

Research Article

An Improved Whale Optimization Algorithm Based on Aggregation Potential Energy for QoS-Driven Web Service Composition

Xuyang Teng, Yuanhao Luo , Tao Zheng, and Xuguang Zhang

School of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, Zhejiang, China

Correspondence should be addressed to Yuanhao Luo; lyh10001@hdu.edu.cn

Received 5 November 2021; Accepted 4 January 2022; Published 7 February 2022

Academic Editor: Yingjie Wang

Copyright © 2022 Xuyang Teng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With more complex user needs, the web service composition (WSC) has become a key research area in the current circumstance. The swarm intelligence algorithms are proved to solve this problem well. However, no researchers have applied the whale optimization algorithm (WOA) to the WSC problem. In this work, we propose a logarithmic energy whale optimization algorithm (LEWOA) based on aggregation potential energy and logarithmic convergence factor to solve this problem. Firstly, the improved algorithm uses a chaotic strategy to enhance the initial swarm diversity. After that, a logarithmic convergence factor is applied to obtain the nonlinear search step. Furthermore, aggregation potential energy as the spatial evaluation is employed in the swarm intelligence algorithms for the first time. Finally, the aggregation potential energy is used to dynamically adjust the nonlinear weight, which improves the search efficiency and prevents the algorithm from falling into local optimization. The experimental results of the benchmark functions show that the LEWOA has better optimization ability and convergence speed than other swarm intelligence algorithms. In the second experiment of the WSC optimization, the effectiveness and superiority of the LEWOA are verified.

1. Introduction

The internet of things (IoT) has become the hot spot of information technology reform. And the research on IoT mainly focuses on privacy protection [1, 2], edge computing and data processing [3], web service composition optimization [4], and so on. In this work, we mainly study the advanced strategies of WOA and the application of the improved whale optimization algorithm in web service composition optimization problems.

Presently under the complex requirements of user needs and scene scheduling, it is far from enough only with a single web service module to provide the solution. Therefore, combining web services to solve the problem has become an inevitable choice for different task requirements that specific workflows can represent. The workflow is a combination of various web services, and each web service has quantity choices of subservices. Obviously, WSC is an NP-hard

problem. In reference [5], Strunk explained the related issues and applications of service composition. Quality of service (QoS) is currently used as the criterion better to evaluate the pros and cons of web services. QoS aims to evaluate web services quantitatively. Although a large number of web services have similar functions, the QoS with different attributes of each web service is different. So the quantity of WSC tends to explode exponentially in the face of specific problems. Furthermore, the time consumption cannot be ignored if the exhaustive algorithm is selected in a large order of magnitude. The swarm intelligence algorithms can be regarded as an outstanding solution to trade off efficiency and effectiveness. In reference [6], Ouarda proposed that the swarm intelligence algorithm effectively solves such large-scale and NP-hard problems.

Here are some typical research and application of various swarm intelligence algorithms in face of WSC problem. In reference [7, 8], the authors proposed IDPSO and

IDIPSO, respectively, aiming to improve the speed and performance of finding the optimal solution of WSC problem. In reference [9], the author proposes an improved ant colony algorithm; EFACO applies a pheromone-driven scheme composed of QoS multiple weights to improve the efficiency of ant colony search. In reference [10], the genetic algorithm (GA) is used to solve the WSC problem based on elitism and an elite-based learning mechanism.

It should be noted that the traditional swarm intelligence algorithm is oriented to a continuous field, but the WSC is a discrete problem. This article will separately integer coding the candidate service set of the workflow and the multidimensional coordinates of the swarm intelligence algorithm, making them correspond to each other. Then, the traditional continuous algorithm has been transformed into a discrete optimization algorithm. Because it only converts the definition of position coordinates without changing the algorithm's optimization principles, the discrete swarm intelligence algorithm can still ensure optimization effectiveness. Nevertheless, the algorithms are more likely to fall into a local optimum in a discrete environment. Aiming to prevent such circumstances and ensure continuous optimization, we must reduce the possibility of search agents gathering. For example, swarm mutation strategy or splitting swarm are to keep the diversity of search agents in the optimization process.

However, not all swarm intelligence algorithms can be applied to WSC. The complexity and the optimization model of an algorithm are the key considerations. WOA is a new swarm intelligence algorithm proposed by Mirjalili and Lewis in reference [11] that is derived from one of the special hunting behaviours of humpback whales called the bubble-net hunting technique [12]. In reference [13], Gharhchopogh made a comprehensive survey of WOA. At present, ant colony system (AG) and particle swarm algorithm (PSO) are widely applied. The optimization principles of WOA are close to PSO in essence. WOA shows its advantages in simpler structure, fewer adjustment parameters, and more excellent global searchability. These essential factors determine WOA can better apply to optimize this problem with minor changes, which ensures the optimization properties of the algorithm will not be significantly affected.

There are still some problems with the WOA algorithm. When facing complex multimodal problems, it shows slow convergence speed and low convergence accuracy and easily falls into the local optimal demerits. Currently, three main ways are able to enhance these defects. First, combining with traditional mathematical principles: in reference [14], Chu incorporates WOA with simulated annealing algorithm to improve the searchability of the algorithm. In references [15, 16], chaotic strategy has been used to initialize the swarm to enhance the diversity of the initial state. Second, combining with the metaheuristic algorithm: in reference [17], Jadhav integrates GWO into WOA and proposes the WGC algorithm. In reference [18], Trivedi leads into the PSO model and presents PSO-WOA to improve the local searchability of WOA. Third, combining with ANN: in

reference [19], WOA is combined with ANN to improve the accuracy of the image segmentation algorithm. In reference [20], ANN is used to find the optimal weight to hasten the convergence speed of WOA and enhance its capacity of jumping out of the local optima, while in reference [21], it increases the recognition accuracy of the algorithm with SVM using WOA to optimize the parameters.

Although there are a large number of improved WOA algorithms, most of them have ignored the influence of the distance relationship between search agents on searchability. In these areas of improving the convergence speed, helping the algorithm jump out of the local optimum, and enhancing the search accuracy, WOA still needs further research. In this work, we first try to improve WOA's optimization ability through three new optimization strategies, focusing on improving the original algorithm based on the aggregation potential of the search agent. In the simulation experiment, this article uses the public test function set to verify the effectiveness of the optimization strategies. Finally, the improved algorithm is transformed into a discrete algorithm DLEWOA through several methods to optimize the WSC problem, and the superiority of the improved algorithm is verified through the QWS public data set.

1.1. Standard WOA Algorithm. In WOA, the search agent will optimize through three strategies: encircling prey, bubble-net attacking, and searching for prey. Moreover, WOA will use the coefficient vector A (in Section 1.2) and the random probability P (in Section 1.3) to control the strategy of the next generation. In encircling strategy, the search agent will select the current optimal individual as the target direction; in search for prey, a random search agent will be selected as the target direction; while in bubble-net attacking, the search agents approach the target along shrinking encircling and spiral updating methods simultaneously.

