serially} \\ \max[T_{ij}, T_{(i+1)j}], & \text{Subtasks } ST_i \text{ and } ST_{i+1} \text{ are executed parallelly} \\ T_{ij} + T_{(i+1)j} + 2\xi_{i(i+1)} \sqrt{T_{ij} \cdot T_{(i+1)j}}, & \text{Subtasks } ST_i \text{ and } ST_{i+1} \text{ are executed in close coupling} \end{cases} \tag{9}$$

where ξ_{ij} is the coupling coefficient between manufacturing services x_{ij} and $x_{(i+1)j}$; its value mainly depends on information interaction efficiency among services. The more frequent the information interaction between two services, the smaller the value of ξ_{ij} ($0 < \xi_{ij} < 1$). For example, when two manufacturing services collaborate frequently and information interaction with each other is smooth, the value ξ_{ij} can be 0.1; conversely, when the two services have no interaction, ξ_{ij} is 0.9.

3.3 Double-constrained multi-objective optimization model

The goal of cloud manufacturing service composition is to select a service collaboration program with the best overall QoS among many composition solutions to complete complex manufacturing tasks submitted by users under various QoS constraint objectives. Therefore the constructed multi-objective optimization model for a cloud manufacturing service composition problem is as follows.

$$f(T) = \delta_1 Sm + \delta_2 CS + \delta_3 Sr + \delta_4 St + \delta_5 Sc \tag{10}$$

$$Sm = \sum_{i=1}^N \max(Sm_{ij} \cdot x_{ij}) \tag{11}$$

$$CS = \sum_{i=1}^{N-1} CS_{i(i+1)} \tag{12}$$

$$\begin{aligned} s.t. \quad St &\leq T_{\max} \\ Sc &\leq C_{\max} \\ Sr &\geq R_{\min} \end{aligned} \tag{13}$$

$$x_{ij} = \begin{cases} 1, & \text{The } i\text{-th manufacturing subtask is assigned to the } j\text{-th cloud service} \\ 0, & \text{others} \end{cases} \tag{14}$$

where T_{\max} , C_{\max} , and R_{\min} are constraints given by the requirement users from the cloud platform in terms of service time, cost, and reliability, $\delta_1, \delta_2, \delta_3, \delta_4$, and δ_5 are the weights of individual metrics, and $\sum_{i=1}^5 \delta_i = 1$.

However, service time and cost are negative indicators that users expect to be as low as possible, service reliability, service matching degree, and service collaboration degree are beneficial metrics for which users expect higher values. Moreover, these five indicators have different scales. Therefore, the above indicators need to be normalized by the formula (15). Formula (15a) is used to calculate the positive indicators, and other cost-based indicators are standardized by formula (15b).

$$\bar{q} = \begin{cases} 1 & q_{\max} - q_{\min} = 0 \\ (q - q_{\min}) / (q_{\max} - q_{\min}) & q_{\max} - q_{\min} \neq 0(a) \\ (q_{\max} - q) / (q_{\max} - q_{\min}) & q_{\max} - q_{\min} \neq 0(b) \end{cases} \tag{15}$$

4 An improved ant colony optimization algorithm

4.1 ACO algorithm

The ant colony optimization algorithm(ACO) is a new metaheuristic calculation method inspired by the foraging behavior of ant colonies in nature, which has been successfully applied to many fields [34]. Corresponding to the ACO algorithm, some parameters in this paper are as follows:

1. M_i denotes the number of candidate services for the i -th manufacturing subtask.

2. m stands for the number of ants.
3. $\tau_{lk}(t)$ means the residual pheromone on the path from node l to node k at the time t . The initial time is set to $\tau_{lk}(0) = C$ and C is the initial pheromone concentration, usually a small constant.
4. $\eta_{lk}(t)$ represents heuristic information, and $\eta_{lk}(t) = 1/d_{lk}$. d_{lk} indicates the distance between node l to node k .

