IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 September 2022, accepted 30 October 2022, date of publication 4 November 2022, date of current version 9 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3219486

RESEARCH ARTICLE

Research and Application of a New Lion Swarm Algorithm

WANGSHENG FANG[1], WEI GUO[1], AND YUJIA LIU[2]
[1]School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China
[2]Institute of Artificial Intelligence, Jiangxi University of Applied Science, Nanchang, Jiangxi 330100, China

Corresponding author: Wei Guo (6720200781@mail.jxust.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 62062037.

ABSTRACT To address the defects of the fast convergence speed of the lion swarm algorithm and ease of falling into a local optimum, a variable-speed elastic collision lion swarm algorithm (VELSO) is proposed. First, the flexibility of the lioness search is increased according to the variable spiral search strategy, and then the two lionesses learn learning strategies of teaching and learning algorithms to enhance the interactive behavior of lioness hunting. Then, the improved refraction reverse learning strategy was used to increase the diversity of the population and make the individual quality of the population. Finally, the variable-speed elastic collision strategy is used to increase the probability of the algorithm jumping out of the local optimum and to improve the ability of the algorithm to obtain the optimal solution. To verify the effectiveness of the proposed algorithm, 16 test functions were used to test the proposed algorithm and compared with other algorithms, which proved that the proposed algorithm was very effective. Finally, the proposed algorithm is applied to DV-Hop positioning, which verifies the feasibility and practicability of the proposed algorithm.

INDEX TERMS Refraction learning, elastic collision, lion swarm algorithm, DV-Hop location.

I. INTRODUCTION

Swarm intelligence is an algorithm inspired by natural biota and is generated by simulating the behavioral laws of things or organisms in nature [1]. Owing to their fast convergence, easy implementation, and simple operation [2], they are widely used in industrial engineering [3], [4], production scheduling [5], information communication [6], and other fields. For example, Anter et al. [7] proposed a new model to identify the state of epileptic seizures. This model uses an efficient hybrid genetic whale optimization algorithm (GWOA) based on Naive Bayes (NB-GWOA) for feature selection and uses an adaptive extreme learning machine based on differential evolution algorithm (DE) for classification. In the NB-GWOA algorithm, the genetic algorithm is used to enhance the search effect of the whale optimization algorithm for the optimal solution, and the naive Bayes method is used to determine the fitness function. Finally, it is verified that the proposed model has good identification and performance in classifying have good identification and accuracy in the classification of epileptic states. To detect COVID-2019, Anter and Olive et al. [8] proposed an optimization model for COVID-2019 diagnosis based on fuzzy C-means (AFCM) and Levy distribution improved slime mold algorithm (SMA), which uses the Levy strategy to increase SMA search area, avoid it falling into the local optimum, and obtain a better solution with the minimum number of iterations. In addition, the FCM algorithm was used to segment lung regions from CXR images, and an image intensity histogram was used to reduce the calculation time and quantity in the clustering process. The final experiment shows that the proposed new model is superior to other methods, which is helpful for the early identification of PATIENTS infected with COVID-19. Anter and Gupta et al. [9] proposed a binary model based on the whale optimization algorithm, chaos theory, and fuzzy logic for detecting faults in sensor processes of sewage treatment plants. In this model, the fast fuzzy c-means is used as the cost function to measure the fuzziness and uncertainty of the data, and many different chaotic sequences are used to estimate and adjust the parameters of the WOA

The associate editor coordinating the review of this manuscript and approving it for publication was Giovanni Pau.

algorithm. The results show that the model can detect faults with high accuracy and guide the operator to make control decisions.

With the application of swarm intelligence algorithms in practical life, in recent years, researchers have proposed kinds of swarm intelligence algorithms such as the gray wolf algorithm (GWO) [10], whale optimization algorithm (WOA) [11], sparrow search algorithm (SSA) [12], particle swarm optimization (PSO) [13], [14], a novel nature-inspired algorithm (GSK) [15], which have evolved according to the survival mode of different species in nature [16]. Other algorithms, such as the gravitational search algorithm (GSA) [17] and multiverse algorithm (MVO) [18], are inspired by the changing laws of nature [19]. In 2018, Liu et al. proposed the Lion Swarm optimization algorithm (LSO) [1], which is easy to understand and implement, according to the group division of labor and hunting behavior of lions. Compared with particle swarm optimization, particle swarm optimization, and gravity search algorithm, it has better convergence speed and accuracy. However, there are many defects in the pride algorithm, such as the cooperative hunting behavior of different lionesses, which reduces the scope of global search. As the convergence speed is too fast, it is easy to fall into the local optimum, so it cannot get the optimal solution stably.

To better apply the lion swarm algorithm to various fields, many scholars have made improvements according to the characteristics of the lion swarm algorithm in recent years. For example, Ji et al. [20] introduced the interactive behavior of the SO algorithm to strengthen the update of the female lion position and avoid the problem of diversity reduction and insufficient global search ability caused by common hunting of LSO the algorithm. Meanwhile, combined with the hunting behavior of the GWO algorithm, the global search ability and local exploration ability of the young lion of the LSO algorithm were enhanced, the improved algorithm was applied to node location and the location accuracy is improved. Yang et al. [21] combined chaotic sequence into a lion swarm algorithm to disturb some individuals when the group fell into local optimization and increased the probability of the algorithm jumping out of local optimization, the algorithm was used to optimize the grid search parameters of LSTM neural network, which solved the problem of long time-consuming and achieved good results. Dai et al. [22] balanced the global and local search capabilities of the lion swarm algorithm by adapting the moving step size of the young lion, and the reinforcement learning strategy was utilized to improve the optimization efficiency of the lion swarm algorithm and used the algorithm to improve the accuracy of tool wear prediction. Zhu et al. [23] combined the lion swarm algorithm with the FastSLAM algorithm to improve the positioning accuracy of the robot, Inspired by the particle filter, the location update strategy of Lion King was improved, which avoided the waste of computing resources caused by the original Lion King update algorithm, and improved the accuracy of particle filter, then the crossover strategy of the genetic algorithm was introduced into the lioness update, which optimized the problem of insufficient particle diversity in the later stage of the combination of lion swarm algorithm and FastSLAM algorithm. Wu et al. [24] combined the update strategy of the lion swarm algorithm with the cooperative competition and self-learning operation of the multi-intelligence system to solve the problem that the performance of the lion swarm algorithm is not good in multi-objective problems so that individuals can exchange information and enhance the searchability and convergence speed of lion swarm algorithm. Huang et al. [25] introduced the learning factor of the particle swarm optimization algorithm into the lion swarm algorithm to strengthen the memory of the Lion King position update, to improve the optimization speed of the algorithm, Then the Solis & Wets (SW) algorithm was applied to the lion swarm algorithm to improve the probability of jumping out of the local optimization, therefore, the improved algorithm is used to optimize the accuracy and rate of shop floor scheduling, and good results have been achieved. Zhang et al. [26] used the lion swarm algorithm to solve the traveling salesman problem, By introducing discrete coding and an ordered crossover operator, the lion swarm algorithm is successfully applied to discrete problems, Then C2 opt algorithm was introduced to strengthen the local search ability of lion swarm algorithm, Finally, a multi-population parallels lion swarm algorithm is proposed, which used the ring topology to strengthen the information interaction between different populations and improve the running speed of the algorithm. Qiao et al. [27] applied a genetic algorithm to the lion swarm algorithm, when the lion swarm algorithm fell into local optimization in the later stage, some lion swarm individuals are cross mutated to obtain a new population, to jump out of the local optimization, then the improved lion swarm algorithm was used to optimize the traditional least squares support vector machine model, so as to solve the instability of carbon dioxide emission prediction.

