



Article ID 1007-1202(2024)04-0323-15 DOI <https://doi.org/10.1051/wujns/2024294323>

Cite this article: ZHOU Liliang, LI Ben, YU Qing, *et al.* Multi-Strategy Improvement of Sparrow Search Algorithm for Cloud Manufacturing Service Composition[J]. *Wuhan Univ J of Nat Sci*, 2024, 29(4): 323-337.

Multi-Strategy Improvement of Sparrow Search Algorithm for Cloud Manufacturing Service Composition

□ ZHOU Liliang¹, LI Ben², YU Qing², DAI Guilan³, ZHOU Guofu^{4†}

1. Tenth Research Institute, China Electronics Technology Group Corporation, Chengdu 610000, Sichuan, China;
2. Department of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, Jiangsu, China;
3. Research Institution of Information Technology, Tsinghua University, Beijing 100084, China;
4. School of Computer Science, Wuhan University, Wuhan 430072, Hubei, China

Abstract: In existing research, the optimization of algorithms applied to cloud manufacturing service composition based on the quality of service often suffers from decreased convergence rates and solution quality due to single-population searches in fixed spaces and insufficient information exchange. In this paper, we introduce an improved Sparrow Search Algorithm (ISSA) to address these issues. The fixed solution space is divided into multiple subspaces, allowing for parallel searches that expedite the discovery of target solutions. To enhance search efficiency within these subspaces and significantly improve population diversity, we employ multiple group evolution mechanisms and chaotic perturbation strategies. Furthermore, we incorporate adaptive weights and a global capture strategy based on the golden sine to guide individual discoverers more effectively. Finally, differential Cauchy mutation perturbation is utilized during sparrow position updates to strengthen the algorithm's global optimization capabilities. Simulation experiments on benchmark problems and service composition optimization problems show that the ISSA delivers superior optimization accuracy and convergence stability compared to other methods. These results demonstrate that our approach effectively balances global and local search abilities, leading to enhanced performance in cloud manufacturing service composition.

Key words: cloud manufacturing; service composition optimization; quality of service; sparrow search algorithm

CLC number: TP183

0 Introduction

Cloud manufacturing is a new paradigm of service-oriented manufacturing, where manufacturing resources and capabilities are virtualized and encapsulated into manufacturing services^[1]. Service composition and optimization selection (SCOS)^[2] is the key technology to realize resource and capacity sharing in the cloud manufac-

turing system. The main process is to search for services from the candidate service pool for each subtask in the cloud manufacturing system, then generate composite services and optimize them^[3]. Therefore, the optimization goal of the SCOS problem is to find the optimal composite services within the feasible time, so that it not only meets the functional requirements of users, but also needs to have the optimal global service quality (QoS)^[4].

Received date: 2024-02-29 © Wuhan University 2024

Foundation item: Supported by the National Natural Science Foundation of China (62272214)

Biography: ZHOU Liliang, male, Senior engineer, research direction: avionics information systems and sensor management. E-mail: zhoul@163.com

† Corresponding author. E-mail: gfzhou@whu.edu.cn

In recent years, many scholars have explored cloud manufacturing service composition, achieving significant progress. Zhou *et al.*^[5] proposed a mixed artificial bee colony method, combining Archimedean distribution with an artificial bee colony to guide the population. This method enhances local search performance but risks falling into local optima during late convergence due to fixed neighborhood modes. Jin *et al.*^[6] improved the whale algorithm's local exploration by using a uniform mutation operator and adaptive probability fusion with the Levy flight strategy, but local information feedback can still lead to local optima. Jiang *et al.*^[7] introduced a variable-length coding scheme with structural information to enhance genetic algorithms' local search ability, though increased cross mutation operations caused frequent individual aggregation near current optimal solutions, leading to local convergence. Yang *et al.*^[8] enhanced the Grey Wolf Algorithm (GWO) by adjusting the control factor and adaptively sharing information, improving local search performance. Liao *et al.*^[9] added adaptive crossover probability and random disturbance to the krill swarm (KH) algorithm, improving local search ability. Zeng *et al.*^[10] proposed a hybrid teaching and learning optimization algorithm, integrating cross optimization in the learning stage to balance local search performance. Liao *et al.*^[11] used a variable field of view strategy and genetic algorithm mutation strategy to enhance the polar bear algorithm's local search performance.

Due to the unique search mechanisms and global optimization capabilities of the sparrow search algorithm (SSA)^[12], which are particularly effective for handling complex tasks with many different candidate service sets. Unlike other commonly used optimization algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO), SSA mimics the foraging behavior of sparrows, allowing it to perform effective global searches and local optimizations. Therefore, in this paper, SSA is first introduced into the cloud manufacturing SCOS problem based on service quality perception for optimal solutions. However, when SSA is applied to composition optimization scenarios, the sparrow population needs to frequently randomly search the solution space to find candidate services that meet the task requirements. To address this, a multi-population evolution mechanism is designed, saving the optimal solutions found in the divided subpopulations in a record layer. To enhance the effectiveness of individual local search strategies in

subpopulations, Logistic chaotic mapping is used to improve population diversity in the feasible domain, helping subgroups escape local optima.

To reduce the impact of low population richness and insufficient search traversal on algorithm optimization, and to ensure the quality of the final optimal service composition solution, this paper proposes an improved SSA (ISSA) based composition optimization method for manufacturing cloud services. This method includes updating the position of the finders in the sparrow population through the golden sine strategy of dynamic adaptive weights, better guiding the group to search for the best food source. Later, a differential Cauchy mutation strategy is used to further improve the search ability of the sparrow population. Finally, the global optimal sparrow individual position is updated through a greedy strategy and restored to the corresponding optimal service combination solution, ultimately improving the quality of the service composition solution. Extensive experiments demonstrate the effectiveness of the proposed algorithm in solving multiple benchmark functions and cloud manufacturing service composition problems of different scales.

1 Problem Description and Mathematical Model

1.1 Cloud Manufacturing Service Composition Problem Description

Figure 1 illustrates the main process of a single objective cloud manufacturing service composition based on service quality, including a complex manufacturing task T , which is usually decomposed into a series of subtasks ST , arranged in a logical relationship and corresponding to a candidate manufacturing service set (CMSS). The SCOS problem is that the cloud manufacturing system searches for the corresponding Manufacturing Cloud Service (MCS) for each subtask from the candidate service pool, and then selects a candidate service that meets the needs of each subtask under the multi-dimensional cloud manufacturing background, and aggregates it into a Composite Manufacturing Service (CMS) that meets the task requirements. This article focuses on the service composition and optimization selection stages of this process, and evaluates the quality of the final composite service through multidimensional QoS attribute indicators. The optimal combination path is selected with the goal of maximum QoS.

Considering that each cloud service in the cloud system contains conflicting and diverse QoS attributes, that is,

$$\text{QoS}(\text{MCS}_{i,j}) = \{q_1(\text{MCS}_{i,j}), \dots, q_2(\text{MCS}_{i,j}), \dots, q_r(\text{MCS}_{i,j})\}$$

where $\text{MCS}_{i,j}$ represents the j -th cloud service in the i -th candidate service CMSS selected from the cloud pool, and r represents the r -th dimension of QoS attribute values, including manufacturing time, cost, reliability, and transaction quality.

To meet the changing task requirements, optimizing service quality attributes is essential. Swarm intelligence algorithms, based on biological population behavior, can

be highly effective in this area. However, when faced with complex service composition optimization problems, these algorithms often suffer from reduced convergence rates and solution quality due to their fixed search ranges and localized information transmission. The SSA, with its characteristics of population guidance, following, and early warning, can ensure good local search performance and convergence speed. To address the issues of low population diversity and insufficient search comprehensiveness, this paper proposes a multi-strategy SSA based on chaotic mapping. This approach adapts to complex solution spaces and improves the quality of service composition solutions.