Thus, the probability that at moment t ant u chooses to move from node l to node k can then be calculated by the following Eq. (16).

$$P_{lk}^u(t) = \begin{cases} \frac{[\tau_{lk}(t)]^\alpha [\eta_{lk}(t)]^\beta}{\sum_{k \in allowedNs(t)} [\tau_{lk}(t)]^\alpha [\eta_{lk}(t)]^\beta}, & k \in allowedNs(l) \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

where α and β denote the relative importance parameters of pheromone τ and heuristic information η , respectively. $allowedNs(l)$ records the set of all nodes allowed to be selected starting from node l , which is also the set of the next optional cloud manufacturing services. When all ants reach the end node, the number of iterations is increased by 1($t \leftarrow t + 1$), and the pheromone is updated as shown in Eq. (17).

$$\tau_{lk}(t + 1) = (1 - \rho) \cdot \tau_{lk}(t) + \Delta\tau_{lk}(t, t + 1) \tag{17}$$

where ρ is the pheromone volatilization coefficient, $\Delta\tau_{lk}(t, t + 1)$ represents the pheromone concentration increment generated by the ant from node l to node k , calculated by the formula (18).

$$\Delta\tau_{lk}(t, t + 1) = \sum_{u=1}^m \Delta\tau_{lk}^u(t, t + 1) \tag{18}$$

where $\Delta\tau_{lk}^u(t, t + 1)$ denotes the pheromone concentration produced by ant u from node l to node k , which is calculated by the Eq. (19).

$$\Delta\tau_{lk}^u(t, t + 1) = \begin{cases} Q/L_u, & \text{ant } u \text{ is transferred from node } l \text{ to node } k \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

where Q is a constant, L_u is the total distance traveled by ant u after traversing all nodes in the current time cycle.

4.2 Improvement strategies

The traditional ACO algorithm described above has been successful in solving the traveling salesman problem (TSP). However, the algorithm has the disadvantages of slow convergence and vulnerability to fall into local optimization, which makes it unsuitable for solving the service composition problem. Therefore, this paper proposes corresponding improvement strategies.

Firstly, the path chosen by the ant after each traversal is divided into N steps, namely $\langle LS \rightarrow S_1 \rangle, \langle S_1 \rightarrow S_2 \rangle, \dots, \langle S_{N-1} \rightarrow S_N \rangle$, whereby the traversed nodes can be represented as follows.

$$allowedNs(S_i) = S_j^i (S_j^i \in S_i, j = 1, 2, 3, \dots, M) \tag{20}$$

Then, cloud manufacturing service composition is a multi-objective optimization problem, which is subject to dual constraints from service requestors and resource providers in the cloud platform. Therefore, it is necessary to eliminate invalid paths that do not satisfy these constraints. Specifically, only ants taking valid paths are selected to update the pheromone concentration, and only optimal paths can trigger pheromone updating. As a result, therefore, both the objective function and the above formula (19) need to be converted into the following equations.

$$f'(T) = 1/\delta_1 Sm + \delta_2 CS + \delta_3 Sr + \delta_4 St + \delta_5 Sc \tag{21}$$

$$\Delta\tau_{lk}^{best}(t, t + 1) = \begin{cases} Q/L_{best}, & \text{the best ant } u \text{ travel } <l \rightarrow k > \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

Although the above improvement strategies enhance the convergence speed of the algorithm, which may make it fall into the local optimum. For this reason, the roulette mechanism is adopted, so that the algorithm can go to the next candidate services with a smaller probability, and traverse the new path to solve the local optimum problem. The flow of the proposed improved ACO algorithm is shown in Fig. 2. The specific solution steps are described below:

Step 1: The set of paths that ants are allowed to visit is set. The number of ants m and the pheromone concentration C is initialized.

Step 2: All indicators are normalized according to Eq. (15), which is used to build the objective function (21).

Step 3: Randomly place m ants on each of the allowed path set nodes.

Step 4: Search all the paths connected to this ant node and calculate the state transition probability that the ant can visit the next node using the formula (16).