So far, the research on the lion swarm algorithm is far from over, although previous researchers have strengthened the hunting mechanism and global search in the algorithm, However, there is no better solution to the convergence speed and local convergence of the lion swarm algorithm. Therefore, a variable speed elastic collision lion swarm algorithm (VELSO) is proposed in this paper. The innovations are as follows:

(1) Firstly, the improved refraction reverse learning strategy is adopted to increase the diversity of the population, improve the traversal effect of the algorithm in the search space, and increase the search breadth of the population in the algorithm.

(2) Secondly, using the variable spiral search strategy to achieve cooperative hunting to improve the search accuracy. Considering that the lionesses cooperate with each other to search in the lion group algorithm, it will reduce the search space and efficiency. The algorithm in this paper proposes that lionesses should first conduct a more flexible searches in different directions to enhance the exploration ability of adult

individuals in unknown areas. Then, according to the teaching and learning optimization algorithm, the interactive behavior of lionesses to assist each other in hunting is strengthened to enhance the mining ability of local locations.

(3) Finally, according to the elastic collision principle, a variable speed elastic collision strategy is proposed for LSO's characteristics of fast convergence and easy fall into local optimal. Using the elastic collision between the worst position individual and the optimal individual, the probability of the algorithm jumping out of the local optimum is increased to obtain better-quality individuals.

In the first section, this paper introduces the research background, application fields, and contributions of researchers of intelligent algorithms this year. The second section introduces the basic lion swarm algorithm. Section 3 describes the innovations of this paper in sequence. Section 4 analyzes the effectiveness of the algorithm. Section 5 analyzes the time complexity of the algorithm in this paper. Section VI tests the VELSO algorithm in 16 test functions and compares the test results with eight different algorithms to verify the performance of the VLESO algorithm. Section 7 applies the proposed algorithm to DV-Hop positioning to verify the feasibility and practicability of the VELSO algorithm. Finally, this paper summarizes the work, and makes plans and prospects for future work.

II. LION SWARM ALGORITHM

In the process of solving the global optimization problem of the objective function by the lion swarm algorithm, the lion swarm Race is mainly divided into three categories: Lion King, Lioness, and Young Lion. They have different social behaviors. According to reference [1], the lion king and the lioness are adult lions, and the number of adult lions will affect the difference in the algorithm population size and the convergence speed, in order to maintain the effect of the algorithm, the proportion of adult lions τ generally less than 0.5, and the proportion of young lions remains between 0.5 and 1.

The lion king is the strongest male lion in the population, responsible for the safety of the population, driving away foreign invaders, and distributing food. The updated formula of the Lion King's position was as follows:

$$X_i^{t+1} = g^t(1 + \gamma \left\| P_i^t - g^t \right\|) \tag{1}$$

In the Lion King position update, t represents the current number of iterations. X_i^{t+1} represents the new location generated after the update. g^t represents the optimal position of the t-generation population. γ is according to the normal distribution $N(0, 1)$ generated random numbers. P_i^t represents the historical optimal location of the *ith* lion in the t generation. The lioness is a hunting lion. It cooperates with each other in hunting, give the best food to the lion king, and are also responsible for leading the cubs to learn to hunt. The lioness position update formula is as follows:

$$X_i^{t+1} = \frac{(P_i^t + p_c^t)}{2}(1 + \alpha_f \gamma) \tag{2}$$

In the lioness position update formula, X_i^{t+1} indicates the location of the lioness after the update. P_c^t represents the best position in the history of randomly selecting a lioness to cooperate with hunting in the t generation. γ is according to the normal distribution N (0, 1) Generated random numbers. α_f represents the step control factor of lioness position update, and the formula is as follows:

$$\alpha_f = step \cdot \exp(-30t/t_{\max})^{10} \tag{3}$$

In formula (3), $step = 0.1(H - L)$ represents the maximum moving step of the lioness's range of motion, where H and L represent the upper and lower bounds of lion group space respectively. t_{max} represents the maximum number of iterations of the algorithm.

young lions have three main behaviors: (1) Hungry will eat near the lion king. (2) When the cubs are full, they will learn to hunt with the lioness. (3) As an adult, it was expelled from the territory by the lion king and challenged the position of the lion king after suffering. The formula for updating the young lion's position is as follows:

$$X_i^{t+1} = \begin{cases} \frac{g^t + P_i^t}{2}(1 + \alpha_c \gamma), & q \le \frac{1}{3} \\ \frac{P_m^t + P_i^t}{2}(1 + \alpha_c \gamma), & \frac{1}{3} \le q < \frac{2}{3} \\ \frac{g^{t'} + P_i^t}{2}(1 + \alpha_c \gamma), & \frac{2}{3} \le q < 1 \end{cases} \tag{4}$$

In young lion position update formula, X_i^{t+1} indicates the location of the young lions after the update. P_m^t represents the best position in the history of the *tth* generation when the young lions follow the female lion to learn hunting. α_c represents the step control factor for the position update of the young lion, $\alpha_c = step*(1-t/t_{max})$. $g^{t'}$ adopts the idea of elite reverse learning, which means that the expelled lion cubs are far away from the position of the lion king, $g^{t'} = H + L - g^t$. q is a probability factor, a random number generated by the uniform distribution $U(0, 1)$.

III. VELSO

A. REVERSE LEARNING STRATEGY BASED ON THE REFRACTION PRINCIPLE

The reverse learning strategy is a learning strategy that strengthens the population to explore the unknown field. It has strong adaptability and can be applied in various algorithms to enhance the diversity of the algorithm population and the global search ability. However, at present, general reverse learning strategy in literature [28], [29] only calculates the reverse solution of an individual, which has limitations. At the same time, such as the mirror reverse learning strategy in the literature [30], although the mirror principle is added, only one parameter can be used to adjust the distance of the reverse learning, which cannot effectively increase the flexibility of the reverse learning strategy. Therefore, an improved refraction reverse learning strategy has been proposed, which uses the refraction index of the refraction principle to change the reverse position of the algorithm according to the number of iterations. The principle is as follows:

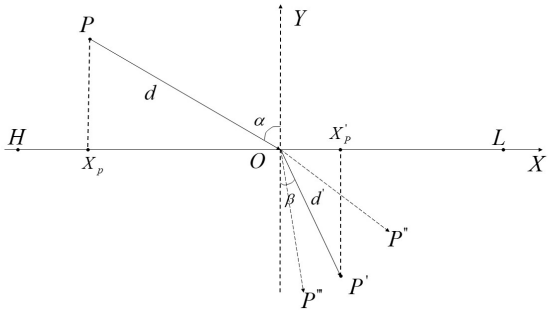

FIGURE 1. Schematic diagram of refraction reverse learning.