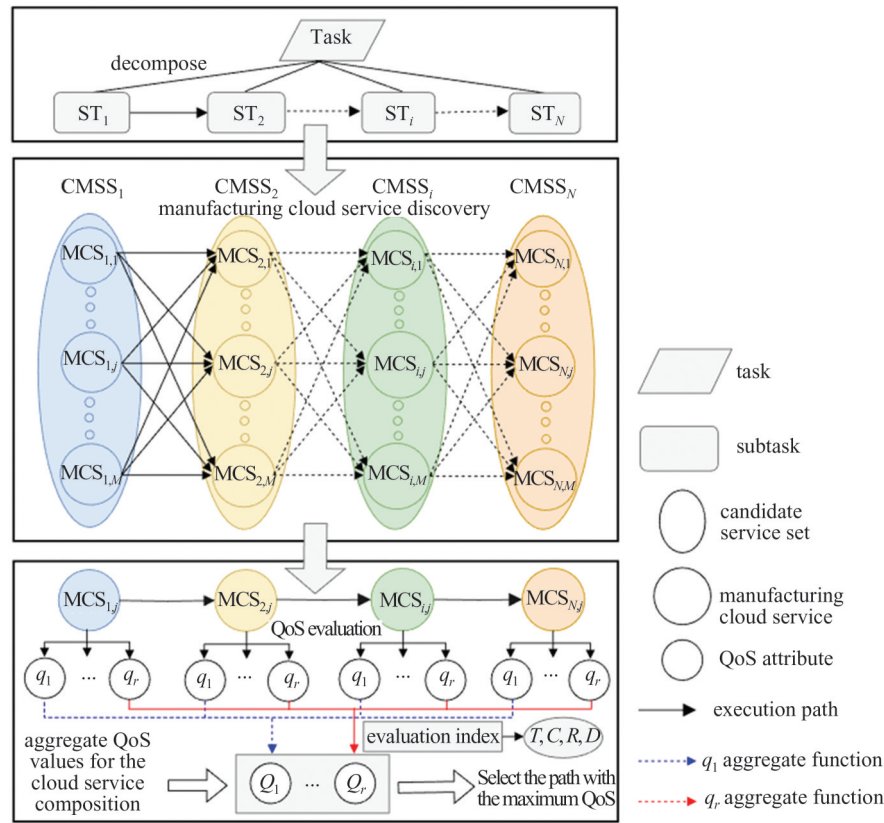


Fig.1 Cloud manufacturing services composition main flowchart

1.2 The Classical QoS Evaluation

Firstly, attribute indicators of manufacturing time, cost and reliability are selected, and then user satisfaction and deliverable quality indicators are further integrated. Considering that the manufacturing cloud service portfolio has four common structures, namely, sequence, parallel, selective, and cyclic^[13], all of which can be simplified into a sequential structure. Therefore, this article uses the aggregation method under the sequential structure for each attribute, as described below.

Manufacturing time T : mainly covering waiting time for processing T_{wait} , customized production time

T_{manu} , and logistics transportation time T_{logis} , namely, $T = T_{\text{wait}} + T_{\text{manu}} + T_{\text{logis}}$. Perform accumulation operation through aggregate function $\sum_{i=1}^n q_T(\text{MCS}_i)$.

Manufacturing cost C : mainly covering processing cost C_{manu} , and transportation costs C_{logis} , namely, $C = C_{\text{manu}} + C_{\text{logis}}$. Perform accumulation operation through aggregate function $\sum_{i=1}^n q_C(\text{MCS}_i)$.

Service reliability R : $R = N_1 / (N_0 + N_1)$, where N_0 and N_1 represent the number of times required for failure and success, respectively. Perform the multiplication op-

eration through aggregate function $\prod_{i=1}^n q_R(MCS_i)$.

User satisfaction S : $S = \sum_{i=1}^n S_i/n$, where S_i represents the i -th service demander obtaining historical service results through score, and n represents the total number of users who provide an evaluated composite service solution. Perform the accumulation operation through aggregate function $\frac{1}{n} \sum_{i=1}^n q_S(MCS_i)$.

Secondly, due to the different forms of QoS expression and quantification units of various manufacturing services, dimensionality elimination and normalization^[14] should be carried out in advance for them, and the specific calculation method used is as follows:

$$\text{QoS}(MCS_{i,j}) = \sum_{q_j \in Q^+} \frac{q_j(MCS_i') - \min q_j}{\max q_j - \min q_j} \omega_i + \sum_{q_j \in Q^-} \frac{\max q_j - q_j(MCS_i')}{\max q_j - \min q_j} \omega_i \quad (1)$$

where, $\max q_j$ and $\min q_j$ represent the maximum and minimum values of the j -th attribute of the candidate service owned by the i -th manufacturing cloud service, respectively, Q^+ , Q^- represent the positive and negative attribute parameters respectively.

1.3 Comprehensive Mathematical Model

According to the analysis in Section 2.1, this paper constructs a problem model with the optimization goal of maximizing QoS, and uses a simple weighting method to obtain the comprehensive utility value (objective function), which can be calculated by formula (2).

$$\max(\text{QoS}) = \max \sum \omega_k \times \text{Norm}Q_k, \sum_{k=1}^r \omega_k = 1 \quad (2)$$

where ω_k is the weight of the k -th evaluation indicator and $\omega_k \in [0, 1]$, $\text{Norm}Q_k$ is the normalized value of the k -th QoS attribute, and r is the number of indicators. The value can be specified based on user preference or system.

Considering the specific requirements of the production and manufacturing process and the characteristics of the service composition problem, the following constraints are provided:

$$\begin{bmatrix} \frac{1}{T(MCS_{i,j})} \\ \frac{1}{C(MCS_{i,j})} \\ R(MCS_{i,j}) \\ S(MCS_{i,j}) \\ D(MCS_{i,j}) \end{bmatrix} \geq \begin{bmatrix} \frac{q_T}{T_{\text{base}}} \\ \frac{q_C}{C_{\text{base}}} \\ q_R R_{\text{base}} \\ q_S S_{\text{base}} \\ q_D D_{\text{base}} \end{bmatrix}, \forall j = 1, \dots, N \quad (3)$$

where $r_{i,j}$ represents the j -th candidate service in the execution path of the i -th composite service. T_{base} , C_{base} , R_{base} , S_{base} , D_{base} respectively represent the worst-case level that users can accept in terms of execution time, cost, reliability, satisfaction, and quality of deliverable, and N indicates the total number of cloud services.

2 The Basic Concept of SSA Algorithm

The SCOS is a typical dynamic and uncertain composition optimization problem. Swarm intelligence algorithms are well-suited for solving such problems due to their adaptability and robustness. Recently, a new intelligent algorithm, the Sparrow Search Algorithm (SSA)^[12], has garnered significant attention.

2.1 The Principle of Algorithm and Mathematical Model

SSA can establish a discoverer-joiner mathematical model based on the foraging process of sparrows. The discoverer will be the first to gain energy in the search for resources, and at the same time to guide the followers to the foraging location. The location formula is as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot T_{\text{max}}}\right), & R_2 < ST \\ X_{i,j}^t + Q \cdot L, & R_2 \geq ST \end{cases} \quad (4)$$

where $R_2 \in [0, 1]$, $ST \in [0.5, 1]$ represent the warning line and safety limit, which is uniformly distributed and generated arbitrarily within the range of (0,1), Q is any number that satisfies the normal law, and L is a 1 by d matrix.