1.2. Encircling Prey. Before encircling prey, the search agent will first select the current optimal candidate solution as the target value. This value is assumed to be the optimal value or close to the optimal value that will be updated with evolutionary iterations. The remaining search agents will approach the target search agent. Equation (1) allows any search agent to update its position for encircling prey. Equation (2) represents the distance between the remaining search agents and the target. The equations are as follows:

$$X(j+1) = X^*(j) - A \cdot D, \quad (1)$$

$$D = |C \cdot X^*(j) - X(j)|, \quad (2)$$

where j indicates the current iteration, A and C are coefficient vectors, and X^* is the position vector of the best solution obtained so far and should be updated in each iteration if there is a better solution. The equations of A and C are as follows:

$$A = 2a \cdot r_a - a, \quad (3)$$

$$C = 2 \cdot r_c, \quad (4)$$

where r_a and r_c are two random vectors distributed uniformly within $[0,1]$. a is the convergence factor that is defined as follows:

$$a = 2 \left(1 - \frac{j}{T_{\max}} \right). \quad (5)$$

The convergence factor a controls the value of A . Encircling prey and hunting method are chosen when $|A| < 1$, while the searching method is selected when $|A| \geq 1$.

1.3. Bubble-Net Attacking Method. At this phase, the search agent has two simultaneous hunting behaviours: Shrinking encircling and spiraling updating.

Shrinking encircling. Search agents will randomly approach the current best agent

Spiraling updating. Search agents will spirally approach the current best agent

The mathematical model is as follows:

$$D' = |X^*(j) - X(j)|, \quad (6)$$

$$X(j+1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(j), \quad (7)$$

where b is a logarithmic helix shape constant and l is a random number in $[-1, 1]$. Equation (7) represents the distance of the j -th whale to the current best agent. Equation (8) represents the search agent updating position along a spiral-shaped path. Due to a decreased linearly, the position updating range of the search agent will be decreased so as to improve the search accuracy. WOA sets a random number P in $[0, 1]$ to model this simultaneous behaviour. The equation is as follows:

$$X(j+1) = \begin{cases} X^*(j) - A \cdot D, & P < 0.5, \\ X^*(j) + D' \cdot e^{bl} \cdot \cos(2\pi l), & P \geq 0.5. \end{cases} \quad (8)$$

1.4. Search for Prey. This phase emphasizes guaranteeing the WOA algorithm to perform a global search. In this phase, $|A| \geq 1$. A random whale will be chosen from the current group. X_{rand} is the position vector of the random whale. Then, the remaining search agents move towards it, reflecting that the whale will follow the whale group as a whole, which can help the WOA algorithm get rid of the local optimum to a certain extent and perform a global search. The mathematical model is as follows:

$$X(j+1) = X_{rand}(j) - A \cdot D, \quad (9)$$

$$D = |C \cdot X_{rand}(j) - X(j)|. \quad (10)$$

2. Improved LEWOA Algorithm

The original WOA performs well in the low-dimensional unimodal optimization field. However, the nonlinear optimization process of the original WOA does not match with linear decreasing search step size, which will weaken the ability of global optimization in the exploration stage and reduce the convergence accuracy in the exploitation stage. For WOA, although it has the probability to escape through the search for prey strategy after falling into the local optimum, the algorithm itself does not have a specific method of avoiding the local optimum. This paper presents an improved whale optimization algorithm based on logarithmic convergence factor and aggregation potential energy, aiming to optimize the defects from three aspects.

2.1. Chaotic Map. The searchability of the swarm intelligence algorithm is greatly affected by the initial diversity of swarm conditions. In the WOA algorithm, the initial population state is generated randomly, which is unable to guarantee the swarm diversity in the search space. In this sense, a chaotic strategy can solve the problem well for its ergodicity, high randomness, and regularity. Reference [22] shows the effectiveness of combining a chaotic strategy with WOA. In general, the more homogeneous the chaotic sequence is, the more diverse the swarm states will be; hence, we choose the cubic map with better uniformity performance to initialize the swarm. The cubic map expression is as follows:

$$\begin{cases} y(n+1) = 4y(n)^3 - 3y(n), \\ -1 \leq y(n) \leq 1, n = 0, 1, 2, \dots, \end{cases} \quad (11)$$

where y_0 cannot be 0 as the initial value of the iteration; otherwise, the chaotic map cannot be established. The steps through the cubic map to initialize the swarm are as follows:

- (1) First, a d -dimensional vector coordinate $y_{1d} = (y_{11}, y_{12}, y_{13}, \dots, y_{1d})$ is randomly generated, substituting it as the first search agent position into equation (13) and iterating it to obtain $n_1 + n_2$ d -dimensional vectors.
- (2) Due to the limitations of the chaotic model, the coordinate values obtained through iteration are all between $[-1, 1]$. And it is necessary to map the chaotic sequence into the search space through the map function. The function is as follows:

$$x_{i,d} = lb + (1 + y_{id}) \frac{(ub - lb)}{2}, \quad (12)$$

$$i = 1, 2, 3, \dots, dim,$$

where d represents the space dimension of the solution, ub is the upper limit of the d -th dimension of the limited space, while lb is the lower limit one. y_{id} is the premapping coordinate of the i th search agent in the d dimension obtained according to equation (13), and $x_{i,d}$ is the postmapping agent.

- (3) Calculate the fitness value of each coordinate and choose the points with the minimum fitness as the initial position of the search agent.

2.2. The Aggregation Potential Energy. In the process of searching from dispersion to aggregation and finally converging to the optimal point, the state of motion between search agents will affect each other. Hence, their position relationship cannot be simply summarized by the motor pattern of a single search agent. In the field of crowd panic detection [23], the state of the crowd motion can be well illustrated by the increase or decrease of the crowd potential energy [24]. Since there are some similarities in the moving between the swarm and the crowd, this paper introduces aggregation potential energy to describe the aggregation level of the swarm. Aggregation potential energy can well represent the characteristics of individual distribution during the optimization process, so as to better understand the position transformation relationship of search agents in the search space. According to the base definition of potential energy, the larger the aggregation potential energy is, the more scattered the search agents' distribution is. In this state, the algorithm is still in the exploration phase; otherwise, the algorithm is in the exploitation phase or falls into the local optima.

Due to the high sophistication of calculating the distance between each coordinate, this paper defines the aggregation potential energy of swarm as the Euclidean distance between individuals, setting the mean values of coordinates of all search agents in each generation as the population centre. In this way, we replace distance between individuals with Euclidean distance between individuals, and the final population aggregation potential energy is obtained by equation (15). The expression is as follows:

$$E(j) = \phi \frac{\sum_{i=1}^n |c_i - \sum_{i=1}^n c_i/n|}{n}, \quad (13)$$

where ϕ is the search range correction factor and is generally 1 here, c_i is the position of the i -th search agent, and n is the total number of search agents.

By comparing the convergence curve with the change of the aggregation potential energy during experiments, it can be shown that the search agents will aggregate to the optimal point with the increase of the iteration. At the same time, the aggregation potential energy will generally decrease and approach 0. For the problem of judging whether the group is conducting the local search or global search, we cannot confirm to a fixed point. In the optimization process, we can only estimate whether the group's motion strategy tends to global or local search, which is the same for aggregation potential energy. The boundary of aggregation potential energy between global and local is an empirical value obtained through a large number of experiments, generally limited to about $10^{-2} \sim 10^{-4}$. And in the face of different search environments, the optimal boundary value of aggregation potential energy needs to be obtained by testing and adjusting.