As shown in Figure 1, taking the X-axis as the dividing line, the upper part of the X-axis is the vacuum of nature, and the lower part is other media. The range of the search interval on the X-axis is $[L, H]$, O is the midpoint of H and L, the line passing through the midpoint perpendicular to the X-axis is the Y-axis, the light is incident from the point P, and the projection of the point P on the X-axis is X_P and the angle of incidence is α. The point P' is obtained by refraction, the refraction angle is β, the projection on the X-axis is X'_P, and the lengths of the incident ray and the refracted ray are d and d' respectively. From this, the refractive index η can be obtained.

$$\eta = \frac{\sin\alpha}{\sin\beta} = \frac{\frac{H+L}{2} - X_p}{X'_P - \frac{H+L}{2}} \cdot \frac{d'}{d} \tag{5}$$

When set $d/d' = c$, then (5) can be transformed into

$$X'_P = \frac{H+L}{2} + \frac{H+L}{2c\eta} - \frac{X_P}{c\eta} \tag{6}$$

When $\eta = 1$, we can get:

$$X'_P = \frac{H+L}{2} + \frac{H+L}{2c} - \frac{X_P}{c} \tag{7}$$

When $\eta = 1$ and $c = 1$, we can get:

$$X_P = H + L - X'_P \tag{8}$$

The above formula (7) and formula (8) are the lens reverse learning strategy in the literature [31] and the general reverse learning strategy in the literature [32], [33]. The new individuals generated by the general reverse learning strategy are fixed, and this learning algorithm has the risk of falling into a local optimum in a high-dimensional search space. The lens imaging reverse learning strategy is very similar to the refraction reverse learning strategy, the difference is that the former controls the position of the new individual through c, and the latter uses the refractive index η and c to adjust the position of the new individual. We introduce a dynamic control parameter η^* to enhance the algorithm's exploration of unknown regions. Changing the value of the control parameter η helps the generation of new individuals and increases the diversity of the population. The expression

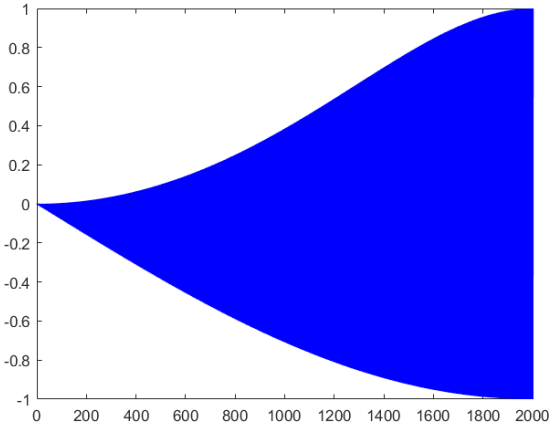

FIGURE 2. Refraction disturbance control diagram.

of η^* is as follows, and the disturbance control diagram of η^* is shown in Figure 2.

$$X'_P = \frac{H+L}{2} + \frac{H+L}{2c\eta} - \frac{X_P}{c\eta} \tag{9}$$

$$\begin{cases} \eta* = \eta + \sin(\frac{\pi}{2}(-t/t_{\max})^k) \\ k = 1, t \ \ is \ \ Odd \ number \\ k = 2, t \ \ is \ \ Even \end{cases} \tag{10}$$

In formula (9), X'_P is the newly generated individual, H and L are the upper and lower bounds of the search space respectively, t is the current iteration number, t_{max} is the maximum iteration number, k is the refraction control factor, which is used to control the disturbance direction of η^*. As shown in Figure 2, it is the perturbation control diagram of η^*. According to the parity of the number of iterations, η^* is perturbed in different directions. When the number of iterations is odd, k takes 1, as shown in Figure 1, the reverse learning of refraction changes from the position P' to P'', when the number of iterations is even, k takes 2, and refractive inversion learning changes from position P' to P'''.

B. COOPERATIVE HUNTING STRATEGY BASED ON VARIABLE SPIRAL SEARCH

In the lion swarm algorithm, the lioness hunts and gives the best food to the lion king and the lioness also leads the cubs to learn to hunt. The position update of the lioness has a huge impact on the effect of the entire algorithm. However, the update of the position of the lioness by the lion swarm algorithm relies on two lionesses hunting together, which reduces the diversity of the population and the effect of the algorithm on the global search. Therefore, the variable spiral search strategy [34] is used to make the lioness conduct the variable spiral search in different directions before hunting. While balancing the lioness's global and local search capabilities, increases the lioness's ability to explore unknown areas, and finally allows the two lionesses to hunt together according to the learning stage of the teaching and learning algorithm [35]. By improving the quality of the new individuals of the lioness,

the quality of the new individuals of the lion cub and the lion king is improved. The updated formula for the position of the lioness is as follows:

$$\begin{cases} X_i^{t+1} = D_i \cdot e^{bL} \sin(2\pi L) \\ X_j^{t+1} = D_j \cdot e^{bL} \cos(2\pi L) \\ D = |gbest - X| \end{cases} \tag{11}$$

$$b = \exp(-5 \cdot (1 - t/t_{\max})) \tag{12}$$

In formula (11), X_i is the lioness i, and X_j is the lioness j. D is the distance from the prey to the lioness. b is the spiral control parameter, which balances the search ability of the algorithm in the early stage and the later stage according to the number of iterations. L is a random value generated from (-1, 1). *gbest* is the contemporary optimal position. Use formula (12) to update the position of lioness i and lioness j to increase the lioness's exploration ability in order to obtain a better prey position.

After the lioness performs a global search, lioness i and lioness j hunt together. When lioness i is in a better position, lioness j moves closer to lioness i to hunt, and the positions of a lioness i and lioness j are updated. The formula is as follows:

$$\begin{cases} X_i^{t+1} = X_i^t + r_1 \cdot (X_i^t - X_j^t) \\ X_j^{t+1} = X_j^t + r_2 \cdot (X_i^t - X_j^t) \end{cases} \tag{13}$$

When the position of lioness j is better than that of lioness i, lioness i approach lioness j and hunt together. The position update formula of lioness i and lioness j is as follows:

$$\begin{cases} X_i^{t+1} = X_i^t + r_1 \cdot (X_j^t - X_i^t) \\ X_j^{t+1} = X_j^t + r_2 \cdot (X_j^t - X_i^t) \end{cases} \tag{14}$$

In formulas (13) and (14), X_i is the position of a lioness i and X_j is the position of lioness j. r_1 and r_2 are random values generated by (0, 1).

C. VARIABLE SPEED ELASTIC COLLISION STRATEGY

An elastic collision is the knowledge of mechanics in physics. It means that ball A collides with stationary ball B at the speed of V_0, and the object does not deform or lose kinetic energy after the collision [18]. There are three situations when two balls A and B collide:

(1) When the mass of ball A is equal to the mass of ball B, ball A is at rest, and ball B moves in the positive direction with velocity V_0.

(2) When the mass of ball A is greater than that of ball B, ball A and ball B move in the positive direction.

(3) When the mass of ball A is less than the mass of ball B, ball A moves in the opposite direction, and ball B moves in the positive direction.

In this paper, the worst individual in each generation is given a certain initial velocity, to hit the optimal particle in each generation at a certain velocity, compares the fitness value of the individual position after the collision, and retains the individual with a better position. The formula for the dynamic initial velocity is as follows:

$$V_0 = T \cdot e^{(-(X_{worst} - X_{\text{rand}})^2)} \tag{15}$$

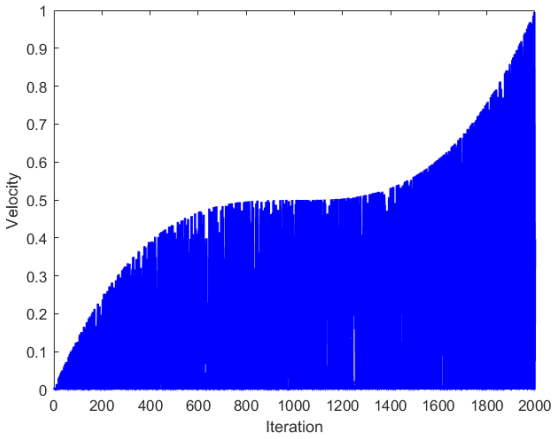

FIGURE 3. Variation diagram of initial speed.

$$T = \frac{1}{2}((\frac{2t}{t_{\max}} - 1)^3 + 1) \tag{16}$$

X_{worst} is the position of the worst individual and X_{rand} is the position of the randomly selected individual in formula (15) and (16) According to the difference between these two individuals, the position information of the group will be perceived, the position information of the specific group will be mastered, and the speed adaptability will get improved. T is the nonlinear control factor controlling the adaptability of initial velocity, where t is the current iteration number and t_{max} is the maximum iteration number. The speed change diagram of initial speed V_0 is shown in Figure 3.