Followers closely follow the discoverer to complete the food search process, and at the same time try their best to grab the discoverer's resources to improve their fitness value. The mathematical model is shown in formula (5):

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{\text{worst}}^t - X_{i,j}^t}{i^2}\right), & i > \frac{N}{2} \\ X_{\text{pbest}}^{t+1} + |X_{i,j}^t - X_{\text{pbest}}^{t+1}| \cdot A^+ \cdot L, & i \leq \frac{N}{2} \end{cases} \quad (5)$$

where X_{worst}^t , X_{pbest}^{t+1} correspond to the global worst and local best regions in the current iteration and the next iteration respectively, and $A^+ = A^T(AA^T)^{-1}$ is a matrix that satisfies and all values are 1 or -1.

The sparrows responsible for monitoring and warning only account for 10% and 20% of the whole population. Their task is to spread the warning signal to the entire population, while leading the population to quickly move to a safe area or randomly approach other sparrow individuals. Their position updates are shown in equation (6):

$$X_{ij}^{t+1} = \begin{cases} X_{gbest}^t + \beta \cdot |X_{ij}^t - X_{gbest}^t|, & f_i \neq f_g \\ X_{ij}^t + K \cdot \left(\frac{|X_{ij}^t - X_{worst}^t|}{(f_i - f_w) + \varepsilon} \right), & f_i = f_g \end{cases} \quad (6)$$

where X_{gbest}^t represents the position of the globally optimal individual, β is a random number and follows a normal distribution, $K \in \text{rand}[-1, 1]$ is a control step parameter, f_g, f_w are the optimal and worst fitness values, respectively, ε is the infinitesimally small constant that regulates the wandering area of sparrows.

2.2 Analysis of Algorithm Advantages and Disadvantages

As a new heuristic algorithm, the SSA features fast convergence speed, easy implementation, few adjustable parameters, and excellent local search performance. It has been successfully applied to various engineering optimization problems in recent years, such as three-dimensional path planning for unmanned aerial vehicles (UAVs)^[15] and reactor parameter optimization^[16].

However, there are still some shortcomings when applying SSA to service composition optimization. One issue is the decreased population diversity. The random initialization of the sparrow population in the search space results in an uneven distribution, limiting the full utilization of environmental information. Unlike PSO and ACO, which gradually move towards the optimal solution, SSA converges directly to the current optimal solution, leading to a rapid decrease in population diversity in the later stages. Additionally, there is insufficient search traversal for discoverer location updates. Discoverers occupy positions with high current fitness values, leading the population based on the best fitness individuals. This method only considers current and optimal positions without incorporating the individual's previous experience, resulting in premature convergence and a limited search scope. These issues contribute to poor convergence accuracy and solution quality, especially for high-dimensional problems.

3 Multi-Strategy ISSA Based on Chaotic Mapping

3.1 Overview of the ISSA

Definition 1 The sparrow individual in ISSA can be represented by the quintuple $S = \{t, T, f(X), X_{best}^*, f(X_{best}^*)\}$, and $X = \{x_1, x_2, \dots, x_d, \dots, x_n\}$, where x_d represents the distribution position of individuals in the D -dimensional solution space, $f(X)$ is the fitness function value, which can judge the quality of each individual in all distributed populations of the sparrow algorithm, and $X_{best}^*, f(X_{best}^*)$ is the optimal individual obtained after updating and its fitness value. The sparrow population is an $n \times d$ vector composed of n individual sparrows, and each sparrow has the same iteration number, fitness function and iterative process of position update.

Definition 2 The i -th sparrow is represented as $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,j}, \dots, x_{i,N})$, where $x_{i,j}$ represents the serial number of the manufacturing cloud service selected in the j -th candidate service set in the composite service, with a range limited to $\{x_{i,j} : 0 \leq x_{i,j} \leq K_j\}$, and K_j represents the total number of cloud services in the j -th sub-task's candidate service set.

Specifically, this article visualizes the mapping relationship between candidate cloud service sets and sparrow individuals using Fig. 2. If the set of CMSS to be selected is composed of $\{MCS_{1,4}, MCS_{2,6}, MCS_{3,5}, MCS_{4,1}, MCS_{5,2}, MCS_{6,3}\}$, it means that the composite solution contains six different types of ser-

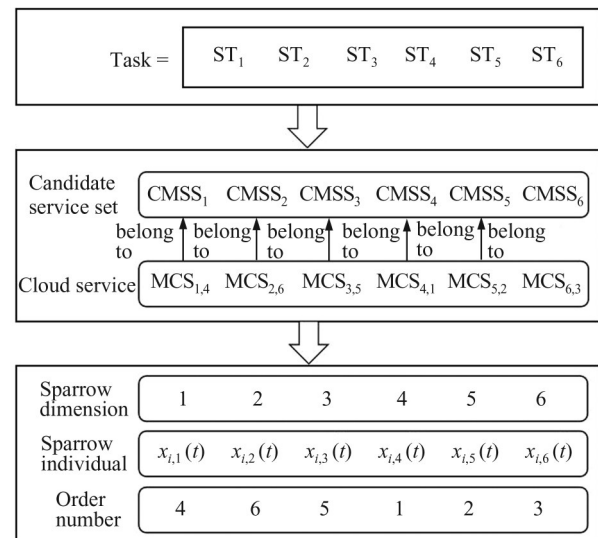


Fig.2 Mapping relationship between composite services and sparrow individuals

vices that have been selected. The first position indicates that the subtask 1 is executed by the fourth manufacturing service instance in its corresponding candidate service set, and the other positions are also the same. Then the final corresponding individual coordinates can be represented as a real number array (4, 6, 5, 1, 2, 3).

As a whole, the ISSA includes three major improvement designs: multi-population evolution mechanism, dynamic global search strategies for discoverers, and mutation perturbations for position update. Specifically, ① Considering the mapping relationship between sparrow individuals and composite services, the original fixed search space is divided into subspaces, and the sparrow population searches for optimal solutions in each subspace in parallel and combines into the best solution required. In order to reasonably allocate the population to search for the target solution in the subspace and improve the traversal accuracy, multi-population evolution and chaotic disturbance search mechanisms are used; ② In response to the situation where discoverers blindly gather too early, the global capture strategy of adaptive weight and golden sine is introduced to better stabilize the exploration performance of the algorithm; ③ In order to solve the problem of insufficient search scope when the search individuals update their positions around elite individuals, cross-variation with reference to the differential strategy makes the diversity of sparrow population more abundant. Meanwhile, the Cauchy operator is used in the mutation process to continuously perturb the optimal solution during the evolution process, and further enhance the global optimization ability.

3.2 Multi-Group Evolution Mechanism

From the overall overview of the ISSA in Section 3.1, it can be seen that when selecting suitable cloud services from a large number of candidate service sets, the ISSA abandons the traditional fixed solution space single search method. Instead, it divides the population into different subspaces corresponding to the actual number of candidate service sets and adopts parallel searches for the current optimal solution. This poses a challenge to the development and exploration process of the sparrow population in the algorithm. To reasonably allocate the population to search for target solutions in subspaces and improve optimization efficiency, this paper designs a mechanism for parallel evolution from multiple populations. The search strategy is described as follows:

1) Subgroup Division: The sparrow population is

evenly divided into n subgroups ($sb_1, sb_2, \dots, sb_n$), each corresponding to m candidate service sets divided by tasks. This ensures that a large sparrow population can be divided into several independent small populations, improving the parallel search efficiency of individuals in different subgroups.