With the optimization of the aggregation potential energy, the global searchability of the algorithm can be improved. When the algorithm falls into the local optimum, the search agents converge, and the aggregation potential energy decreases. At this time, the chaotic map disturbance being added to the swarm, the algorithm can get rid of the current state by regenerating half search agents. The steps are as follows:

- (1) Evaluating whether the algorithm falls into local optimum. Different aggregation potential energy thresholds should be set for different problem models to improve the estimation accuracy. If the aggregation potential energy reaches the set threshold value but the current optimal value does not reach the theoretical optimal value, the algorithm will be judged to fall into the local optimum.
- (2) Jumping out of the local optimization through cubic chaotic map. A d -dimensional vector quantity is randomly generated using the cubic chaotic model, iterating it into equation (13) for times. After which particle points are randomly selected and mapped to the search area through equation (14). Then the original half of the search agents are substituted with the worst fitness to enter the iteration again.

The algorithm can be well restarted under the preferable uniformity of the cubic chaotic model by rescattering search agents and replacing them with poor fitness. In the new search process, the search agents that previously fell into the local optimum will reoptimize driven by the new search agents, thus improving the optimization ability of WOA.

2.3. Nonlinear Inertia Weight and Logarithmic Convergence Factor. In WOA, the step sizes of each generation are controlled by the coefficient A . The convergence factor a controls the change of A . However, as a linear diminishing factor, a leads to the mismatch between the linear diminish of the search step size and the nonlinear convergence of the algorithm. Therefore, this paper converts a in logarithmic convergence form, which improves the optimization ability of the algorithm while guaranteeing that the convergence factor a decreases. The optimization process can generally be divided into two stages: global optimization and local optimization. The former generally lasts a short time, and the latter lasts a long time. Therefore, the linearly reduced convergence factor cannot match the convergence characteristics of this algorithm. The convergence factor obtained by logarithmic type can quickly reduce the convergence step in the early stage and change the search step in the later stage, so it is appropriate for a to be advanced in this new type. The new expression of a is as follows:

$$a = 2 - 2 \cdot \ln\left((e - 1) \frac{j}{T_{\max}} + 1\right). \quad (14)$$

Swarm intelligence algorithm requires a large step size in the exploration phase. While in the exploitation phase, it requires a small one. In the overall search process, it is

adverse for algorithm optimization if the linear variation of step size in the search process does not conform to the actual nonlinear search process. Reference [25] corroborates the effectiveness of the nonlinear adaptive strategy to improve the optimization ability of swarmed-based algorithms. Hence, this paper introduces nonlinear adaptive inertia weight to improve the step size, particularly, combining the aggregation potential energy based on the mathematical model proposed in reference [25]. In the light of the aggregation potential energy, the search state of each search agent can be judged, controlling the algorithm to distribute different inertia weight strategies, aiming to enhance the rate of convergence and the accuracy of convergence. The improved nonlinear inertia weight equation is as follows:

$$w(t)_i = \begin{cases} \gamma \cdot \left(w_1 - \frac{(w_2 - w_1)}{T_{\max}} \cdot \frac{f(j)_i - f(j)_{\min}}{E(j) \cdot (f(j)_{\max} - f(j)_{\min})} \right), & f(j)_i < f(j)_{\text{avg}}, E \geq 1 \\ \gamma \cdot \left(w_2 + \frac{(w_2 - w_1)}{T_{\max}} \cdot \frac{f(j)_i - f(j)_{\text{avg}}}{E(j) \cdot (f(j)_{\max} - f(j)_{\min})} \right), & f(j)_i \geq f(j)_{\text{avg}}, E \geq 1 \\ \gamma \cdot \left(w_1 - \frac{(w_2 - w_1)}{T_{\max}} \cdot \frac{f(j)_i \cdot E(j)}{f(j)_{\max} - f(j)_{\min}} \right), & E < 1 \end{cases}, \quad (15)$$

where $w(j)_i$ represents the inertia weight value of the i -th search agent of the j -th generation; w_1 and w_2 are the initial minimum and maximum inertia weight values, respectively; and T_{\max} is the maximum number of iterations. $E(j)$ is the j -th value of the aggregation potential energy. $f(j)_i$ is the i -th fitness of the search agents of the j -th generation, and $f(j)_{\text{avg}}$ is the average fitness value of all the search agents of the j -th generation. As for $f(j)_{\max}$ and $f(j)_{\min}$, they are the maximum fitness value and the minimum one of the swarm of the j -th generation. γ is the search range correction factor, which is inversely proportional to ϕ , and their product is 1.

Equation (17) is the inertia weight equation controlled by the aggregation potential energy E . Based on the experiment analysis, we find that when $E \geq 1$, the aggregation potential energy can be considered large with the relatively scattered distribution of search agents. The algorithm is basically in the exploration phase. In this case, if the fitness value of an individual is less than the average fitness value, the search agent is close to the current optimal point, and it should be assigned a smaller inertial weight to approach the optimal point; on the contrary, search agents with poor fitness should be assigned larger inertia weights to enlarge the search step size that is able to improve the global search capability. When $E < 1$, the aggregation potential energy is small, and it is appropriate to search locally with the concentrated

distribution of search agents. Each search agent can be distributed with a smaller inertia weight under the protection of jumping-from-local optimal strategy to optimize with high precision around the optimal point, thus ensuring convergence accuracy. The equation of the update of the new position of the search agent is as follows:

$$X(j+1) = \begin{cases} w(t) \cdot X^*(j) - A \cdot D, & |A| < 1, P < 0.5 \\ w(t) \cdot X_{\text{rand}}(j) - A \cdot D, & |A| \geq 1, P < 0.5 \\ w(t) \cdot X^*(j) + D' \cdot e^{bl} \cdot \cos(2\pi l), & |A| < 1, P \geq 0.5 \end{cases} \quad (16)$$

2.4. The Flow of the LEWOA. The flowchart of LEWOA is shown in Figure 1, and the detailed steps are described as follows:

Step 1: initialize parameters n , n_1 , n_2 , d , and T_{\max} , set iteration initial value j as 1, and identify optimization targets and search areas.

Step 2: use equation (13) to generate $n_1 + n_2$ d -dimensional vectors, map them into the search area by equation (14), calculate their fitness values, and select the n points with the worst fitness values as the initialization search agents.

Step 3: calculate the fitness value of each search agent to update the current optimal fitness search agent as X^* .

Step 4: update parameters a , l , P , A , C , E , and w .

Step 5: estimate whether the algorithm is trapped in a local optimum by aggregation potential energy E and the current optimal fitness value. If local optimal is entered, regenerate $n/2$ points through step 2, replacing the $n/2$ search agents with the worst current fitness values, and if not, skip this step.

Step 6: update the positions of search agents through equation (18).

Step 7: determine whether the current generation has reached the maximum number of iterations. If not, $j = j + 1$ and turn to step 3. If reached, exit the loop and output the X^* .