It can be seen from Figure 3 that with the increase in the number of iterations, the initial speed gradually increases, which is conducive to the algorithm jumping out of the local optimum in the later stage. However, in the process of location updating, the lion swarm algorithm only considers the location of the best individual of each generation or the update of the optimal position in the history of the race. This will ignore the impact of the worst location on the algorithm, which results in the distance between the worst individual and the randomly selected individual being too far, as a result, the preset initial speed is too small or even close to 0 during an elastic collision so that the amplitude of collision is too small or there is no collision. Finally, the purpose of making the optimal individual jump out of the local optimum cannot be achieved. Therefore, inspired by the universal gravitation of the gravity search algorithm in the literature [19], [37], [38], we assume that there is a gravity F between the optimal individual and the worst individual, Fixing the optimal individual allows the worst individual to be affected by force F to accelerate the movement, then we can the final velocity V, and reduce the probability that the individual velocity is 0. Besides, the collision between two individuals increases the probability that the optimal individual jumps out of the local optimal. The specific idea of variable speed elastic collision strategy is shown in Figure 4.

In Figure 4, supposing that the fitness value of the worst individual is m_1 and the fitness value of the best individual

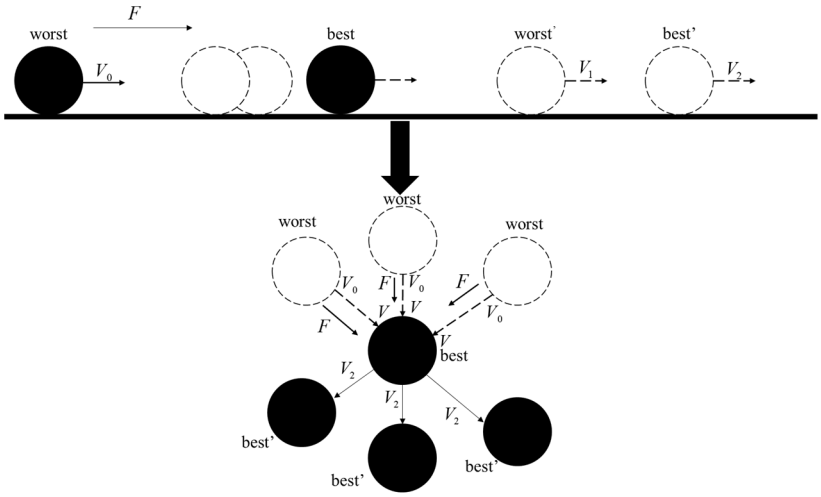

FIGURE 4. Schematic diagram of variable speed elastic collision.

is m_2. We assume that the optimal individual m_2 is stationary and give the worst individual m_1 a dynamic initial velocity. The worst individual performs acceleration motion under the influence of the initial velocity and the force F to obtain the final velocity V. The formula of the force F is as follows:

$$F = \frac{G \cdot m_1 \cdot m_2}{(R_{wb} + \varepsilon)^2} \cdot (X_{worst} - X_{best}) \tag{17}$$

$$G = \exp(-40 \cdot (t_{\max} - t/t_{\max})) \tag{18}$$

Formulas (17) and (18) are the formulas of force F, in which G is the gravitational constant. R_{WB} is the Euclidean distance between the worst and the best individuals. ε is to prevent the denominator from being 0 and obeying the standard normal distribution. X_{worst} represents the position of the worst individual and X_{best} represents the position of the best individual. t and t_{max} are the current number of iterations and the maximum number of iterations, respectively.

According to Newton's second law, the relationship between acceleration and force F can be obtained as follows:

$$a = \frac{F}{m_1} \tag{19}$$

According to the acceleration and initial speed, we can get the speed V when the worst individual reaches the optimal individual position. The formula of speed V is as follows:

$$V = T \cdot \exp(-(rand \cdot V_0 + a)^2) \tag{20}$$

In formula (20), T is the adaptive nonlinear control factor in formula (16). a is the acceleration. V_0 is the initial speed. *rand* is a random value of (0, 1). The speed variation diagram of speed V is shown in Figure 5.

It can be seen from Figure 5 that after acceleration, the problem of speed close to 0 is reduced, the probability of elastic collision is increased, and then the probability of the algorithm jumping out of the local optimum is increased. The gradually increasing speed is also conducive to the algorithm

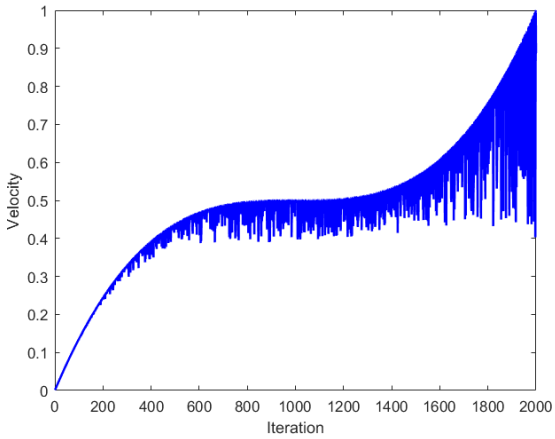

FIGURE 5. Variation diagram of final speed.

reducing the probability of falling into the local optimum in the later stage.

The worst individual impacts the best individual with velocity V after acceleration. According to the elastic collision and the law of momentum conservation:

$$\begin{cases} m_1 V = m_1 V_1 + m_2 V_2 \\ \frac{1}{2} m_1 V^2 = \frac{1}{2} m_1 V_1^2 + \frac{1}{2} m_2 V_2^2 \end{cases} \tag{21}$$

From the above formula, we can get the speed V_1 of the worst individual after the collision and the speed V_2 of the best individual after the collision. The formula is as follows:

$$V_1 = \frac{m_1 - m_2}{m_1 + m_2} \cdot V \tag{22}$$

$$V_2 = \frac{2m_1}{m_1 + m_2} \cdot V \tag{23}$$

Inspired by the particle swarm optimization algorithm in literature [13], [14], the individual position plus the speed of the individual is equal to the new position of the individual. The

new positions of the worst individual and the best individual after a collision are obtained as follows:

$$X'_{worst} = X_{worst} + V_1 \tag{24}$$

$$X'_{best} = X_{best} + V_2 \tag{25}$$

X_{worst} represents the position of the worst individual and X_{best} represents the position of the best individual. Compare the individual before the collision with the individual after the collision and reserve the individual with a good position.

D. STEPS OF VELSO ALGORITHMS

To sum up, the variable-speed elastic collision lion-swarm algorithm is proposed to solve the problem of fast convergence speed, weak global search ability, and easily falling into local optimum. In the lioness position update stage, a strategy based on variable helical search was proposed to strengthen the lioness's global search ability, better coordinate hunting, and then drive the position update of the lion king and the lion cubs. At the same time, refraction reverse learning is improved, the diversity of races is increased, and the individual's ability to explore unknown areas is improved. Finally, an elastic collision strategy is proposed to make the algorithm have the ability to jump out of the local optimum. The idea of variable speed increases the probability of individuals jumping out of the local optimum. And aiming at the problem of small initial velocity, the idea of variable speed is proposed to increase the probability of individuals jumping off the local optimum. The specific implementation steps are as follows:

Step 1: population initialization, determine the population size, the number of adult lions and cubs, and the number of iterations.

Step 2: Calculate the fitness value, save the position of the best individual and the worst individual and the fitness value.

Step 3: Update the location of the lion king. A strategy based on the variable spiral search is used to update the position of the lioness as well as the position of the cub and perform boundary processing.

Step 4: Update the position of the individual according to the improved refraction reverse learning strategy, retain the individual with a better position, perform boundary processing and calculate the fitness value, and update the optimal position of each generation and the historical optimal position of the individual.

Step 5: Select the optimal position and the worst position of the current generation.