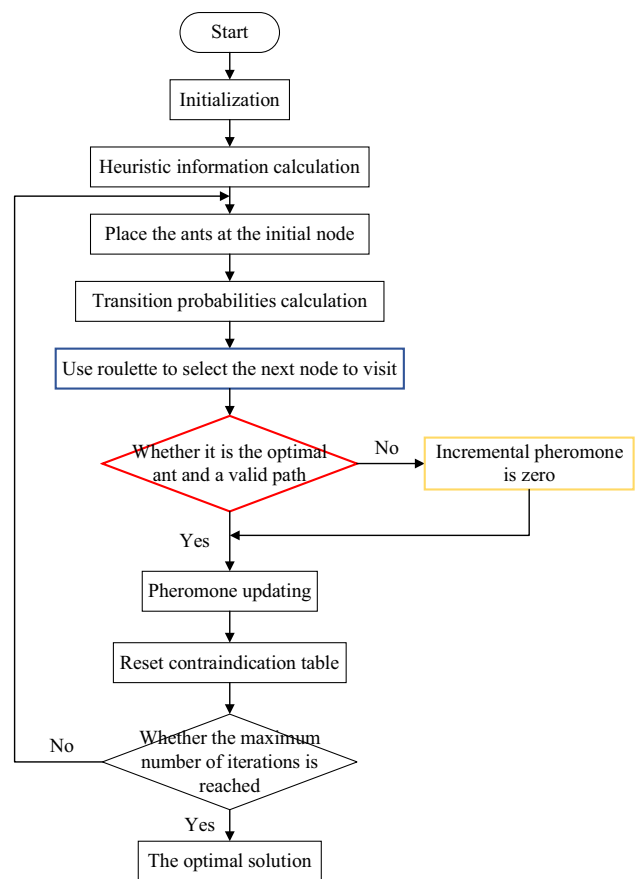


Fig. 2 Flow of IACO algorithm

Step 5: A roulette mechanism is used to select the next node where the ant will visit based on the probability calculated in Step 4.

Step 6: Determine whether it is the optimal ant and whether the path is valid. If the answer is true, the access path pheromone is updated; otherwise, the pheromone increment on the invalid path is set to 0.

Step 7: Pheromones are updated and the contraindication table is cleared based on Eqs. (17)–(19).

Step 8: Repeat Steps 3 to 7 until the maximum number of iterations, and then the optimal solution is outputted.

5 Application case

5.1 Case description

To verify the feasibility and effectiveness of the proposed method in solving the cloud manufacturing service composition optimization problem, this paper applies the method to a cloud service application case of automotive component manufacturing. In this case, there are 100 car bumper

manufacturing tasks are submitted by the service requestor to the cloud platform that has been developed and commercially operated by the Chongqing University intelligent manufacturing team for more than 10 years. The requestor specified the execution constraints for these manufacturing tasks, including execution time T_{max} of no more than 30 days, total cost C_{max} of no more than RMB 80,000, and service reliability R_{min} of no less than 0.93. Based on the manufacturing process, this complex manufacturing task is divided into five sub-tasks by invoking the task decomposition tool in the platform, including mold design and forging subtask ST_1 , part processing subtask ST_2 , assembly subtask ST_3 , performance testing subtask ST_4 , painting and packaging subtask ST_5 . Each subtask has been matched with a corresponding set of manufacturing cloud services using an intelligent search engine in the cloud platform, namely $S = (S_1, S_2, S_3, S_4, S_5)$, and QoS attribute values for each candidate service are shown in Table 1, which are dynamically updated in the cloud platform according to the service market. Logistics time and cost between candidate services are shown in Table 2.

Table 1 QoS attribute of candidate services

Candidate services	Service time (St/day)	Service costs (Sc/¥)	Service reliability (Sr)	
S_1	S_1^1	2	246	0.96
	S_1^2	2	229	0.92
	S_1^3	3	195	0.98
	S_1^4	4	183	0.91
	S_1^5	3	157	0.98
S_2	S_2^1	4	112	0.92
	S_2^2	5	100	0.99
	S_2^3	6	85	0.96
	S_2^4	3	95	0.93
	S_2^5	3	86	0.97
S_3	S_3^1	3	60	0.98
	S_3^2	5	45	0.96
	S_3^3	5	68	0.97
	S_3^4	5	31	0.95
	S_3^5	3	63	0.93
S_4	S_4^1	2	55	0.99
	S_4^2	5	52	0.95
	S_4^3	4	56	0.92
	S_4^4	4	59	0.91
	S_4^5	3	34	0.95
S_5	S_5^1	6	238	0.98
	S_5^2	4	151	0.92
	S_5^3	3	179	0.95
	S_5^4	6	168	0.97
	S_5^5	4	243	0.96