2) Parallel Search: Each subgroup performs a parallel search for the current optimal solution within its designated subspace. The optimal individual P_b in each sparrow subgroup guides other individuals in the group by sharing its advantageous experience. This ensures that feasible solutions can be found among the numerous candidate solutions.

3) Optimal Individual Retention: After the evolution of any subgroup, the optimal sparrow individual P_b in the subgroup is retained, and its position is stored in their respective message blocks ($M_1, M_2, \dots, M_n$). As indicated by step ① in Fig. 3, the positions of optimal individuals from all subgroups are combined into a preferred decision layer.

4) Preferred Decision Layer: This layer, indicated by step ② in Fig. 3, is responsible for recording feasible solutions selected from different subgroups. It contains the optimal individuals of all subgroups, ensuring that the best solutions are retained and refined throughout the iterations.

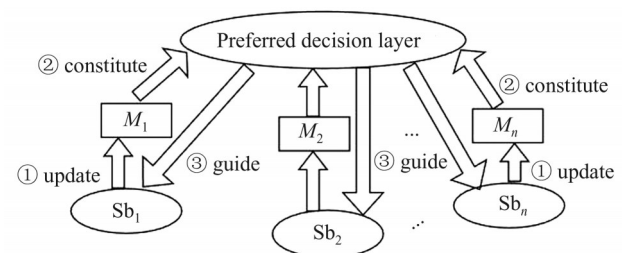


Fig.3 Schematic diagram of the evolution mechanism of multiple populations

In order to ensure the effectiveness of individual local search strategies in the subpopulation, the use of Logistic chaotic mapping with improved population diversity in the feasible domain helps subgroups escape the dilemma of local optimization, ensuring that the algorithm searches for feasible candidate services for on-demand combinations. To reduce the impact of low population richness and insufficient search traversal on algorithm optimization and ensure the quality of the final optimal service composition solution, this paper takes service quality as the evaluation indicator. An

ISSA-based composition optimization method for manufacturing cloud services is proposed. This method specifically includes updating the position of the finders in the sparrow population through the golden sine strategy of dynamic adaptive weights to better guide the group to search for the best food source and later using the differential Cauchy mutation strategy to further improve the search ability of the sparrow population. Finally, the global optimal sparrow individual position is updated through a greedy strategy, and it is restored to the corresponding optimal service combination solution, ultimately improving the quality of the service composition solution.

Based on the information recorded in the preferred decision layer, it is determined that if the value of the i -th record board does not change for six consecutive iterations, the subgroup is likely performing redundant iterations near a local extremum due to decreased individual competitiveness, ultimately falling into the local extreme range. To encourage each subgroup to continue searching for feasible solutions and to avoid premature convergence, more optimal individuals are selected from the preferred decision layer to guide the underperforming subgroup by learning from their experiences multiple times, corresponding to position ③ in Fig. 3. Position changes are then performed by formulas (4)-(6) to better optimize the search area.

Since dominant sparrows from all subgroups are stored in the preferred decision layer, it is necessary to guide those in the underperforming subgroup to learn from the search experiences of these dominant individuals. This guidance helps other individuals in the subgroup to continue absorbing information from better-performing individuals, ensuring that all populations are actively engaged in iterative evolution. If there are no better individuals in the preferred layer to provide effective information, and a subgroup falls into a local optimum again, the subgroup will experience evolutionary stagnation. To address this, chaotic sequences are selected to perturb the subgroup. Chaotic sequence mapping has the characteristics of randomness and sensitivity to initial values. Logistic mapping, a typical representative of chaotic mapping, is widely used in optimization search problems due to its uniform sequence distribution.

$$Y_{n+1} = \mu Y_n (1 - Y_n) \quad (7)$$

where Y_n is a chaotic sequence, which indicates the n th chaotic sequence value generated, and $0 < \mu \leq 4$ is the control parameter of the chaotic sequence.

By integrating chaotic perturbations, the ISSA enhances the exploration capabilities of subgroups, helping them escape local optima and maintain diversity in the search process. This method ensures a more thorough and effective optimization, ultimately improving the quality of the service composition solution.

As shown in Fig. 4, the horizontal axis represents the control parameter, the vertical axis represents the range of changes in the Logistic sequence values. When the control parameter μ is changed while the sparrow is fixed, it is found that the chaos state is better when $\mu \in [3.85, 4]$ and the μ is closer to 4. Therefore, in order to enable sparrow individuals trapped in local extreme regions to explore other effective search regions as much as possible through perturbation for traversing more service components, this article sets the parameter μ to 4.

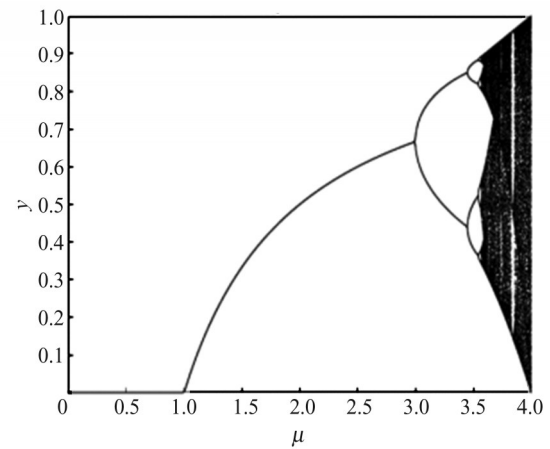


Fig.4 Logistic chaotic map bifurcation graph

In order to enhance the search diversity of the sparrow population, all the individuals in the subgroup are first arranged in increasing order of fitness values, with higher positions indicating more advantageous individuals. Next, the top 10% of sparrows in the subgroup are selected, representing those with the highest fitness values. These top-performing sparrows gradually replace the bottom 10% of sparrows in the subgroup which are the individuals with the lowest fitness values.

After this replacement process, Logistic chaos is used to implement a one-dimensional perturbation process on the updated population to further diversify the search. This process is described by equation (8), which introduces chaotic perturbations to the positions of the individuals, ensuring a more thorough exploration of the solution space and preventing premature convergence.

$$X_{i,d} = \text{Lb}_d + \frac{(\text{Ub}_d - \text{Lb}_d)}{2} (1 + Y_{i,d}) \quad (8)$$

where, Ub_d represents the upper bound in the d -dimension for the current optimal solution, Lb_d represents the lower bound in the d -dimension for the current optimal solution, and i represents the dimension that needs to be perturbed, which is randomly selected. This strategy mainly perturbs the current optimal solution with a one-dimensional logistic chaotic sequence.

The single-dimensional perturbation strategy based on the multi-population evolution mechanism and Logistic chaos is specifically designed for the single search space of service composition optimization problems. This strategy has a dual purpose. On one hand, it retains most of the excellent dimensional information of the current optimal solution during the service composition process, ensuring the algorithm's convergence to the service composition optimization problem. On the other hand, it utilizes the Logistic sequence and multi-population evolution mechanism to perturb the single dimension of the current optimal solution.

As the sparrow algorithm iterates, this self-adjustment strategy within the subgroups helps them escape the constraints of local optima. By continuously introducing chaotic perturbations, the strategy enhances the global optimization ability of the algorithm. This approach not only improves search accuracy but also ensures convergence stability across different subspaces. Through these mechanisms, the ISSA effectively balances exploration and exploitation, ultimately leading to more robust and reliable service composition solutions.