Step 6: Update the optimal and worst individuals according to the variable-speed elastic collision strategy.

Step 7: Determine whether the maximum number of iterations has been reached, if so, output the optimal position, if not, continue to perform the third step.

IV. ALGORITHM EFFECTIVENESS ANALYSIS

In order to intuitively show the effectiveness of the VELSO algorithm, we compare the VELSO algorithm with the LSO algorithm in the unimodal function sphere function and multimodal function Ackley function. The population number is 100, the number of iterations is 10, and the population distribution after running for 10 times is shown in Figure 6.

(a) and (b) in Figure 6 are the population distribution diagram of the VELSO algorithm and LSO algorithm under the Sphere Function. (c) and is the population distribution map of the VELSO algorithm and LSO algorithm under the Ackley Function. It can be seen from Figures (a) and (b) that under the single-peak test function, the VELSO algorithm has the characteristics of multi-directional convergence, fast convergence speed, and high convergence accuracy. It can be seen from Figures (c) and (d) that the VELSO algorithm also has the advantages of fast convergence speed and high convergence accuracy under the multimodal test function. Therefore, it is verified that the algorithm in this paper has more advantages in two different types of test functions.

V. TIME COMPLEXITY ANALYSIS

Time complexity is one of the effective indicators to analyze the effect of the algorithm. It can determine the amount of calculation of the algorithm from both macro and micro perspectives. Suppose the population number of the algorithm is P, the dimension of the function is D, and the maximum number of iterations is T. From a macro perspective, the time complexity of the LSO algorithm is $O(P \times D \times T)$, while VELSO algorithm does not add additional cycles, and the time complexity does not change.

From the microscopic point of view, it is assumed that the proportion of the lioness is Nc, and the proportion of cubs is Np, the calculation time of the original lioness position update is t, the calculation time of cooperative hunting for a variable spiral search is t_1, the calculation time of the improved refraction learning strategy is t_2,, and the calculation time of the variable speed elastic collision strategy is t_3. Then the update of the lioness position increases by $O(P \times D \times T \times Nc \times (t_1 - t))$, $O(T \times t_2)$ is added in the improved refraction reverse learning stage, in variable speed elastic collision stage increase $O(T \times t_3)$. Although the total time complexity of VELSO algorithm increases by $O(T \times (P \times D \times (t_1 - t) \times NC + t_2 + t_3))$, it does not cause an order of magnitude improvement, which is worth the sacrifice compared with the improvement of algorithm performance.

VI. EXPERIMENT AND ANALYSIS

In order to verify the feasibility and effectiveness of the VELSO algorithm, 16 benchmark functions were selected to test the VELSO algorithm according to the literature [2,26 ∼ 27,32]. The test function information is shown in Table 1.

At the same time, in order to verify the effect of this algorithm, eight algorithms are selected and compared with this algorithm. They are the improved lion swarm optimization algorithm (ILSO) proposed by Ji et al. [20], the lens learning sparrow search algorithm (LLSSA) proposed by Ouyang et al. [30], the Multi-Strategy sparrow search algorithm (IHSSA) proposed by Wang et al. [31], Gray Wolf algorithm (GWO), lion swarm optimization algorithm (LSO), particle swarm optimization (PSO), whale optimization algorithm

TABLE 1. 16 benchmark functions.

Function	Dim	Interval	Min
$F_1(x)=\sum_{i=1}^{n}x_i^2$	30	[-100,100]	0
$F_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}x_i$	30	[-10,10]	0
$F_3(x)=\sum_{i=1}^{n}(\sum_{i=1}^{i}x_j^2)^2$	30	[-100,100]	0
$F_4(x)=\max_i\{\lvert x_i\rvert,1\le i\le n\}$	30	[-100,100]	0
$F_5(x)=\sum_{i=1}^{n}ix_i^4+random[0,1)$	30	[-1.28,1.28]	0
$F_6(x)=\sum_{i=1}^{n}-x_i\sin(\sqrt{\lvert x_i\rvert})$	30	[-500,500]	-418.9829n
$F_7(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	30	[-5.12,5.12]	0
$F_8(x)=-20e^{(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})}-e^{((\frac{1}{n})\sum_{i=1}^{n}\cos(2\pi x_i))}+20+e$	30	[-32,32]	0
$F_9(x)=(1/500+\sum_{i=1}^{25}(1/i+(x_1-a_{1i})^6+(x_2-a_{2i})^6))^{-1}\ where$ $a=\begin{pmatrix}-32 & -16 & 0 & 16 & 32 & -32 & K & 0 & 32 & 32\\ -32 & -32 & -32 & -32 & -32 & -16 & K & 32 & 32 & 32\end{pmatrix}$	2	[-65.536,65.536]	0.998
$F_{10}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$ $a=(0.1975,0.1947,0.1735,0.16,0.0844,$ $0.0627,0.0456,0.0342,0.0323,0.235,0.246)$ $b=(0.25,0.5,1,2,4,6,8,10,12,14,16)$	4	[-5,5]	0.0003075
$F_{11}(x)=-\sum_{i=1}^{m}\left(\sum_{j=1}^{n}(x_j-a_{ji})^2+c_i\right)^{-1},where$ $a=\begin{pmatrix}4&1&8&6&3&2&5&8&6&7\\4&1&8&6&7&9&3&1&2&3.6\\4&1&8&6&3&2&5&8&6&7\\4&1&8&6&7&9&3&1&2&3.6\end{pmatrix}$	4	[0,10]	-10.5364
$F_{12}(x)=418.9829n-\sum_{i=1}^{d}x_i\sin(\sqrt{\lvert x_i\rvert})$	30	[-500,500]	0
$F_{13}(x)=\sum_{i=1}^{n}x_i^2+\left(\sum_{i=1}^{n}0.5ix_i\right)^2+\left(\sum_{i=1}^{n}0.5ix_i\right)^4$	10	[-5^{10},10^{10}]	0
$F_{14}(x)=2x_1^2-1.05x_1^4+\frac{x_1^6}{6}+x_1x_2+x_2^2$	2	[-5,5]	0
$F_{15}(x)=\sum_{i=1}^{n/4}\left[\begin{matrix}(x_{4i-3}+10x_{4i-2})^2+5(x_{4i-1}-x_{4i})^2+\\(x_{4i-2}-2x_{4i-1})^4+10(x_{4i-3}-x_{4i})^4\end{matrix}\right]$	30	[-4,5]	0
$F_{16}(x)=x_1^2+10^6\sum_{i=2}^{n}x_i^2$	10	[-10^{10},10^{10}]	0