3. Web Service Composition Problem Modeling

3.1. QoS-Driven Web Service Composition. WSC problem generally evaluates service capability through QoS. This work divides QoS attributes into positive attributes and negative attributes, including response time, availability, success ability, reliability, service price, throughput rate, credibility, and so on. The establishment of this model selects the first four kinds of attributes for research. In addition, in order to standardize the attribute value of each QoS, the normalization function of QoS is established as follows:

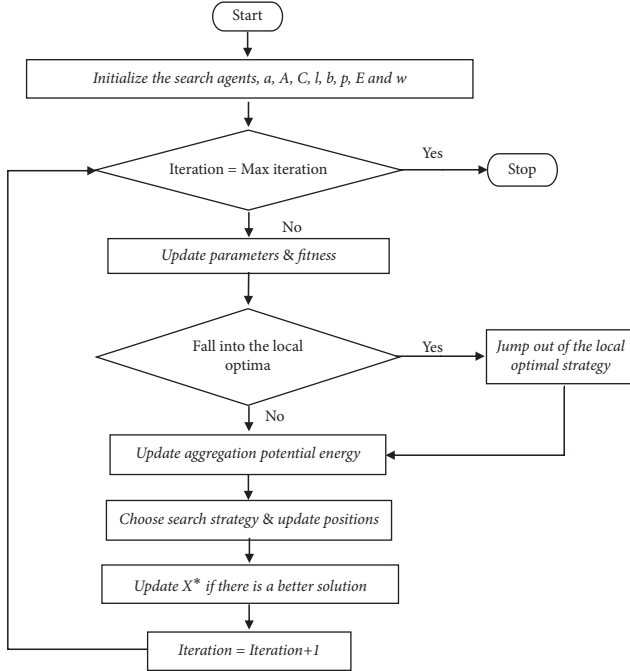


FIGURE 1: The flowchart shows LEWOA algorithm search strategies.

$$QoS'_{i,j}(S) = \begin{cases} \frac{QoS_{i,j}(S) - QoS_{j_min}}{QoS_{j_max} - QoS_{j_min}}, & \text{Positive - factor,} \\ \frac{QoS_{j_max} - QoS_{i,j}(S)}{QoS_{j_max} - QoS_{j_min}}, & \text{Negative - factor,} \end{cases} \quad (17)$$

where i is the service number, j is the QoS attribute of the j -th service, $QoS_{i,j}(S)$ is the value of the i -th service and the j -th attribute, and QoS_{j_max} and QoS_{j_min} are the maximum and minimum values of the j -th QoS attribute, respectively. $QoS'_{i,j}(S)$ is the normalized value.

WSC is often carried out by the following several common workflow patterns: series, concurrent, select, and circulation, as shown in Table 1. Further to say, all of these four patterns can be transformed into series types. Hence, in this work, we choose a series WSC workflow as an instance to complete the experiment, as shown in Figure 2.

3.2. Problem Modeling. The basic idea of applying the LEWOA algorithm to the WSC problem is as Figure 3 shows: in the WSC problem, m specific candidate services are selected from each abstract service class to form a service composition. Therefore, the coordinate dimension of the search agents can be set to m to map the corresponding abstract service class. The coordinate value range of a dimension is mapped to n candidate services available for selection in the corresponding abstract service class. Figure 3 shows the specific mapping process. It should be noted that in practical problems, the number of candidate services in each service class might be different. Therefore, a specific

TABLE 1: Definition of the aggregate function.

QoS	Series	Concurrent	Select	Circulation
Response time	$\sum_i^n T_i$	$\max T_i$	$\sum_i^n p_i * T_i$	$m * \sum_i^n T_i$
Availability	$\prod_i^n A_i$	$\min A_i$	$\prod_i^n p_i * A_i$	$\prod_i^n A_i$
Success ability	$\prod_i^n S_i$	$\min S_i$	$\prod_i^n p_i * S_i$	$\prod_i^n S_i$
Reliability	$\prod_i^n R_i$	$\prod_i^n R_i$	$\prod_i^n p_i * R_i$	$\prod_i^n R_i$

Note: Q_i is the QoS of the i -th candidate service; n is the number of tasks; and m is the number of cycles.

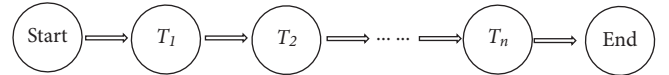


FIGURE 2: The model of the series structure. Every T_n means one-part specific task of the whole WSC workflow.

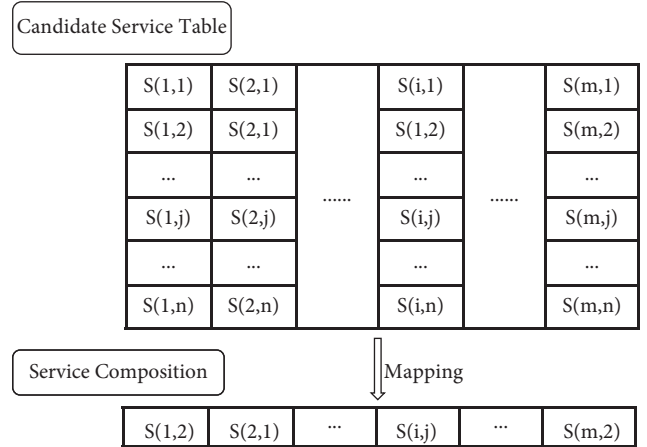


FIGURE 3: Candidate services table with m columns and n rows, and each grid represents a specific service. The service composition is obtained through the combination of mapping.

search agent represents a specific combination of web services, and its QoS can be calculated through the fitness function. Based on the value of QoS, we can use LEWOA to optimize WSC, and then the appropriate service composition can be obtained.

According to the above selected parameters and workflow patterns, the fitness function can be converted as follows:

$$f(x) = F(\min T(s), \min A(s), \min S(s), \min R(s)), \quad (18)$$

where $T(s)$ is the response time, $A(s)$ is availability, $S(s)$ is success ability, and $R(s)$ is reliability.

3.3. Feasibility Analysis of Improved Strategy. After the QoS-driven WSC problem is converted into a single objective optimization problem, its essence has been transformed into efficiently selecting the optimal composition from a large

number of permutations. For most swarm intelligence algorithms, the search logic can be regarded as taking randomly from the coding candidate set and feeding back according to the obtained value to optimize the next selected generation, which is a logical and efficient random extraction. According to this extraction principle, we can find two ways to improve the optimization: on the one hand, having as many extraction times as possible in limited generations and, on the other hand, ensuring that the extracted value can better optimize the extraction strategy of the next generation.

Based on the first optimization principle, considering the time and complexity of the WSC problem, it is hard for the swarm intelligence algorithm to guarantee to search the optimal value of the web service composition. Therefore, only when the algorithm keeps searching within limited generations will it not cause search waste. First, we need to avoid the algorithm falling into the local optimum. Second, the algorithm needs to ensure the diversity of the search population, so as to obtain the most information in each generation of extraction. For the improved algorithm, first, the cubic chaotic mapping principle is adopted to ensure the diversity of initial search agents; second, the aggregation situation is estimated by the aggregation potential energy of the search agent. As long as the algorithm enters the local search stage, the split mutation strategy is adopted to regenerate a new part of the population to ensure the swarm diversity of the next generation.