3.3 Discoverer's Dynamic Global Search Strategy

Due to the significant impact of inertia weight on the iterative update of sparrow search algorithms, this article introduces an adaptive weight factor for the position of the original sparrow discoverer, as expressed as follows:

$$\omega = (f_{\text{best}} - f_{\text{worst}}) \cdot \frac{\cos\left(\frac{\pi t}{T_{\text{max}}} + \omega_{\text{max}}\right)(\omega_{\text{max}} - \omega_{\text{min}})}{2} + 1 \quad (9)$$

where f_{best} , f_{worst} are the best and worst fitness values respectively, ω_{max} , ω_{min} are the maximum value and minimum value of the weight factor, respectively. Multiple experiments have shown that the effect reaches its maximum when $\omega_{\text{max}} = 0.85$, $\omega_{\text{min}} = 0.25$.

It can be seen from Fig. 5 that in the early iteration stage of the sparrow individuals, they approach the

global optimal at a faster speed. In the middle and later stages of the iteration, the decline gradually begins to slow down, so as to facilitate the subsequent more accurate mining, and finally achieve the effect of improving the convergence speed and global space exploration capability.

Then the weight factor is combined with gold sine to improve the discoverer's position. The updated formula is as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot |\sin r_1| - \omega \cdot r_2 \cdot \sin r_2 \cdot |\beta_1 X_{\text{best}}^t - \beta_2 X_{i,j}^t|, R_2 < ST \\ X_{i,j}^t + Q \cdot L, R_2 \geq ST \end{cases} \quad (10)$$

where r_1 is the number generated randomly within $[0, 2\pi]$, and r_2 is the number generated randomly within $[0, \pi]$, β_1, β_2 are parameters constructed using the golden section coefficient, which will cooperate with the weight factor to ensure the uniform effect of the overall search of the algorithm and further improve the convergence speed of the algorithm. Among them $\beta_1 = -\pi + 2\pi \cdot (1 - \tau)$, $\beta_2 = -\pi + 2\pi \cdot \tau$, $\tau = (\sqrt{5} - 1)/2$.

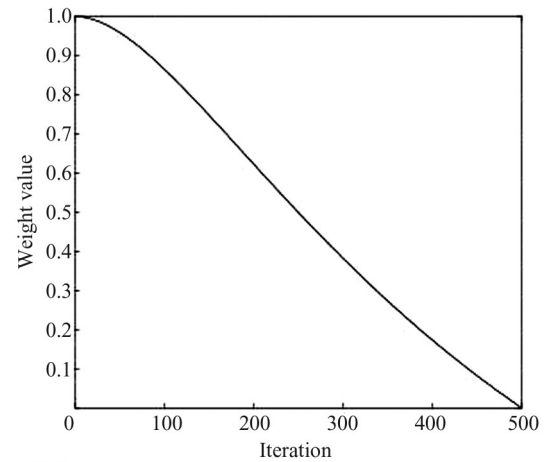


Fig.5 Adaptive weight factor iteration diagram

3.4 Design of Optimal Position Variation Disturbance

The differential evolution strategy, an evolutionary algorithm proposed by Storn *et al*^[17], is based on the genetic algorithm. It generates individuals with variations from the original population and then performs cross-over mutations to enrich population diversity. This paper adopts this strategy within the SSA. The specific process is as follows: First, two different sparrows are randomly selected from the entire population. The position changes of these two sparrows are calculated. Next, these changes are perturbed with the globally optimal individual to obtain a mutated new individual. This pro-

cess introduces new genetic material into the population, helping to maintain diversity and avoid premature convergence.

By incorporating differential evolution into SSA, the algorithm benefits from enhanced exploration capabilities, ensuring a more thorough search of the solution space and improving the overall optimization performance. The mutation expression is as follows:

$$X_{\text{new}}^{t+1} = X_{\text{best}}^{t+1} + P \cdot X_{r1}^t - X_{r2}^t \quad (11)$$

where X_{r1}^t , X_{r2}^t represent two sparrows chosen at random, $P \in (0, 6)$ represents the scaling parameter, X_{best}^t indicates the position of the sparrow with the best fitness value in the current population.

In order to further enhance the optimization effect of SSA and prevent the algorithm from falling into the dilemma of local optimal solution due to a sharp decrease in population diversity during the search process, the Cauchy distribution strategy is used to perturb the position of sparrows:

$$X_{\text{new}}^{t+1} = X_i^t + t(\text{iter}) \cdot X_i^t \quad (12)$$

where X_i^t denotes the position of the i -th sparrow individual, X_{new}^{t+1} represents the position of the individual after mutation, and $t(\text{iter})$ represents the t distribution with the current iteration number iter as the degree of freedom.

Although the above mutation strategy can increase the chances of escaping local optima, it does not guarantee that the newly generated sparrow individual will be better than the original optimal individual. Therefore, we incorporate a greedy approach to ensure that individuals with higher fitness values are selected at each step. Specifically, after generating the new mutated individual through differential evolution, the fitness values of the new and original individuals are compared. The individual with the larger fitness value is retained for the next iteration. This greedy strategy ensures that the population continually improves, enhancing the overall optimization process and ensuring a convergence to a better solution. The specific selection and updating operation are as follows:

$$X_{\text{best}} = \begin{cases} X_{ij}^{t+1}, & f(X_{ij}^{t+1}) \geq f(X_{\text{best}}) \\ X_{\text{best}}, & f(X_{ij}^{t+1}) < f(X_{\text{best}}) \end{cases} \quad (13)$$

3.5 Analysis of Composition Optimization of Cloud Manufacturing Services Based on ISSA

According to above description, the algorithm flow of cloud manufacturing service composition optimization based on the ISSA is given and described in Algorithm 1.

Algorithm 1 ISSA framework for solving the cloud manufacturing service composition problem based on service quality

Input: population size (PopSize), spatial search dimension (Dim), maximum number of iterations (MaxIter), proportion of subscribers (PD), proportion of alert value (SD) and alert threshold (R_2), the objective function value ($\max \text{QoS} = \max \sum_{k=1}^5 \omega_k \times Q_k$)

Output: output the optimal sparrow position and its fitness value

1. **While** $t \leq \text{MaxIter}$ **do**
 2. Calculate the QoS value of composite services in the group PopSize according to equation (2), sort the value and save the optimal QoS value.
 3. Obtain the fitness value of the current sparrow population and decrease it in order, and then get the current best fitness and its corresponding position, as well as the current worst fitness and its corresponding position.
 4. Divide the population into different sub-groups, and follow equation (8) to perturb the replaced new subgroup using the logistic chaotic sequence.
 5. $R_2 = \text{rand}(1)$
 6. **for** $i = 1:\text{PD}$ **do**
 7. The sparrow with the best fitness value will be the discoverer through weight distribution, and its position will be updated according to formula (10).
 8. **end for**
 9. **for** $i = (\text{PD} + 1):\text{PopSize}$ **do**
 10. Use the surplus sparrows in the population as followers and update their positions according to formula (5).
 11. **end for**
 12. **for** $i = \text{SD}:\text{PopSize}$ **do**
 13. Update the position of the watcher according to formula (6).
 14. **end for**
 15. According to the current fitness value and average fitness value, differential variation strategy or Cauchy distribution perturbation strategy is selected to disturb the current optimal solution, generate new solutions.
 16. Update the fitness value through the greedy formula (13), and record the global optimal position and the sparrow's individual optimal position information.
 17. **end while**
 18. **return** the optimal location and its optimal fitness value to obtain the optimal QoS value
-

4 Simulation Results and Experimental Analysis

4.1 Performance Verification of the Algorithm on Benchmark Problems

In order to verify the practicability of the proposed

method, it is applied to solve six classical standard reference functions, namely Sphere (f_1), Schwefel 2.22 (f_2), Schwefel 2.21 (f_3), Griewank (f_4), Alpine (f_5) and Shekel (f_6), including single mode and multimodal functions, see Table 1. The term $f_{\min}$ refers to the theoretical minimum value of the objective function, which is the best solution that the algorithm aims to approximate or achieve. The functions f_1 - f_3 are unimodal and have only one global optimal value, so the convergence speed of the algorithm is more important than the optimization results. The functions f_4 - f_6 are multimodal functions that have many local optima. Therefore, the importance of optimization results is greater than that of the convergence speed, which can reveal the algorithm's ability to break free from local optima and find global optima.