Table 2 Logistics time and cost between candidate services

Logistics connection	Logistics time (day)/Logistics costs(¥)				
	1	2	3	4	5
$S_1^1-S_2^j$	5/45	4/46	3/26	2/18	2/25
$S_1^2-S_2^j$	3/25	6/65	4/40	4/36	2/20
$S_1^3-S_2^j$	3/25	2/30	5/34	5/46	4/50
$S_1^4-S_2^j$	4/35	6/60	4/50	5/55	3/36
$S_1^5-S_2^j$	2/25	4/36	1/12	2/27	4/34
$S_2^1-S_3^j$	1/17	1/18	4/32	2/23	5/42
$S_2^2-S_3^j$	4/41	1/16	5/44	2/25	3/32
$S_2^3-S_3^j$	4/43	5/58	1/18	1/20	2/28
$S_2^4-S_3^j$	3/31	2/21	4/35	2/24	3/26
$S_2^5-S_3^j$	3/21	4/36	2/16	3/18	4/32
$S_3^1-S_4^j$	1/8	2/22	1/9	2/15	2/13
$S_3^2-S_4^j$	3/15	4/38	3/26	2/14	2/10
$S_3^3-S_4^j$	4/46	3/35	5/65	4/50	6/68
$S_3^4-S_4^j$	3/30	2/25	3/25	4/42	4/45
$S_3^5-S_4^j$	5/66	4/45	2/20	2/36	3/28
$S_4^1-S_5^j$	5/48	4/42	5/55	3/32	4/42
$S_4^2-S_5^j$	5/48	6/72	3/35	5/60	4/45
$S_4^3-S_5^j$	6/60	5/55	2/25	1/12	2/18
$S_4^4-S_5^j$	1/15	2/27	3/38	3/32	4/45
$S_4^5-S_5^j$	1/16	3/38	2/24	3/32	5/48

Combining with the service history records shown in the cloud platform, the values of $\lambda_1, \lambda_2,$ and λ_3 are set to 0.4, 0.3, and 0.3, respectively, so the matrix of service matching degree Sm can be obtained by using the Eq. (7).

$$Sm = \begin{bmatrix} 0.72 & 0.29 & 0.43 & 0.53 & 0.49 \\ 0.77 & 0.37 & 0.44 & 0.53 & 0.54 \\ 0.62 & 0.48 & 0.53 & 0.47 & 0.67 \\ 0.31 & 0.65 & 0.69 & 0.44 & 0.64 \\ 0.89 & 0.61 & 0.72 & 0.65 & 0.74 \end{bmatrix}$$

According to the above definition of service collaboration degree and its formulae (8) and (9), the service collaboration degree CS can be calculated as follows:

$CS_{12} =$	$\begin{bmatrix} 0.736 & 0.684 & 0.546 & 0.632 & 0.667 \\ 0.588 & 0.784 & 0.625 & 0.772 & 0.736 \\ 0.909 & 1.000 & 0.589 & 0.770 & 0.770 \\ 0.625 & 0.835 & 0.567 & 0.760 & 0.624 \\ 0.523 & 0.836 & 0.628 & 0.668 & 0.914 \end{bmatrix}$	$, CS_{13} =$	$\begin{bmatrix} 1.000 & 0.579 & 0.668 & 0.775 & 0.652 \\ 0.559 & 0.909 & 0.715 & 0.628 & 0.715 \\ 0.652 & 0.553 & 0.855 & 0.756 & 0.954 \\ 0.834 & 0.910 & 0.669 & 0.770 & 0.717 \\ 0.910 & 0.810 & 0.773 & 0.825 & 0.866 \end{bmatrix}$
$CS_{14} =$	$\begin{bmatrix} 0.834 & 0.628 & 0.566 & 0.755 & 0.850 \\ 0.866 & 0.910 & 0.909 & 0.563 & 0.772 \\ 1.000 & 0.910 & 0.905 & 0.788 & 0.699 \\ 0.556 & 0.770 & 0.642 & 0.885 & 0.916 \\ 1.000 & 0.766 & 0.769 & 0.854 & 0.559 \end{bmatrix}$	$, CS_{15} =$	$\begin{bmatrix} 0.595 & 0.668 & 0.866 & 0.705 & 0.910 \\ 0.899 & 0.866 & 0.909 & 1.000 & 0.699 \\ 0.755 & 0.769 & 0.599 & 0.527 & 0.625 \\ 0.771 & 0.671 & 0.593 & 0.925 & 0.936 \\ 1.000 & 0.855 & 0.647 & 0.980 & 0.735 \end{bmatrix}$
$CS_{23} =$	$\begin{bmatrix} 0.835 & 0.863 & 0.985 & 0.772 & 0.593 \\ 0.542 & 0.754 & 0.910 & 0.855 & 0.776 \\ 0.669 & 0.996 & 1.000 & 0.885 & 0.668 \\ 0.886 & 0.735 & 0.863 & 0.566 & 0.775 \\ 0.566 & 0.897 & 0.910 & 0.915 & 0.866 \end{bmatrix}$	$, CS_{24} =$	$\begin{bmatrix} 0.860 & 0.960 & 0.875 & 0.834 & 0.836 \\ 0.755 & 0.760 & 0.541 & 0.536 & 0.952 \\ 0.845 & 0.865 & 0.596 & 0.945 & 0.987 \\ 0.696 & 0.910 & 0.955 & 0.996 & 0.771 \\ 0.962 & 0.625 & 0.788 & 0.667 & 0.911 \end{bmatrix}$
$CS_{25} =$	$\begin{bmatrix} 0.569 & 0.910 & 0.854 & 0.770 & 0.996 \\ 0.820 & 0.520 & 0.778 & 0.652 & 0.837 \\ 0.885 & 0.746 & 0.910 & 0.996 & 0.945 \\ 0.536 & 0.642 & 0.668 & 0.886 & 0.955 \\ 0.996 & 0.756 & 0.734 & 0.634 & 0.566 \end{bmatrix}$	$, CS_{34} =$	$\begin{bmatrix} 0.778 & 0.623 & 0.668 & 0.996 & 0.886 \\ 0.714 & 0.557 & 0.770 & 0.996 & 0.995 \\ 0.910 & 0.565 & 0.835 & 0.832 & 0.885 \\ 0.945 & 0.856 & 0.775 & 0.536 & 0.533 \\ 0.966 & 0.910 & 0.752 & 0.760 & 0.886 \end{bmatrix}$
$CS_{35} =$	$\begin{bmatrix} 0.863 & 0.775 & 0.966 & 0.775 & 0.911 \\ 0.927 & 0.863 & 0.523 & 0.715 & 0.775 \\ 0.759 & 0.912 & 0.690 & 0.910 & 0.836 \\ 0.886 & 0.662 & 0.556 & 0.678 & 0.833 \\ 0.889 & 0.799 & 0.835 & 0.978 & 0.955 \end{bmatrix}$	$, CS_{45} =$	$\begin{bmatrix} 0.887 & 0.775 & 0.663 & 0.995 & 0.669 \\ 0.863 & 0.669 & 0.559 & 0.667 & 0.775 \\ 0.998 & 0.687 & 0.597 & 0.710 & 0.812 \\ 0.915 & 0.966 & 0.956 & 0.855 & 0.872 \\ 0.823 & 0.833 & 0.956 & 0.996 & 0.886 \end{bmatrix}$

5.2 Experimental analysis

The experiment is carried out using MATLAB R2018a, and the IACO algorithm parameters are set as follows: the number of ants $m = 10$, the initial pheromone concentration $C = 1$, the pheromone increment coefficient $Q = 1$,

the pheromone factor $\alpha = 1$, the heuristic factor $\beta = 5$, the pheromone volatilization coefficient $\rho = 0.25$, the weights of five metrics $\delta_1 = \delta_2 = \delta_3 = \delta_4 = \delta_5 = 0.2$, and the maximum number of iterations is set to 100. The experimental result is shown in Fig. 3.