The second optimization principle needs a swarm intelligence algorithm to efficiently feedback the search information obtained by the previous generation and choose the suitable search strategy. For example, in applying the ant colony algorithm to web service composition, pheromones are efficiently used to transmit the search information of the previous generation. However, in the original WOA algorithm, no such parameter can better feedback information to the next generation. In the face of the solution of most continuity problems, WOA's search logic will not have a great defect. However, it is difficult to have good optimization performance when encountering complex high-dimensional problem models and discrete problems. The new parameter of aggregation potential energy in LEWOA proposed in this paper can be better used as a medium to transmit the search information between each generation. In the aspect of search strategy, the search strategy of the next generation is determined according to the aggregation potential energy of each generation. In an aspect of step size, the improved algorithm changes the linear step size and convergence factor to nonlinear variation, which is to match the actual nonlinear search process.

3.4. Encoding Rules for Mapping. Since the coordinate values of each dimension of the search agent in the LEWOA are continuous, the serial number of the specific candidate service is discrete. Therefore, the fuzzy function f_d should be used to convert the coordinate values of the search agent into corresponding integer code to form the discrete logarithmic energy whole optimization algorithm (DLEWOA). The fuzzy function f_d is as follows:

$$f_d(x_{i,d}^t) = \begin{cases} z, (z - 0.5) < x_{i,d}^t < (z + 0.5) \\ \text{iff}(Y = 0, z, z + 1), x_{i,d}^t = (z + 0.5) \\ m_d, x_{i,d}^t \in [0, 0.5] \cup (m_d - 0.5, m_d] \\ \text{iff}(Y = 0, m_d, 1), x_{i,d}^t = 0.5 \end{cases}, \quad (19)$$

where $x_{i,d}^t$ is the coordinate value of the t generation of the search agent i in the d dimension, m_d is the number of candidate services of the abstract service class corresponding to this dimension, and z is the integer on $[1, m_d]$. The random variable Y is the result of a Bernoulli test with a probability of 0.5. The value of the function $\text{iff}(P, u, v)$ depends on whether the proposition P is true. If true, it is u ; else, it is v .

4. Simulation Experiment and Analysis

4.1. Parameters Setting and Benchmark Functions. To evaluate the optimization ability of the LEWOA, several simulation experiments are conducted based on eight benchmark functions (shown in Table 2), whose experiment results are compared to other swarm intelligence algorithms. The experiment parameters are set as follows: the number of the search agents is 30, the average experimental simulation is 30 times, and the maximum number of iterations is 1,500. As for MWOA, the inertia weight is set according to reference [25], whose minimum value w_1 equals 0.01 and the maximum one w_2 equals 0.4. For the reason that the original fitness parameters are weighted by the aggregation potential energy of the population, after a multitude of experiments, the parameter of the inertia weight is $w_1 = 0.01$, and the maximum weight parameter is $w_2 = 0.08$. The simulation experiment in this section is conducted in the conditions of Intel Core, CPU i7-7700HQ, 2.80 GHz, 8 GB, and MATLAB 2016a. Table 3 manifests the specific algorithmic parameters.

Table 2 is the model of eight benchmark functions. F1, F2, F3, and F4 belong to unimodal function mainly utilized to test the convergence rate and accuracy of the algorithm; F5 and F6 are multimodal functions mainly wielded to test the ability of global search of the algorithm; and F7 and F8 belong to the multimodal function of mixed dimensions, which are applied to test the impact of search dimension on the searchability of the algorithm.

4.2. Experimental Results and Analysis. In this section, a comprehensive analysis of the LEWOA will be presented. The experiment records the fitness values of the best search agent and calculates the mean value and the standard deviation to evaluate the accuracy and stability of algorithm optimization results. It analyses the number of convergence and depicts the variation curves of the fitness values of the best search agent and convergence generations to evaluate the algorithm's convergence rate. The results are shown in Table 4, in which the black-labeled represents the optimal data.

TABLE 2: Eight classical benchmark test functions.

Function	Dimension	Range	Optimal value
$F_1(x) = \sum_{i=1}^n x_i^2$	30	[-100,100]	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10,10]	0
$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	[-100,100]	0
$F_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30	[-100,100]	0
$F_5(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1]$	30	[-1.28,1.28]	0
$F_6(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
$F_7(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6})^{-1}$	2	[-65,65]	1
$F_8(x) = \sum_{i=1}^2 [a_i - \frac{x_i (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	4	[-5,5]	3.00×10^{-4}

TABLE 3: Algorithm parameter setting.

Algorithm	Parameter	Value
MFO [26]	Convergence factor [a]	Linearly decreases from -1 to -2
ABC [27]	Step size [a]	1
	Fraction [P_a]	0.25
	Acceleration [C_1, C_2]	2
PSO [28]	Inertia weight [w]	0.8
	V_{\max}	0.05
	Acceleration [A]	20
GSA [29]	Power	1
	Gravitational [G_0]	100
WOA [11]	Convergence factor [a]	Linearly decreases from 2 to 0
MWOA [25]	Convergence factor [a]	Nonlinearly decreases from 2 to 0
LEWOA	Convergence factor [a]	Nonlinearly decreases from 2 to 0

TABLE 4: Performance comparisons of three algorithms on eight benchmark functions.

Function	Evaluation	WOA	MWOA	LEWOA
F1	Ave	$3.60 e - 277$	0	0
	Std	0	0	0
	Generations	1,500	796	620
F2	Ave	$9.36 e - 156$	0	0
	Std	$4.45 e - 155$	0	0
	Generations	1,500	902	740
F3	Ave	11022.94	0	0
	Std	8145.24	0	0
	Generations	1,500	807	604
F4	Ave	36.16	0	0
	Std	29.77	0	0
	Generations	1,500	887	739
F5	Ave	$7.04 e - 4$	$1.94 e - 5$	$3.66 e - 5$
	Std	$8.22 e - 4$	$6.22 e - 5$	$4.04 e - 5$
	Generations	1,500	1,500	1,500
F6	Ave	0	0	0
	Std	0	0	0
	Generations	422	37	19
F7	Ave	1.72	1.69	1.65
	Std	2.95	0.933	0.819
	Generations	1,500	1,500	1,500
F8	Ave	$5.52 e - 4$	$3.59 e - 4$	$3.53 e - 4$
	Std	$2.28 e - 4$	$4.99 e - 5$	$5.56 e - 5$
	Generations	1,500	1,500	1,500

According to the test results from F1 to F6 in Table 4, LEWOA has surpassed WOA from the perspectives of each comparison. This result shows that LEWOA is obviously superior to WOA in the accuracy, stability, and rate of optimization based on the six benchmark functions. From F1 to F4 and in F6, LEWOA and MWOA all reach the target values, and the former convergence rate is better since the convergence number of iterations of LEWOA is evidently less than those of MWOA, which is more apparent in the unimodal function test environment. In terms of multimodal functions from F5 to F8, except for F8 in which the standard deviation of LEWOA is slightly inferior to those of MWOA, there are no disparities on account of the order of magnitude. For other benchmark functions, the accuracy and stability of LEWOA's convergence are better than those of MWOA, and the convergence rate is also preponderant. In the case of F8, considering that the increase of the search dimensions has the corresponding influence on the effect of optimization, it can be concluded that the test results in the mean value and standard deviation of the optimal fitness of LEWOA are much better than those of WOA, and it also has better accuracy and stability of convergence for MWOA.

To more intuitively reflect the algorithms' convergence in the benchmark function, a convergence curve is drawn due to one experiment that is the closest to the 30 times of the average test results, comparing the characteristics of convergence of WOA, MWOA, and LEWOA. Figure 4 relatively corresponds to functions F1 to F8, in which the bold font is the best result.