For the benchmark function, the proposed ISSA is compared with three meta-heuristic methods, including Butterfly Optimization Algorithm (BOA), Grey Wolf Optimization Algorithm (GWO), and SSA. The parameters of GWO are the initial and final control parameters, i.e., $a_{\text{initial}}=2$, $a_{\text{final}}=0$; BOA is the switching probability $P=0.8$, sensory mode $c=0.001$, $a=0.1$; The parameter of SSA is PD=0.2, SD=0.2, ST=0.8; The same parameters in the four algorithms are set identically: that is, the population size (Popsiz) is 30, and the maximum running times (MaxIter) is 500.

4.1.1 Analysis of search accuracy

In this paper, ISSA and the comparison algorithm proposed in this paper are independently run 30 times on each benchmark function. The optimal value, average value, standard deviation, and worst value are used as

evaluation criteria for the performance of each algorithm. These metrics are chosen to comprehensively assess the accuracy and stability of the algorithms. Specifically, the optimal value represents the best objective function value achieved, the worst value represents the worst objective function value achieved, the average value represents the mean objective function value over multiple runs, and the standard deviation indicates the variability and consistency of the results. Finally, the data generated by each algorithm will be recorded in an exponential manner, highlighted in black and bold to highlight the best results, and the average time spent at the end of each run process will be recorded. The comparison of the results of the benchmark function is shown in Table 2.

As can be seen from Table 2, the ISSA shows a better convergence rate compared with the other three algorithms when facing the first three unimodal test functions, and their optimal values and mean values are both 0. It can be seen that the chaotic initialization population plays a crucial role, which greatly improves its convergence rate, and thus verifies that the improved algorithm has good stability. For multimodal test functions, although the original algorithm and the improved algorithm do not differentiate when dealing with function f_4 , the algorithm in this paper still has great advantages based on the comparison of the solution performance of function f_5 . Finally, for the multimodal test function f_6 , the proposed algorithm has strong robustness in searching the entire solution space. This is be-

Table 1 Test functions description

Function	Name	Dimension	Range	$f_{\min}$
$f_1 = \sum_{i=1}^n x_i^2$	Sphere	30	$[-100,100]$	0
$f_2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	Schwefel 2.22	30	$[-10,10]$	0
$f_3 = \max_i \{ x_i , 1 \leq i \leq n\}$	Schwefel 2.21	30	$[-100,100]$	0
$f_4 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \frac{x_i}{\sqrt{i}} + 1$	Griewank	30	$[-600,600]$	0
$f_5 = \sum_{i=1}^d x_i \sin(x_i) + 0.1x_i $	Alpine	30	$[-10,10]$	0
$f_6 = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6})^{-1}$	Shekel	30	$[-65,65]$	1

cause the addition of differential mutation strategy and Cauchy perturbation in the later stage helps the algorithm successfully overcome the local extremum dilemma and obtain the global optimal solution. Although all four algorithms are extremely close to the theoretical values, further analysis of their respective mean and standard deviation reveals that ISSA has significantly better overall stability than other algorithms.

4.1.2 Analysis of the rate of convergence

Considering that the convergence speed of each algorithm cannot be directly reflected only by the average value and standard deviation, the convergence curve is further drawn to better show the process of each algorithm converging to the optimal solution.

For different types of test functions, it can be seen

from Fig. 6(a) to Fig. 6(c) that when the four algorithms converge to the same fitness, the ISSA takes the least number of iterations and the fastest rate of convergence, which is due to the early multi-population mechanism and Logistic one-dimensional chaotic perturbation. Observing that the curves of GWO and BOA gradually tend to moderate, and finally there will be stagnation, that is, they fall into a local optimum, while the convergence trend of the ISSA shows a gradient wave drop overall, indicating that the differential Cauchy disturbance in the later stage helps the algorithm successfully get rid of the local extreme value dilemma. As the process progresses, for these benchmark functions with local optima, the ISSA has a stronger global search ability than other algorithms.

Table 2 Results of benchmark functions

Function	Algorithm	Best	Worst	Avg	Std
f_1	BOA	5.96E-11	8.10E-11	6.89E-11	5.72E-12
	GWO	1.76E-35	1.67E-32	2.19E-33	3.37E-33
	SSA	5.23E-158	7.37E-122	4.77E-123	1.57E-122
	ISSA	0	0	0	0
f_2	BOA	9.59E-09	2.99E-08	2.15E-08	5.79E-09
	GWO	1.20E-20	3.49E-19	6.83E-20	7.24E-20
	SSA	4.52E-96	2.72E-72	1.01E-73	4.99E-73
	ISSA	0	7.71E-195	2.57E-196	0
f_3	BOA	1.65E-06	2.53E-06	2.09E-06	2.31E-07
	GWO	2.39E-09	8.55E-08	2.81E-08	2.27E-08
	SSA	0	6.06E-62	2.02E-63	1.11E-62
	ISSA	0	0	0	0
f_4	BOA	6.27E-13	3.02E-11	9.54E-12	7.19E-12
	GWO	0	3.65E-02	3.93E-06	9.78E-03
	SSA	0	0	0	0
	ISSA	0	0	0	0
f_5	BOA	7.11E-11	1.43E-08	1.17E-09	2.97E-09
	GWO	4.38E-20	1.59E-03	3.10E-06	4.33E-04
	SSA	0	3.11E-24	1.03E-25	5.68E-25
	ISSA	0	2.92E-277	9.76E-279	0
f_6	BOA	0.998	1.050	1.004 3	1.33E-02
	GWO	0.998	12.670	3.455 0	3.26
	SSA	0.998	12.670	1.642 0	2.51
	ISSA	0.998	0.890	0.998 0	2.48E-10

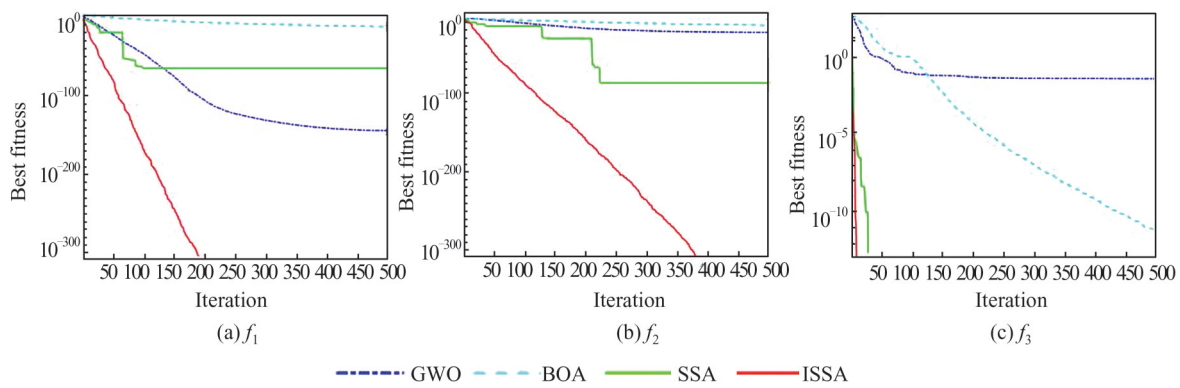


Fig.6 Convergence curves for benchmark functions

To sum up, the results in this section show that the ISSA is highly competitive with high-dimensional single-mode functions and multi-mode functions. The ISSA retains the unique guidance mechanism of followers and the warning mode of scouts. Meanwhile, the ISSA uses the golden sinusoidal fusion dynamic factor to improve the position update stage of discoverers in the SSA. Finally, Cauchy differential perturbation is used to enhance the optimization ability of the algorithm. This not only maintains population diversity, but also improves the optimization efficiency and accuracy of the algorithm.