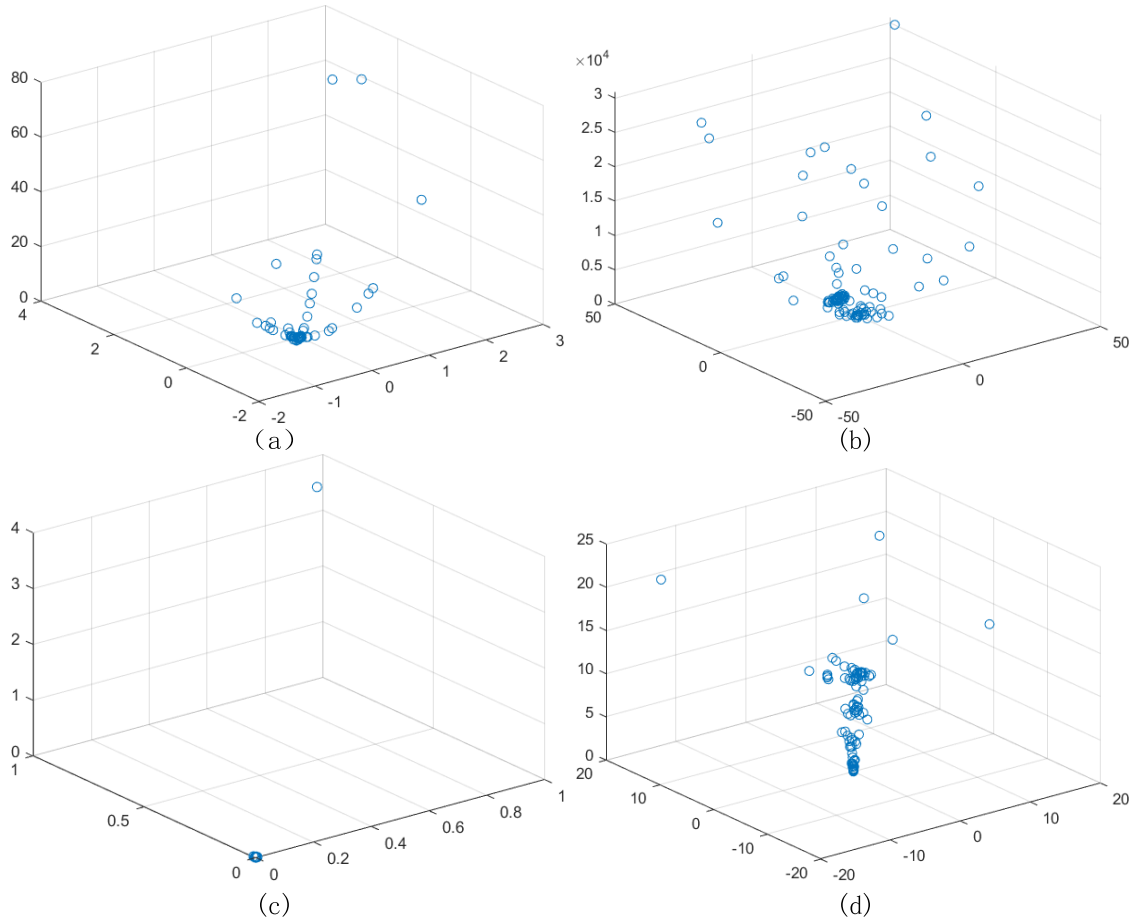

FIGURE 6. Population distribution.

TABLE 2. Parameter settings of each algorithm.

Algorithm	Parameter
VELSO	$\tau = 0.3$
ILSO	$\tau = 0.3$
LLSSA	$PD = 0.2, ST = 0.6, SD = 0.2$
IHSSA	$PD = 0.2, ST = 0.6, SD = 0.2$
GWO	$a_{max} = 2, a_{min} = 0$
LSO	$\tau = 0.3$
PSO	$c_1 = c_2 = 1.49445$
SSA	$PD = 0.2, ST = 0.6, SD = 0.2$
WOA	$P = 0.5$

(WOA). Meanwhile, in order to ensure fairness among algorithms, the population of all algorithms is 100, the maximum number of iterations is 200, and the parameters of the algorithm are original parameters, as shown in Table 2. Each algorithm is tested on MATLAB 2020a, windows 10 operating system, and 8G memory. Moreover, in order to ensure the authenticity of the algorithm test, each algorithm runs independently 30 times under each test function. The performance of the algorithm is analyzed from the perspectives of optimal value, worst value, average value, standard deviation, median value, and run time. The convergence diagram of different algorithms under different test functions is shown in Figure 7, and the test values of different algorithms under different test functions are shown in Table 3.

It can be seen from Table 3 that compared with other algorithms, the VELSO algorithm can find the optimal value in F_1-F_4, F_7, and F_{12}-F_{16}, and has a strong search performance. Except for F_{12}, for the rest of the test functions, the VELSO algorithm has strong stability. For F_{5-6}, and F_8, although these three functions have not searched for the optimal value, the VELSO algorithm has better searchability than other algorithms. And compared with other algorithms, among the 16 test functions, except for F_{10} and F_{11}, the convergence accuracy of the VELSO algorithm is the best. Among these test functions, only the LLSSA algorithm in F1 has the same optimal value as the algorithm in this paper, and in F_2 and F_4 only the SSA algorithm has the same optimal value as the algorithm in this paper. In F_3, F_{14}, and F_{16}, the three algorithms of LLSSA, IHSSA, and SSA have the same optimal value as the algorithm in this paper. Although the VELSO algorithm has more advantages on F_5, the advantages are not particularly obvious. On F_6 the ILSO and WOA algorithms have the same optimal values as the VELSO algorithm.

TABLE 3. Optimization results of different algorithms.