From the convergence curve of Figure 4, LEWOA has a qualitative improvement over WOA in search efficiency and local search accuracy. Especially in F3 and F4, it can be seen that compared with the improved algorithms, WOA proves to fall into the local optimum easily when processing complex problems, while the improved algorithms using the nonlinear strategy still have better convergence. For F1, F2, and F6, the WOA appears very weak in the later local search phase due to its linear allocation step size. On the contrary, the convergence rate of the improved algorithms does not decrease at the later stage because the later algorithms can still allocate an appropriate search step. Compared with the MWOA algorithm, LEWOA is better in convergence generations while maintaining astringency because, under the control of aggregation potential energy, LEWOA can more efficiently allocate optimal search strategies for search agents at each generation.

In theory, the optimization ability of the swarm intelligence algorithm increases exponentially with the increase of the swarm population. Therefore, in order to demonstrate the optimization ability of LEWOA, Table 5 discusses the influence of the ability of optimization of LEWOA on the population size, in which the black-labeled represents the optimal data. The swarm number in Table 5 takes into account the nonlinear growth optimization ability of the swarm intelligence algorithm. From unimodal test functions $f_1 \sim f_4$, we can find that the change of population has little impact on the excellent optimization ability of LEWOA. In general, small populations have more efficient optimization speed in such a simple search environment. In the

multimodal function of $f_4 \sim f_8$, although the large population has relatively good optimization results, careful observation of the data shows that the small population's optimal value and standard deviation have only a decline of one order of magnitude at most. The LEWOA algorithm for the small population, even in a complex environment still maintains good convergence accuracy and stability. Compared with PSO, GSA, ABC, and other algorithms that need a large number of populations to maintain their astringency, the LEWOA algorithm still occupies a dominant position in the field of small population search.

Table 6 compares data from LEWOA and the current swarm intelligence algorithms, selecting MFO, PSO, ABC, GSA, and WOA for performance comparison. To ensure the objectivity and accuracy of the experimental data, each algorithm runs 30 times independently, and the test functions are $f_1 \sim f_8$. The average value of optimal solutions of six algorithms and the standard deviation of 30 times of independent operation are shown in Table 5, in which the bold font is the best result.

Concluded from the test data of 8 benchmark functions $f_1 \sim f_8$ in Table 6, the optimal data of LEWOA are not acquired merely in f_7 , which embodies the superiority of optimization of the improved algorithm. For the high-dimensional unimodal function $f_1 \sim f_4$, LEWOA is able to converge to the optimal value every time under the 1,500 generations, which other comparison algorithms cannot achieve. The excellent optimization results not only show that LEWOA has higher precision local convergence ability in unimodal problems but also has better algorithm stability. The improvement of the local convergence ability of LEWOA is due to the improved inertia weight and nonlinear convergence factor, which enables the algorithm to allocate appropriate search steps to deal with the complex search environment. In higher dimensional multimodal functions $f_5 \sim f_6$, LEWOA excels better in the accuracy of optimization and stability than other current algorithms other than WOA, which indicates WOA itself possesses a good global optimization ability of global optimization. According to the results of f_5 , LEWOA is better in the accuracy and stability of convergence than those of WOA. It proves the progress of the ability of global optimization of the improved algorithm in higher dimensional multimodal problems and the effectiveness of the combination of aggregation potential energy with nonlinear inertia weight strategy to improve the ability of optimization. For mixed low dimensional multimodal functions f_7 and f_8 , the optimal solution and standard deviation of LEWOA are only slightly smaller than that of the ABC algorithm in f_7 . However, from f_7 to f_8 , with the increase of function dimension, it exactly turns out LEWOA performs better than the ABC algorithm both in convergence accuracy and stability. This change shows that with the help of aggregation potential energy, LEWOA can better jump out of local optimal and conduct global optimization, which proves its astringency and dominance in complex problems.

4.3. Experimental Simulation of Web Service. In this section, the LEWOA is applied to the QoS-driven WSC problem and compared the optimization performance in the QWS data

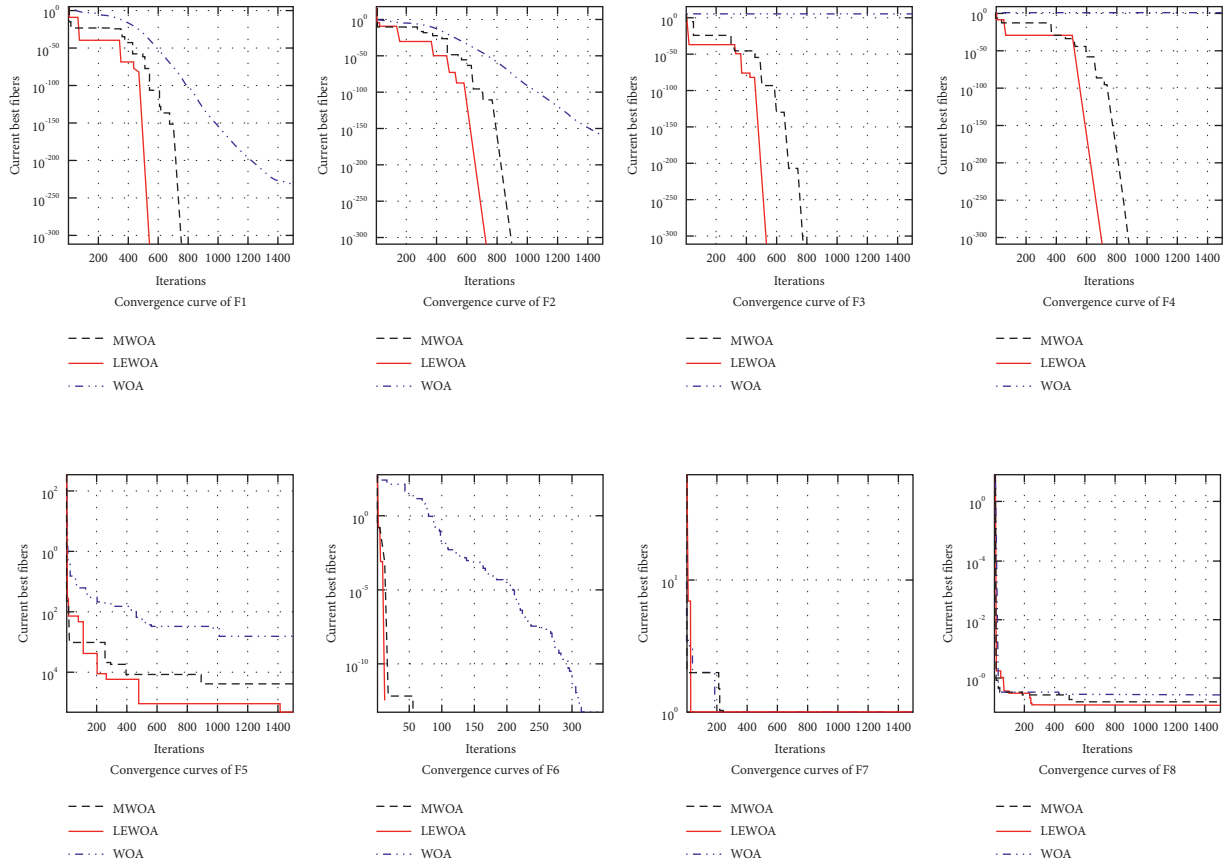


FIGURE 4: Convergence curves of 8 benchmark functions. The plot of current best fitness with respect to iteration recorded for 1,500 iterations. Display 3 different algorithms convergence curves.