Fig. 3 Experimental result

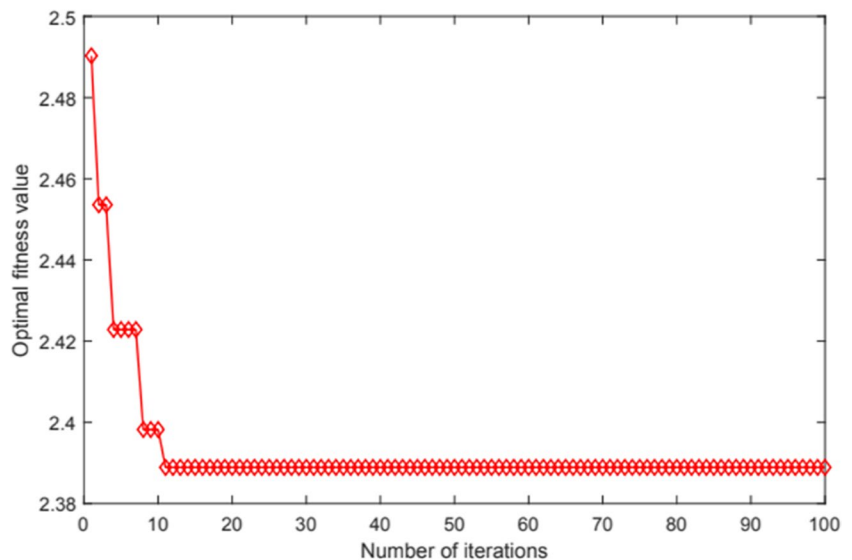
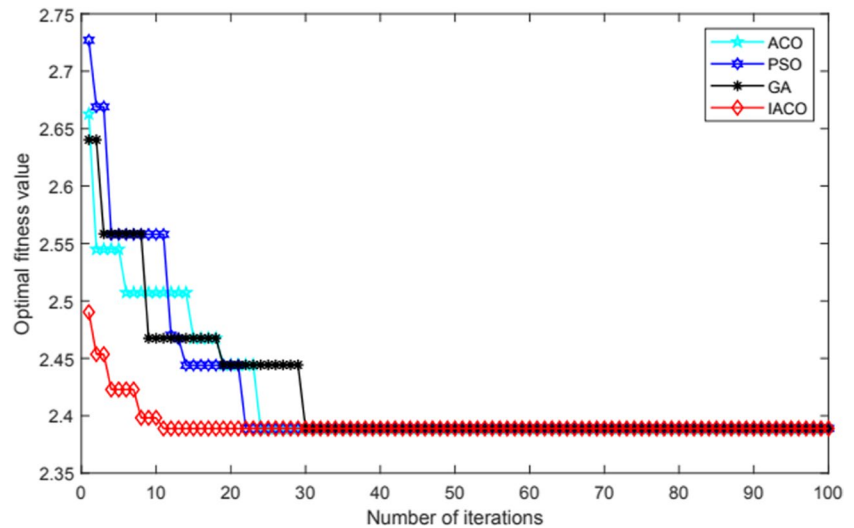


Fig. 4 Comparison of four algorithms



As illustrated in Fig. 3, the experimental result is convergent after 11 iterations of the algorithm, and the optimal fitness value obtained is 2.3889. Correspondingly, the values of five objective QoSs are $St = 28$, total $Sc = 79100$, $Sr = 0.968$, $Sm = 3.41$, and $CS = 8.157$, and the optimal service composition solution is $(S_1^1, S_2^5, S_3^1, S_4^5, S_5^1)$.

In order to validate the performance of the IACO algorithm, it was compared with the Particle Swarm Algorithm (PSO), Genetic Algorithm (GA), and the original Ant Colony Optimisation Algorithm (ACO) to solve the above service composition optimization problem via MATLAB R2018a. As shown in Fig. 4, it can be found that the proposed algorithm in this paper has a faster convergence rate and outperforms the other three algorithms in terms of accuracy and optimization effectiveness. Therefore, the proposed method in this paper has better performance in solving cloud service composition optimization problems.

6 Conclusions

In order to realize more efficient and accurate matching and execution of various personalized manufacturing requirements and cloud services on the manufacturing service platform, the concepts of service matching degree between cloud services and manufacturing tasks, as well as the collaboration degree among cloud services were proposed in this paper. A novel evaluation system of cloud manufacturing service composition, including service time, service cost, service reliability, service matching degree, and service collaboration degree, was constructed. Then, taking into account the interests of both the service supplier and requestors in the platform, a two-constraint service composition optimization model was constructed, and solved by using the IACO algorithm. Finally, through a car bumper

cloud manufacturing case from the cloud platform that has been developed and commercially operated by the Chongqing University intelligent manufacturing team, the feasibility and effectiveness of the proposed method were verified.