4.2 Performance Verification of the Algorithm on Composition Optimization Problem

To further demonstrate the practicality and rationality of the ISSA in cloud manufacturing service composition, the cost, time, reliability, satisfaction and quality of deliverables of each manufacturing cloud service are randomly generated between $[0.8, 0.95]$. Their weights in customer QoS preferences are respectively set as $\omega_{\text{time}} = 0.3$, $\omega_{\text{cost}} = 0.2$, $\omega_{\text{rel}} = 0.2$, $\omega_{\text{sta}} = 0.15$, $\omega_{\text{del}} = 0.15$. The parameter settings for each algorithm remain the same as in Section 4.1, with a specified number of runs of 200. Finally, a series of experiments are conducted to verify the effectiveness and efficiency of the algorithm in this paper.

4.2.1 Convergence analysis

This section sets the number of subtasks to 30-50, with an increment of 10. The number of candidate manufacturing cloud services for each subtask ranges from 150 to 300 in an increment of 150. For example, T-30-150 indicates that the number of subtasks (abstract services) is 30 and the number of candidates MCS for each subtask (candidate services) is 150.

It can be seen from Table 3 that the average fitness

value obtained by the ISSA is superior to other comparison algorithms, and the optimal solution obtained by the ISSA is also better than other comparison algorithms. Obviously, as the number of subtasks and candidate services increases, the optimal solutions of all four algorithms have decreased, which may be related to the complexity of the solution space of the problem. Based on various comparisons, it can be concluded that the performance of the ISSA in this paper outperforms the other three comparison algorithms.

Table 3 Performance of algorithms for different problems

Problem	Index	GWO	BOA	SSA	ISSA
T-30-150	Best	0.621 826	0.633 259	0.643 377	0.655 992
	Mean	0.603 508	0.619 831	0.619 323	0.643 833
	Std	0.019 493	0.001 560	0.002 281	0.001 814
T-30-300	Best	0.612 089	0.624 239	0.636 423	0.647 660
	Mean	0.599 853	0.600 516	0.613 575	0.625 253
	Std	0.002 114	0.002 128	0.001 495	0.001 397
T-40-150	Best	0.613 625	0.620 262	0.637 534	0.647 132
	Mean	0.572 418	0.596 248	0.613 721	0.623 557
	Std	0.027 369	0.020 129	0.002 645	0.001 826
T-40-300	Best	0.603 147	0.615 891	0.626 173	0.648 037
	Mean	0.562 496	0.599 357	0.604 606	0.615 540
	Std	0.028 836	0.021 302	0.001 019	0.001 550
T-50-150	Best	0.593 338	0.614 630	0.617 572	0.646 715
	Mean	0.481 938	0.572 384	0.593 959	0.601 124
	Std	0.028 486	0.002 192	0.002 004	0.021 308
T-50-300	Best	0.582 143	0.605 041	0.617 277	0.647 724
	Mean	0.492 829	0.565 409	0.604 414	0.615 193
	Std	0.026 441	0.028 065	0.001 627	0.001 448

Then Figure 7 plots the optimal fitness value obtained by four algorithms in solving 50 manufacturing subtasks and the required number of iterations.

Under the same conditions, the optimal fitness value obtained by the comparison algorithm is small, and the overhead of the number of iterations is relatively high, which is likely to be affected by the population position. The ISSA can obtain better optimal solutions while guaranteeing the number of iterations, introduce multiple swarm mechanisms and Logistic chaotic perturbation strategies to ensure the diversity of the algorithm's early population and the balance of exploration and development in the later stage. This indicates that the ISSA has more accurate optimization results than other algorithms, because it expands more search ranges during the evolution process to enhance its possibility of jumping out of local optimal. In summary, the ISSA is superior to the other four algorithms in terms of overall search efficiency and solving quality.

4.2.2 Time performance analysis

Finally, in order to further effectively compare the average running time required by the four algorithms in solving the manufacturing cloud service composition model based on service quality, this paper conducts experiments in two scenarios respectively. In the first scenario, the number of candidate services is guaranteed to be 200, and the number of manufacturing tasks is gradually increased from 20 to 200. In the second scenario, the number of manufacturing tasks remains the same at 50, and the number of candidate services changes from 50 to 500, giving the time trend graph in Fig. 8.

As can be seen from Fig. 8(a), when fewer manufacturing tasks are arranged, all four algorithms spend relatively less time. However, as manufacturing tasks increase, the gap between them gradually widens. Especially for the GWO, its running time shows an exponential growth trend. This is because the increase of the

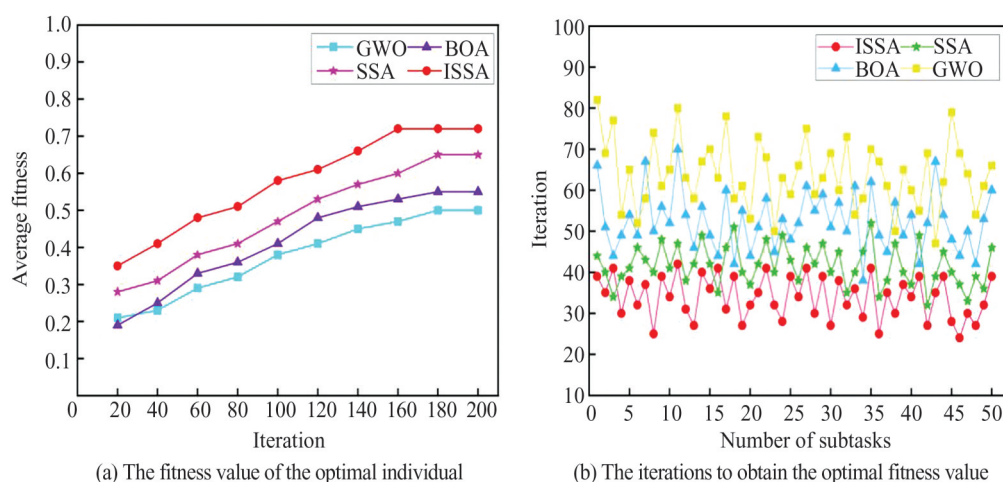


Fig.7 Optimal fitness values of algorithms

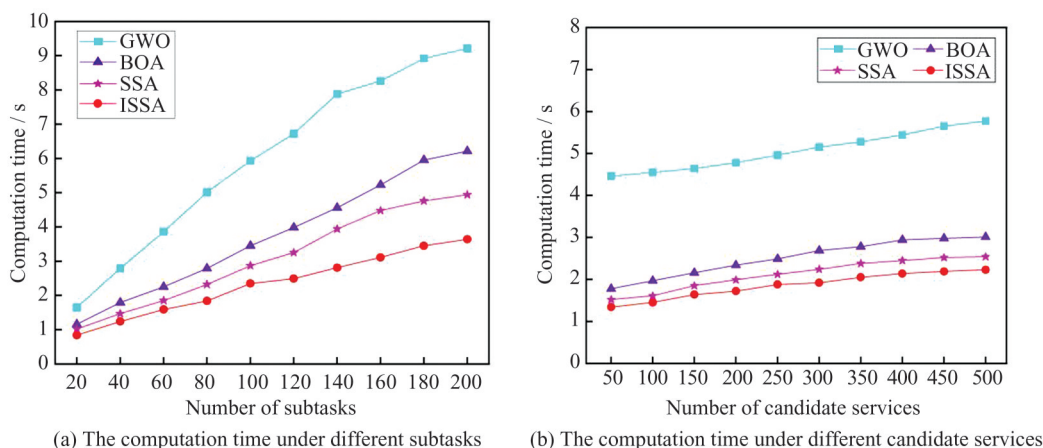


Fig.8 Comparison of the computation time of four algorithms in different scenarios