TABLE 5: Performance comparisons of LEWOA on eight benchmark functions different population size.

Function	Evaluation	The number of search agents			
		10	20	25	30
F1	Ave	0	0	0	0
	Std	0	0	0	0
	Generations	356	345	348	352
F2	Ave	0	0	0	0
	Std	0	0	0	0
	Generations	465	463	456	459
F3	Ave	0	0	0	0
	Std	0	0	0	0
	Generations	340	359	355	348
F4	Ave	0	0	0	0
	Std	0	0	0	0
	Generations	467	451	459	461
F5	Ave	$1.14 e-4$	$3.98 e-5$	$3.75 e-5$	$3.59 e-5$
	Std	$1.16 e-4$	$4.82 e-5$	$3.57 e-5$	$3.19 e-5$
	Generations	1,500	1,500	1,500	1,500
F6	Ave	0	0	0	0
	Std	0	0	0	0
	Generations	18	16	16	14
F7	Ave	3.51	2.57	1.35	1.34
	Std	3.25	2.93	0.74	0.67
	Generations	1,500	1,500	1,500	1,500
F8	Ave	$4.06 e-4$	$3.56 e-4$	$3.58 e-4$	$3.46 e-4$
	Std	$1.14 e-4$	$4.18 e-5$	$4.93 e-5$	$3.17 e-5$
	Generations	1,500	1,500	1,500	1,500

TABLE 6: Performance comparisons of six algorithms on eight benchmark functions.

Function	Evaluation	LEWOA	WOA	MFO	PSO	ABC	GSA
F1	Ave	0	$3.1739 e - 230$	$3.3126 e - 47$	0.0505	0.0010	$7.593 e - 17$
	Std	0	0	$1.0324 e - 46$	0.0204	$9.6424 e - 4$	$3.0781 e - 17$
F2	Ave	0	$2.4810 e - 156$	1.6667	0.5709	7.0518	$4.4585 e - 8$
	Std	0	$1.3409 e - 155$	4.6113	0.7609	18.9981	$9.0015 e - 9$
F3	Ave	0	$1.928 e + 4$	500.000	9.0176	$5.9959 e + 4$	258.5095
	Std	0	$6.9793 e + 3$	$1.52556 e + 3$	3.4258	$8.8142 e + 3$	127.6924
F4	Ave	0	28.7110	0.8791	2.1892	59.2963	0.0102
	Std	0	31.5702	1.7280	0.8402	4.8688	0.0504
F5	Ave	$3.1616 e - 15$	$9.7329 e - 4$	0.0063	0.0133	0.4046	0.0453
	Std	$3.2410 e - 5$	0.0010	0.0053	0.0054	0.1437	0.0138
F6	Ave	0	0	24.1561	27.4233	229.7770	26.1342
	Std	0	0	16.0788	10.5341	15.4841	15.9193
F7	Ave	1.5272	1.7813	2.7264	2.7730	0.9980	2.8253
	Std	0.8536	2.4926	3.5507	2.4099	$2.7390 e - 5$	1.9226
F8	Ave	$3.5318 e - 4$	$5.2018 e - 4$	0.0022	$4.2767 e - 4$	0.0012	0.0024
	Std	$5.5550 e - 5$	$2.7276 e - 4$	0.0050	$3.586 e - 4$	$1.5012 e - 4$	0.0014

TABLE 7: Performance comparisons of three algorithms on QWS.

Algorithm	Ave	Worse	Best	Std
DLEWOA	1.4664	1.5121	1.4034	0.0293
DWOA	1.6223	1.7296	1.5447	0.0492
DMWOA	1.6311	1.7080	1.5602	0.0471
DPSO	1.6530	1.7085	1.4295	0.0633
DMFO	1.7607	1.8020	1.6120	0.0579

TABLE 8: Difference level analysis relative to DLEWOA.

Algorithm	P -value	α	Confidence interval
DLEWOA	—	0.05	[0, 0]
DWOA	$2.4457 e - 17$	0.05	[0.1353, 0.1560]
DMWOA	$5.5559 e - 20$	0.05	[0.1657, 0.1836]
DPSO	$1.5275 e - 14$	0.05	[0.1687, 0.2054]
DMFO	$1.6085 e - 19$	0.05	[0.2784, 0.3103]

set with the DWOA algorithm and DPSO. The experiment sets 10 service classes, each service class contains 250 kinds of services and constructs a web service composition model in series. Four QoS evaluation criteria in QWS are selected, which are response time, availability, success ability, and reliability. They are brought into equation (17) for normalization. In addition to availability, the other three are treated as positive factors, and the weights of the four are set as 0.2, 0.3, 0.2, and 0.3.

The parameter settings of the following comparison algorithms mentioned in Table 7 are the same as those in Table 3. In order to adapt to the discrete search process of WSC problem, the coordinates of each algorithm are integer processed by fuzzy function equation (19).

Since, in the WSC problem of this work, the search scope is expanded to two orders of magnitudes, in order to ensure the search efficiency, ϕ and γ will be changed to 0.01 and 100, respectively. Besides, the optimization ability of the WSC problem is guaranteed by the global search as mentioned in

Section 3.3. Therefore, in order to ensure the diversity of the population and enhance the global searchability of the DLEWOA, the aggregation potential energy threshold is set to 1, and 20 independent experiments are conducted under the same conditions. The other experimental parameters are set the same as before. The experimental results are as follows, among which bold font is the best:

To further illustrate DLEWOA, we use the P -value obtained by t -test to test whether the result values of each algorithm belong to the same distribution to prove the uniqueness of the LEWOA algorithm. P -value is a probability of observed samples and more extreme cases on the premise that the original hypothesis is true. We initially assumed that the values obtained by other algorithms belong to the same distribution as those obtained by LEWOA. Therefore, the smaller the P -value obtained, the more rejected the original hypothesis. Generally speaking, when the value of P -value is less than 0.001, it can be considered that there is a significant difference. In Table 8, where α is the

significance level and set to 0.05 and then the confidence level is 95%. In this article, we use the t -test function from MATLAB to obtain P value and confidence interval.

It can be seen from Table 7 that compared with the other algorithms, the average optimization result and the optimal value of the DLEWOA are much better, and it also has the lowest standard deviation, which shows that the DLEWOA algorithm in the WSC problem guarantees both validity and stability. Obviously, based on the data of Table 8, no matter which algorithm DLEWOA is compared with, the order of magnitude of the p -value obtained is small enough to verify its uniqueness.

At the same time, this paper tries to conduct a comparative experiment on experimental data through a pseudorandom traversal algorithm in the same experimental environment. When the optimization result reaches about 1.50, the pseudorandom algorithm needs to iterate around 9,000 generations, which takes several times as the swarm intelligence algorithm. Although the pseudorandom traversal algorithm can get better optimization results with enough iterations, the swarm intelligence algorithm is obviously a better choice when time cost is considered.