In future research, with the penetrating application of the new generation artificial intelligence technologies in the industrial field, we will devote ourselves to studying the application of advanced algorithms, such as deep learning and reinforcement learning, to support the efficient operation of cloud manufacturing services.

Author contribution Chao Yin and Xiaobin Li contributed to the core ideas and concepts of the paper. The method design, material preparation, case data collection, and analysis were performed by Xiaobin Li and Shanglin Li. The first and final revised version of the manuscript was accomplished by Xiaobin Li and Shanglin Li. All authors read and approved the manuscript.

Funding The authors gratefully acknowledge financial support from the National Key R&D Program of China (No. 2022YFB3305603), the National Natural Science Foundation of China (No. 52075060, No. 51875065), and the Fundamental Research Funds for the Central Universities (No. 2023CDJXY-021).

Declarations

Competing interests The authors declare no competing interests.

References

1. Pei J, Wang Z, Li X, Wang XV, Yang B, Zheng J (2023) Energy consumption prediction and optimization of industrial robots based on LSTM. *J Manuf Syst* 70:137–148
2. Alzoubi YI, Al-Ahmad A, Kahtan H, Jaradat A (2022) Internet of things and blockchain integration: security, privacy, technical, and design challenges. *Future Internet* 14(7):216

3. Li BH, Zhang L, Wang SL, Tao F, Cao JW, Jiang XD, Song X, Chai XD (2010) Cloud manufacturing: a new service-oriented networked manufacturing model. *Comput Integr Manuf Syst* 16(1):1–7,16
4. Wang Y, Wang S, Yang B, Gao B, Wang SB (2022) An effective adaptive adjustment method for service composition exception handling in cloud manufacturing. *J Intell Manuf* 33(3):735–751
5. Li XB, Fang ZW, Yin C (2020) A machine tool matching method in cloud manufacturing using Markov decision process and cross-entropy. *Robot Comput Integr Manuf* 65:101968
6. Tao F, Zhang L, Guo H, Luo YL, Ren L (2011) Research on cloud manufacturing characteristics and key issues of cloud service composition. *Comput Integr Manuf Syst* 17(03):477–486SS
7. Chen YL, Wang L, Liu J, Zuo LD, Niu YF (2019) Optimization of cloud manufacturing resource service composition selection based on i-NSGA-II-JG algorithm. *Comput Integr Manuf Syst* 25(11):2892–2904
8. Yi AB, Yao XF, Zhou HF, Zhang CJ (2017) Multi-objective optimization selection of equipment resources in cloud manufacturing environment. *Comput Integr Manuf Syst* 23(06):1187–1195
9. Gong XR, Yin C, Li X (2019) A grey correlation-based supply-demand matching of machine tools with multiple quality factors in cloud manufacturing environment. *J Ambient Intell Hum Comput* 10(3):1025–1038
10. Yuan MH, Zhou Z, Cai XX, Sun C, Gu WB (2020) Service composition model and method in cloud manufacturing. *Robot Comput Integr Manuf* 61(101840):1–13
11. Zhou J, Yao X (2017) Multi-population parallel self-adaptive differential artificial bee colony algorithm with application in large-scale service composition for cloud manufacturing. *Appl Soft Comput* 56:379–397
12. Li XB, Zhuang P, Yin C (2019) A metadata-based manufacturing resource ontology modeling in cloud manufacturing systems. *J Ambient Intell Hum Comput* 10(3):1039–1047
13. Yang YF, Yang B, Wang SL, Liu W, Jin TG (2019) An improved grey wolf optimizer algorithm for energy-aware service composition in cloud manufacturing. *Int J Adv Manuf Technol* 105(7–8):3079–3091
14. Ghomi EJ, Rahmani AM, Qader NN (2019) Cloud manufacturing: challenges, recent advances, open research issues, and future trends. *Int J Adv Manuf Technol* 102(9–12):3613–3639