Fun	Algorithm	Best	Worst	Avg	Stg	Median	Run Time
F1	VELSO	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.00421993
	ILSO	6.0036E-15	6.5400E-10	5.6000E-11	1.6294E-10	9.8028E-13	0.00264921
	LLSSA	0.000E-000	1.804E-172	6.014E-174	0.000E-000	0.000E-000	0.00104028
	IHSSA	5.233E-251	1.569E-249	0.000E-000	0.000E-000	0.000E-000	0.00202993
	GWO	5.7328E-15	2.3019E-13	4.3067E-14	4.4309E-14	2.8838E-14	0.00081358
	LSO	5.0574E-41	5.1838E-34	5.2144E-35	1.4622E-34	5.0347E-37	0.00264122
	PSO	4.1800E-04	8.7610E-03	2.2290E-03	1.6440E-03	1.9520E-03	0.00036091
	SSA	2.882E-245	9.645E-244	0.000E-000	0.000E-000	0.000E-000	0.00089752
	WOA	1.5018E-43	3.8570E-37	3.0218E-38	8.1033E-38	4.1841E-40	0.00031619
F2	VELSO	0.000E-000	6.945E-167	2.679E-168	1.122E-167	4.663E-168	0.00409996
	ILSO	1.8125E-09	8.1526E-06	8.8114E-07	1.6148E-06	2.6686E-07	0.00535788
	LLSSA	1.559E-162	1.371E-101	4.592E-103	2.461E-102	6.551E-148	0.00105082
	IHSSA	9.874E-163	3.442E-124	1.147E-125	6.178E-125	8.759E-162	0.00189422
	GWO	1.8114E-09	2.8734E-08	7.7910E-09	5.2462E-09	6.7026E-09	0.00089418
	LSO	2.5649E-53	2.2638E-49	2.4405E-50	4.9458E-50	7.0671E-51	0.00251373
	PSO	2.4016E-02	4.5390E-01	9.8416E-02	7.8315E-02	7.6072E-02	0.00038218
	SSA	0.000E-000	2.812E-109	9.374E-111	5.048E-110	1.064E-301	0.00096322
	WOA	3.7640E-27	7.9819E-22	4.1113E-23	1.4548E-22	4.2474E-24	0.00035287
F3	VELSO	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.00866747
	ILSO	2.1387E-07	1.1166E-00	6.9167E-02	2.1173E-01	5.9916E-04	0.01013107
	LLSSA	0.000E-000	3.159E-156	1.053E-157	0.000E-000	1.851E-291	0.00249062
	IHSSA	0.000E-000	1.438E-178	4.795E-180	0.000E-000	0.000E-000	0.00404241
	GWO	1.9450E-03	3.1085E-01	2.3789E-02	5.5876E-02	8.3360E-03	0.00210359
	LSO	5.5575E-40	2.4177E-33	9.9408E-35	4.3270E-34	2.5720E-36	0.00355122
	PSO	6.2146E+01	1.7392E+02	1.1138E+02	3.0460E+01	1.0824E+02	0.00156479
	SSA	0.000E-000	2.034E-232	6.781E-234	0.000E-000	0.000E-000	0.00208809
	WOA	1.9033E+04	6.5312E+04	4.0942E+04	1.1154E+04	3.9907E+04	0.00148888
F4	VELSO	0.000E-000	4.538E-167	6.471E-168	0.000E-000	2.037E-168	0.00423567
	ILSO	2.5319E-11	1.1569E-05	1.3037E-06	2.4658E-06	2.5844E-07	0.01296969
	LLSSA	1.778E-164	3.041E-112	1.016E-113	5.458E-113	8.282E-163	0.00117719
	IHSSA	5.057E-165	8.084E-135	2.695E-136	1.451E-135	4.164E-163	0.00176503
	GWO	2.6600E-04	2.3160E-03	1.0470E-03	5.8000E-04	8.4500E-04	0.00084665
	LSO	3.8093E-22	7.4533E-18	1.2421E-18	2.1353E-18	3.5089E-19	0.00243278
	PSO	6.9868E-01	2.4281E-00	1.2998E-00	3.9323E-01	1.2228E-00	0.00035794
	SSA	0.000E-000	1.307E-170	4.356E-172	0.000E-000	1.105E-233	0.00086150
	WOA	5.2200E-03	8.5171E+01	3.4170E+01	2.9175E+01	2.9606E+01	0.00029393
F5	VELSO	1.5987E-06	1.2408E-04	4.1560E-05	3.3078E-05	3.5869E-05	0.00588125
	ILSO	2.1773E-05	7.1695E-04	2.5967E-04	1.7840E-04	2.0530E-04	0.01675776
	LLSSA	4.9188E-06	1.5845E-03	3.6888E-04	3.9966E-04	2.0794E-04	0.00226066
	IHSSA	2.2870E-05	1.6355E-03	3.1686E-04	3.1339E-04	2.3809E-04	0.00261393
	GWO	6.0100E-04	4.6300E-03	2.1190E-03	1.1800E-03	1.9630E-03	0.001440109
	LSO	1.8737E-05	4.0978E-04	1.7405E-04	1.1650E-04	1.4124E-04	0.003015725
	PSO	2.9231E-02	8.3860E-02	5.1490E-02	1.2694E-02	5.0234E-02	0.000907832
	SSA	8.9942E-06	9.2975E-04	2.8009E-04	2.4622E-04	2.4705E-04	0.001548576
	WOA	8.8420E-06	7.6684E-03	2.2713E-03	1.9761E-03	1.9686E-03	0.000863467
F6	VELSO	-12569.500	-12568.800	-12569.400	1.1980E-01	-12569.500	0.00551987
	ILSO	-12569.500	-12569.500	-12569.500	2.4520E-03	-12569.500	0.01978808
	LLSSA	-10304.800	-7597.7900	-9054.5300	6.7131E+02	-9049.4600	0.00147453
	IHSSA	-12569.300	-5470.8200	-8554.7600	1.8145E+03	-8510.5800	0.00209474
	GWO	-7665.5700	-4566.7100	-6340.8700	7.2416E+02	-6311.4300	0.00100159
	LSO	-11475.200	-6331.4700	-9440.8600	1.3176E+03	-9610.4500	0.00262247
	PSO	-6793.3200	-3215.3800	-5098.0000	9.4473E+02	-4998.4100	0.00055405
	SSA	-9460.2900	-7183.5000	-8107.2200	6.3094E+02	-8096.1500	0.00106202
	WOA	-12569.500	-7970.7500	-10728.600	1.5436E+03	-10884.800	0.00043039
F7	VELSO	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.00400593
	ILSO	0.000E-000	1.6644E-09	6.9364E-11	2.9949E-10	2.9843E-12	0.02263796
	LLSSA	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.00118856
	IHSSA	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.00196188
	GWO	2.9542E-10	2.3147E+01	7.3032E-00	4.8700E-00	6.3915E-00	0.00091383
	LSO	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.00252409
	PSO	3.1686E+01	7.6815E+01	5.3716E+01	1.3354E+01	5.4271E+01	0.00043620
	SSA	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.00092639
	WOA	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.00034022
F8	VELSO	8.8818E-16	8.8818E-16	8.8818E-16	0.000E-000	8.8818E-16	0.00415865
	ILSO	1.3075E-09	3.4712E-06	7.2619E-07	8.1012E-07	4.1061E-07	0.02534400
	LLSSA	8.8818E-16	8.8818E-16	8.8818E-16	0.000E-000	8.8818E-16	0.00206693
	IHSSA	8.8818E-16	8.8818E-16	8.8818E-16	0.000E-000	8.8818E-16	0.00194581