5. Conclusions

This paper mainly advances WOA into LEWOA combining three strategies aimed at its defects in processing some multimodal functions and studies the application of DLEWOA in the QoS-driven WSC problem. The DLEWOA is proposed by using integer coding with the fuzzy function, which solves the problems of mismatch between continuity algorithm and discrete problem model. In the first analysis of LEWOA with eight test functions, the improved algorithm demonstrates its strengths in the convergence rate, optimization ability, and convergence accuracy. In the meantime, this paper tests the impact of swarm's quantity on algorithms, showing that the improved algorithm can still ensure a higher convergence rate and search accuracy in the small population. In the second experiment of QoS-driven WSC, this paper tests the DLEWOA through the QWS data set, and the experiment proves the superiority of the improved algorithm in the comprehensive performance of the WSC optimization problem. As a result, the above experiments validate the effectiveness and superiority of the improved algorithm: LEWOA.

Data Availability

The optimization functions used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant nos. 61906055, 61771418,

61872105, and 62072136 and the National Key R&D Program of China under Grant no. 2020YFB1710200.

References

- [1] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–599, 2018.
- [2] T. Liu, Y. Wang, Y. Li, X. Tong, L. Qi, and N. Jiang, "Privacy protection based on stream cipher for spatiotemporal data in IoT," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7928–7940, 2020.
- [3] A. Alnoman, S. K. Sharma, W. Ejaz, and A. Anpalagan, "Emerging edge computing technologies for distributed IoT systems," *IEEE Network*, vol. 33, no. 6, pp. 140–147, 2019.
- [4] Y. Song and Y. Z. Gong, "Web service composition on IoT reliability test based on cross entropy," *Computational Intelligence*, vol. 36, no. 4, pp. 1650–1662, 2020.
- [5] A. Strunk, "QoS-aware service composition: a survey," in *Proceedings of the Eighth IEEE European Conference on Web Services IEEE*, pp. 67–74, Ayia Napa, Cyprus, December 2010.
- [6] O. Zedadra, A. Guerrieri, N. Jouandea, G. Spezzano, H. Seridi, and G. Fortino, "Swarm intelligence-based algorithms within IoT-based systems: a review," *Journal of Parallel and Distributed Computing*, vol. 122, pp. 173–187, 2018.
- [7] L. Huang, X. Zhang, Y. Huang, G. Wang, and R. Wang, "A QoS optimization for intelligent and dynamic web service composition based on improved PSO Algorithm," in *Proceedings of the 2011 Second International Conference on Networking and Distributed Computing*, pp. 214–217, Beijing, China, September 2011.
- [8] X. Zhao, B. Song, P. Huang, Z. Wen, J. Weng, and Y. Fan, "An improved discrete immune optimization algorithm based on PSO for QoS-driven web service composition," *Applied Soft Computing*, vol. 12, no. 8, pp. 2208–2216, 2012.
- [9] F. Dahan, K. E. Hindi, A. Ghoneim, and H. Alsalman, "An enhanced ant colony optimization based algorithm to solve QoS-aware web service composition," *IEEE Access*, vol. 9, pp. 34098–34111, 2021.
- [10] M. Chen, Q. Wang, W. Sun, X. Song, and N. Chu, "GA for QoS satisfaction degree optimal web service composition selection model," in *Proceedings of the 2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC)*, Beijing, China, October 2019.
- [11] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [12] W. A. Watkins and W. E. Schevill, "Aerial observation of feeding behavior in four baleen whales: *Eubalaena glacialis*, *balaenoptera borealis*, *megaptera novaeangliae*, and *balaenoptera physalus*," *Journal of Mammalogy*, vol. 60, no. 1, pp. 155–163, 1978.
- [13] F. S. Gharehchopogh and H. Gholizadeh, "A comprehensive survey: whale optimization Algorithm and its applications," *Swarm and Evolutionary Computation*, vol. 48, pp. 1–24, 2019.
- [14] D. Chu, H. Chen, and X. Wang, "Whale optimization algorithm based on adaptive weighting and simulated annealing," *Acta Electronica Sinica*, vol. 5, no. 5, pp. 992–999, 2019.
- [15] K. Gaganpreet and A. Sankalap, "Chaotic whale optimization algorithm," *Journal of Computational Design and Engineering*, vol. 5, pp. 275–284, 2018.
- [16] D. Oliva, M. Abd El Aziz, and A. Ella Hassanien, "Parameter estimation of photovoltaic cells using an improved chaotic

- whale optimization algorithm,” *Applied Energy*, vol. 200, pp. 141–154, 2017.
- [17] A. N. Jadhav and N. Gomathi, “WGC: hybridization of exponential grey wolf optimizer with whale optimization for data clustering,” *Alexandria Engineering Journal*, vol. 57, no. 3, pp. 1569–1584, 2018.
- [18] I. N. Trivedi, P. Jangir, A. Kumar, N. Jangir, and R. Totlani, “A novel hybrid PSO-WOA algorithm for global numerical functions optimization,” *Advances in Computer and Computational Sciences*, vol. 554, pp. 53–60, 2018.
- [19] Virupakshappa and B. Amarapur, “Computer-aided diagnosis applied to MRI images of brain tumor using cognition based modified level set and optimized ANN classifier,” *Multimedia Tools and Applications*, vol. 79, no. 5-6, pp. 3571–3599, 2018.
- [20] I. Aljarah, H. Faris, and S. Mirjalili, “Optimizing connection weights in neural networks using the whale optimization algorithm,” *Soft Computing*, vol. 22, no. 1, pp. 1–15, 2018.
- [21] A. M. Al-Zoubi, H. Faris, J. f. Alqatawna, and M. A. Hassonah, “Evolving Support Vector Machines using Whale Optimization Algorithm for spam profiles detection on online social networks in different lingual contexts,” *Knowledge-Based Systems*, vol. 153, pp. 91–104, 2018.
- [22] Z. Xu, Y. Yu, H. Yachi, J. Ji, Y. Todo, and S. Gao, “A novel memetic whale optimization algorithm for optimization,” *Advances in Swarm Intelligence. Lecture Notes in Computer Science*, vol. 10941, pp. 384–396, 2018.
- [23] X. Zhang, Q. Yu, and H. Yu, “Physics inspired methods for crowd video surveillance and analysis: a survey,” *IEEE Access*, vol. 6, pp. 66816–66830, 2018.
- [24] G. Xiong, J. Cheng, X. Wu, Y.-L. Chen, Y. Ou, and Y. Xu, “An energy model approach to people counting for abnormal crowd behavior detection,” *Neurocomputing*, vol. 83, pp. 121–135, 2012.
- [25] Y. Zhang and F. Chen, “A modified whale optimization algorithm,” *Computer Engineering*, vol. 44, no. 3, pp. 208–213, 2018.
- [26] S. Mirjalili, “Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm,” *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.
- [27] D. Karaboga and C. Ozturk, “A novel clustering approach: artificial Bee Colony (ABC) algorithm,” *Applied Soft Computing*, vol. 11, no. 1, pp. 652–657, 2011.
- [28] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of the ICNN’95—International Conference on Neural Networks*, pp. 1942–1948, Perth, WA, Australia, 1995.
- [29] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, “GSA: a gravitational search algorithm,” *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.