number of subtasks will lead to the increase of the dimensionality of the target vector, which will make the calculation of the QoS fitness function more complicated, and thus bring additional time cost. By observing Fig. 8(b), it can be observed that increasing the number of candidate services does not have a significant impact on the four algorithms, as increasing the number of candidate services will expand the solution space but will not affect the optimization efficiency. From a global perspective, the ISSA has significantly less solving time than other algorithms. This is because the use of a Logistic single dimensional perturbation strategy in sparrow individuals lacking diversity in each subpopulation further improves the convergence accuracy and optimization speed of the algorithm. Moreover, the difference variation strategy and Cauchy distribution disturbance are added later to help the algorithm successfully get rid of the local extreme value dilemma. The stronger the global detection ability of the algorithm, the better the results.

5 Conclusion

This article addresses the current challenges in cloud manufacturing service composition by improving the SSA to solve the optimal portfolio solution. To counter the disadvantages of loss of population diversity and lack of global optimization ability in SSA, we introduce a multi-population evolution mechanism and Logistic chaotic mapping to enhance population diversity in the early stages of the search. Additionally, a global capture strategy based on dynamic weight factors and the golden sine method is employed to stabilize the algorithm's exploration performance. In the later stages, differential Cauchy mutation perturbation is used to further enhance the algorithm's global optimization ability.

The superiority of the proposed algorithm is verified through benchmark function problems and service composition scenarios. Results show that the improved algorithm not only reduces computing time but also ensures convergence accuracy, demonstrating the correctness and effectiveness of the proposed model and algorithm in solving cloud manufacturing service composition optimization problems. Although the improved algorithm can solve the issues in the cloud manufacturing service portfolio to a certain extent, it typically results in suboptimal solutions, requiring compromises between ef-

fectiveness, utility value, and computing time. In the future, exploring the use of mature algorithms from other fields to solve the cloud manufacturing service portfolio problem could be a highly effective approach.

References

- [1] Li B H, Chai X D, Hou B C, *et al.* Cloud manufacturing system 3.0—New intelligent manufacturing system in the era of "intelligence+"[J]. *Computer Integrated Manufacturing Systems*, 2019, **25**(12): 2997-3012(Ch).
- [2] Li X B, Zhuang P J, Yin C. A metadata based manufacturing resource ontology modeling in cloud manufacturing systems [J]. *Journal of Ambient Intelligence and Humanized Computing*, 2019, **10**(3): 1039-1047.
- [3] Yao J, Xing B, Zeng J, *et al.* Overview of cloud manufacturing service portfolio research[J]. *Computer Science*, 2021, **48**(7): 245-255(Ch).
- [4] Bouzary H, Chen F F. Service optimal selection and composition in cloud manufacturing: A comprehensive survey[J]. *The International Journal of Advanced Manufacturing Technology*, 2018, **97**(1): 795-808.
- [5] Zhou J J, Yao X F. A hybrid artificial bee colony algorithm for optimal selection of QoS-based cloud manufacturing service composition[J]. *The International Journal of Advanced Manufacturing Technology*, 2017, **88**(9): 3371-3387.
- [6] Jin H, Lv S P, Yang Z, *et al.* Eagle strategy using uniform mutation and modified whale optimization algorithm for QoS-aware cloud service composition[J]. *Applied Soft Computing*, 2022, **114**: 108053.
- [7] Jiang Y R, Tang L, Liu H L, *et al.* A variable-length encoding genetic algorithm for incremental service composition in uncertain environments for cloud manufacturing[J]. *Applied Soft Computing*, 2022, **123**: 108902.
- [8] Yang Y F, Yang B, Wang S L, *et al.* An improved grey wolf optimizer algorithm for energy-aware service composition in cloud manufacturing[J]. *The International Journal of Advanced Manufacturing Technology*, 2019, **105**(7): 3079-3091.
- [9] Liao S C, Sun P, Liu X C. Service combinatorial optimization based on improved krill swarm algorithm [J]. *Computer Application*, 2021, **41**(12): 3652-3657(Ch).
- [10] Zeng J, Yao J, Gao M, *et al.* A service composition method using improved hybrid teaching learning optimization algorithm in cloud manufacturing[J]. *Journal of Cloud Computing*, 2022, **11**(1): 66.
- [11] Liao W L, Wei L, Wang Y. Manufacturing cloud service composition optimization based on modified polar bear algo-

- rithm[J]. *Computer Application Research*, 2022, **39**(4): 1099-1104(Ch).
- [12] Xue J K, Shen B. A novel swarm intelligence optimization approach: Sparrow search algorithm[J]. *Systems Science & Control Engineering*, 2020, **8**(1): 22-34.
- [13] Tao F, Zhao D M, Hu Y F, *et al.* Correlation-aware resource service composition and optimal-selection in manufacturing grid[J]. *European Journal of Operational Research*, 2010, **201**(1): 129-143.
- [14] Zhou J J, Yao X F. Multi-population parallel self-adaptive differential artificial bee colony algorithm with application in large-scale service composition for cloud manufacturing [J]. *Applied Soft Computing*, 2017, **56**(C): 379-397.
- [15] Liu G Y, Shu C, Liang Z W, *et al.* A modified sparrow search algorithm with application in 3d route planning for UAV[J]. *Sensors*, 2021, **21**(4): 1224.
- [16] Zhu Y L, Yousefi N. Optimal parameter identification of PEMFC stacks using adaptive sparrow search algorithm[J]. *International Journal of Hydrogen Energy*, 2021, **46**(14): 9541-9552.
- [17] Storn R, Price K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces[J]. *Journal of Global Optimization*, 1997, **11**: 341-359.

麻雀搜索算法在云制造服务组合中的多策略改进

周礼亮¹, 李犇², 郁清², 戴桂兰³, 周国富^{4†}

1. 中国电子科技集团公司第十研究所, 四川 成都 610000

2. 南京邮电大学 计算机科学系, 江苏 南京 210023

3. 清华大学 信息技术研究院, 北京 100084

4. 武汉大学 计算机学院, 湖北 武汉 430072

摘要: 在现有研究中, 基于服务质量的云制造服务组合算法优化常常由于在固定空间中进行单一种群搜索和信息交换不足, 导致收敛速度和解质量下降。本文提出了一种改进的麻雀搜索算法 (ISSA) 来解决这些问题。将固定的解空间划分为多个子空间, 并使用并行搜索, 从而加快目标解的发现速度。为了提高这些子空间内的搜索效率并显著改善种群多样性, 采用了多组进化机制和混沌扰动策略。此外, 结合了基于黄金正弦的自适应权重和全局捕捉策略, 更有效地引导个体发现者。最后, 在麻雀位置更新过程中使用差分柯西变异扰动, 以增强算法的全局优化能力。在基准问题和服务组合优化问题上的模拟实验表明, ISSA在优化精度和收敛稳定性方面优于其他方法。结果表明, 本文的方法有效平衡了全局搜索和局部搜索能力, 从而在云制造服务组合中表现出更佳的性能。

关键词: 云制造; 服务组合优化; 服务质量; 麻雀搜索算法

□