15. Jiang YR, Tang L, Liu HL, Zeng A (2021) A variable-length encoding genetic algorithm for incremental service composition in uncertain environments for cloud manufacturing. *Appl Soft Comput* 123:108902
16. Li TY, He T, Wang ZJ, Zhang YF (2020) SDF-GA: a service domain feature-oriented approach for manufacturing cloud service composition. *J Intell Manuf* 31(3):681–702
17. Zhou K, Wen YZ, Wu WY, Ni ZY, Jin TG, Long XJ (2020) Cloud service optimization method based on dynamic artificial ant-bee colony algorithm in agricultural equipment manufacturing. *Math Probl Eng* 2020:9134695
18. Yu CX, Zhang LP, Zhao WF, Zhang SC (2020) A blockchain-based service composition architecture in cloud manufacturing. *Int J Comput Integr Manuf* 33(7):701–715
19. Neshati E, Kazem AAP (2018) QoS-based cloud manufacturing service composition using ant colony optimization algorithm. *Int J Adv Comput Sci Appl* 9(1):437–440
20. Hu YJ, Zhu FF, Zhang L, Lui YK, Wang ZL (2019) Scheduling of manufacturers based on chaos optimization algorithm in cloud manufacturing. *Robot Comput Integr Manuf* 58:13–20
21. Wang ZN, Wang SL, Yang B, Wang YK, Chen RH (2021) A novel hybrid algorithm for large-scale composition optimization problems in cloud manufacturing. *Int J Comput Integr Manuf* 34(9):898–919
22. Jin H, Jiang C, Lv SP, He HP, Liao XT (2022) A hybrid teaching-learning-based optimization algorithm for QoS-aware manufacturing cloud service composition. *Computing*. <https://doi.org/10.1007/s00607-022-01083-4>
23. Gavvala SK, Jatoth C, Gangadharan GR, Buyya R (2019) QoS-aware cloud service composition using eagle strategy. *Futur Gener Compt Syst* 90:273–290
24. Zhu LN, Li PH, Zhou XL (2019) IHDETBO: a novel optimization method of multi-batch subtasks parallel-hybrid execution cloud service composition for cloud manufacturing. *Complexity*:7438710. <https://doi.org/10.1155/2019/7438710>
25. Bouzary H, Chen FF (2019) A hybrid grey wolf optimizer algorithm with evolutionary operators for optimal QoS-aware service composition and optimal selection in cloud manufacturing. *Int J Adv Manuf Technol* 101(9–12):2771–2784
26. Tao F, Zhao D, Hu Y, Zhou Z (2010) Correlation-aware resource service composition and optimal-selection in manufacturing grid. *Eur J Oper Res* 201(1):129–143
27. Chattopadhyay S, Banerjee A (2020) QoS-aware automatic web service composition with multiple objectives. *ACM Trans Web* 14:12
28. Yin C, Xu JS, Li XB (2022) Optimization method of cloud manufacturing service composition based on NSGA-III algorithm. *Comput Integr Manuf Syst* 28(04):1164–1176
29. Bi XX, Yu D, Liu JS, Hu Y (2020) A preference-based multi-objective algorithm for optimal service composition selection in cloud manufacturing. *Int J Comput Integr Manuf* 33(8):751–768
30. Liang HG, Wen XQ, Liu YK, Zhang HF, Zhang L, Wang LH (2021) Logistics-involved QoS-aware service composition in cloud manufacturing with deep reinforcement learning. *Rob Comput Integr Manuf* 67:101991
31. Liu JW, Hu LQ, Cai ZQ, Xing LN, Tan X (2019) Large-scale and adaptive service composition based on deep reinforcement learning. *J Vis Commun Image Represent* 65:102687
32. Jin H, Yao XF, Chen Y (2017) Correlation-aware QoS modeling and manufacturing cloud service composition. *J Intell Manuf* 28(8):1947–1960
33. Li XB, Lan YK, Jiang P, Cao HJ, Zhou J (2022) An efficient computation for energy optimization of robot trajectory. *IEEE T Ind Electron* 69:11436–11446
34. Dorigo M, Gambardella LM (1997) Ant colonies for the traveling salesman problem. *Bio Syst* 43(2):73–81

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.