TABLE 3. *(Continued.)* **Optimization results of different algorithms.**

	GWO	1.8374E-08	1.1478E-07	4.8561E-08	2.2064E-08	4.4672E-08	0.00095469
	LSO	8.8818E-16	8.8818E-16	8.8818E-16	0.000E-000	8.8818E-16	0.00254837
	PSO	1.1177E-02	1.3480E-00	3.7783E-01	2.4353E-01	5.6100E-02	0.00043209
	SSA	8.8818E-16	8.8818E-16	8.8818E-16	0.000E-000	8.8818E-16	0.00095997
	WOA	8.8818E-16	1.5099E-14	7.0462E-15	3.4241E-15	7.9936E-15	0.00034836
	VELSO	9.9800E-01	9.9850E-01	9.9802E-01	9.5361E-05	9.9800E-01	0.01656017
	ILSO	9.9800E-01	9.9800E-01	9.9800E-01	1.8578E-16	9.9800E-01	0.03256613
	LLSSA	9.9800E-01	1.2671E+01	2.3714E-00	2.8945E-00	9.9800E-00	0.00493958
	IHSSA	9.9800E-01	1.2671E+01	1.7788E-00	2.6911E-00	9.9800E-01	0.00593592
F9	GWO	9.9800E-01	1.0763E+01	2.8342E-00	2.8169E-00	2.4871E-00	0.00345846
	LSO	9.9800E-01	2.6167E-00	1.0852E-00	3.3568E-01	9.9801E-01	0.00358843
	PSO	9.9800E-01	1.9920E-00	1.1637E-00	3.7045E-01	9.9800E-01	0.00318708
	SSA	9.9800E-01	1.2671E+01	2.5646E-00	3.2751E-00	9.9800E-01	0.00464793
	WOA	9.9800E-01	5.9288E-00	1.4931E-00	1.1049E-00	9.9800E-01	0.00335750
	VELSO	3.1315E-04	4.8113E-04	3.6026E-04	3.9691E-05	3.4978E-04	0.00420776
	ILSO	3.0700E-04	1.2230E-03	3.7900E-04	2.2700E-04	3.0900E-04	0.03340013
	LLSSA	3.0700E-04	2.0363E-02	1.3730E-03	3.5380E-03	7.5500E-04	0.00121681
	IHSSA	3.0749E-04	3.2621E-04	3.0861E-04	3.4182E-06	3.0761E-04	0.00199442
F10	GWO	3.0800E-04	2.0363E-02	2.4100E-03	5.9890E-03	3.5600E-04	0.00088469
	LSO	3.0900E-04	2.2260E-03	7.0800E-04	4.9600E-04	5.2500E-04	0.00230245
	PSO	3.0700E-04	1.5940E-03	7.9200E-04	2.9800E-04	7.9900E-04	0.00038637
	SSA	3.0749E-04	4.5697E-04	3.1261E-04	2.6809E-05	3.0751E-04	0.00095074
	WOA	3.0800E-04	1.5250E-03	5.5400E-04	2.7500E-04	5.3200E-04	0.00031954
	VELSO	-10.536389	-10.536083	-10.536270	5.6974E-05	-10.536280	0.00345656
	ILSO	-10.536410	-10.536355	-10.536497	1.7978E-05	-10.536409	0.03475976
	LLSSA	-10.536410	-10.536294	-10.536406	2.0851E-05	-10.536409	0.00136637
	IHSSA	-10.536410	-10.536409	-10.536410	3.3233E-15	-10.536409	0.00224793
F11	GWO	-10.536100	-2.4214100	-10.260900	1.4558E-00	-10.531500	0.00062822
	LSO	-10.532300	-10.532300	-10.535200	1.0730E-03	-10.532300	0.00079949
	PSO	-10.536400	-2.4217300	-7.8214600	3.5993E-00	-10.536400	0.00050559
	SSA	-10.536400	-5.1284800	-9.9956200	1.6224E-00	-10.536400	0.00121564
	WOA	-10.536300	-1.6761200	-7.2180200	3.4283E-00	-7.7915100	0.00054864
	VELSO	0.000E-000	1.8066E-01	2.0929E-02	3.4979E-02	9.6520E-03	0.00481171
	ILSO	3.8200E-04	1.6081E-02	2.2050E-03	3.3880E-03	6.0800E-04	0.03760746
	LLSSA	2.6144E+03	4.6948E+03	3.6125E+03	5.1075E+02	3.5587E+03	0.00129872
	IHSSA	5.7835E-01	5.1741E+03	2.9555E+03	1.6339E+03	3.2566E+03	0.00203382
F12	GWO	5.0453E+03	9.6698E+03	6.4993E+03	1.2280E+03	6.4449E+03	0.00099434
	LSO	4.7760E+02	7.1165E+03	3.3238E+03	1.9457E+03	3.5065E+03	0.00237659
	PSO	6.1298E+03	9.4350E+03	7.6786E+03	9.3676E+02	7.6343E+03	0.00053665
	SSA	4.6287E+03	6.1429E+03	3.4695E+03	5.6131E+02	4.5917E+03	0.00099749
	WOA	5.2714E-01	4.2721E+03	1.5015E+03	1.4133E+03	1.0153E+03	0.00040349
	VELSO	0.000E-000	2.263E-119	7.548E-121	4.062E-120	3.824E-163	0.00252789
	ILSO	1.6982E+14	5.7002E+16	5.8217E+15	1.2490E+16	1.5165E+15	0.03880041
	LLSSA	9.5368E+01	4.4274E+02	3.2083E+02	7.6339E+01	3.2119E+02	0.00099779
	IHSSA	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.00190900
F13	GWO	4.4329E+14	6.0789E+16	6.2318E+15	1.1759E+16	2.1482E+15	0.00042479
	LSO	1.5411E-00	1.1824E+28	4.9797E+26	2.1316E+27	3.7148E+20	0.00088948
	PSO	4.0693E+42	6.7472E+43	2.3198E+43	1.5014E+43	1.9576E+43	0.00020546
	SSA	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.00076625
	WOA	5.7364E+05	2.7894E+17	9.5464E+16	1.1473E+17	2.3274E+16	0.00023749
	VELSO	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.00179158
	ILSO	3.666E-122	1.933E-110	6.456E-112	3.469E-111	1.000E-116	0.03944015
	LLSSA	0.000E-000	1.028E-212	3.432E-214	0.000E-000	0.000E-000	0.00098611
	IHSSA	0.000E-000	6.417E-304	2.757E-305	0.000E-000	0.000E-000	0.00160812
F14	GWO	8.336E-172	1.206E-121	4.021E-123	2.202E-122	1.186E-145	0.00025933
	LSO	9.0876E-111	1.233E-102	4.5730E-104	2.2097E-103	1.220E-106	0.00038196
	PSO	7.4090E-27	5.7317E-21	4.4715E-22	1.0563E-21	9.8761E-23	0.00014153
	SSA	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.00075649
	WOA	6.0651E-76	3.5434E-41	1.1811E-42	6.3606E-42	6.0496E-59	0.00025875
	VELSO	0.000E-000	3.247E-182	1.562E-183	0.000E-000	8.731E-190	0.00434353
	ILSO	7.4847E-15	1.7359E-10	1.2055E-11	3.4277E-11	5.4992E-13	0.04239442
	LLSSA	1.4709E-07	1.6600E-00	5.8705E-02	2.9753E-01	4.4817E-05	0.00131754
	IHSSA	5.4837E-11	6.0406E-06	3.2685E-07	1.0765E-06	1.6501E-08	0.00191386
F15	GWO	1.9947E-05	2.0686E-04	6.6807E-05	3.7094E-05	6.1301E-05	0.00108017
	LSO	3.1441E-47	9.4302E-42	1.1003E-42	2.0877E-42	1.2861E-43	0.00244754
	PSO	3.2274E-02	5.7363E-01	1.7745E-01	1.3385E-01	1.3634E-01	0.00057112
	SSA	0.000E-000	8.231E-237	2.744E-238	0.000E-000	0.000E-000	0.00108585
	WOA	1.4078E-39	4.0986E-06	3.8366E-07	9.5439E-07	2.3194E-12	0.00050619

TABLE 3. *(Continued.)* **Optimization results of different algorithms.**

F16	VELSO	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.00265359
	ILSO	8.8183E+06	2.6894E+11	2.0613E+10	5.0957E+10	1.9809E+09	0.04357680
	LLSSA	0.000E-000	3.304E-168	1.101E-169	0.000E-000	0.000E-000	0.00108505
	IHSSA	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.00162931
	GWO	1.1162E-15	3.3534E-11	4.0663E-12	6.7315E-12	6.6473E-13	0.00047689
	LSO	2.2562E+21	1.5712E+24	2.5894E+23	3.6025E+23	1.0435E+23	0.00092698
	PSO	3.9159E+25	1.4906E+26	1.0195E+26	2.5920E+25	1.0131E+26	0.00025148
	SSA	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.000E-000	0.00085274
	WOA	5.7876E-24	3.5751E-16	1.3171E-17	6.4054E-17	1.8866E-20	0.00027147

Except for GWO and PSO on F_7, the rest of the algorithms have the same optimal value as VELSO. Except for SSA, GWO, and PSO on F_8, the rest of the algorithms have the same optimal value as the VELSO algorithm. There is no obvious difference in F_9, and the position of the optimal value can be found. The algorithm in this paper has obvious advantages in F_{12}. In F_{13}, IHSSA and SSA have the same optimal values as VLESO. In F_{15}, only SSA has the same optimal value as our algorithm. In F_{10} VELSO has no obvious advantage, and in F_{11} VELSO is only slightly better than GWO, LSO, and WOA. In terms of stability, the VELSO algorithm has no order of magnitude change between the average value and the optimal value on F_1, F_3, F_6-F_{11}, F_{14}, and F_{16}, which shows that the algorithm in this paper has strong stability under these test functions. Although the average value does not keep the order of magnitude unchanged from the optimal value under the other test functions, it can be seen from the values of their standard deviation and median that the VELSO algorithm still maintains strong stability. The VELSO algorithm is slower in running time, which is due to the increase in the running time due to increase of the strategy, but it still has an advantage over the running time of the ILSO algorithm. Combining the above content, we can know that among the 16 test functions, the algorithm in this paper has strong search performance and stability. Although the increase of the strategy affects the running time, the time consumption of the relative performance improvement is understandable.

Fig. 7 shows the convergence speed of each algorithm in different test functions, it can be seen from the figure that the VELSO algorithm has excellent convergence speed in F_1-F_5, F_8, F_{13}-$_{14}$, and F_{16}, and they all have good convergence accuracy. Although F_7 and F_{14} also have the advantage of convergence speed, the effect is not obvious, but the convergence accuracy is still very good in the later stage of iteration. In F_6, F_9-$_{10}$, F_{12}, and F_{15}, although the convergence speed is slightly inferior, the optimization accuracy is also ensured in the later stage of iteration. Therefore, the VELSO algorithm has great advantages in both convergence speed and convergence accuracy.

In order to more intuitively reflect the advantages of the algorithm in this paper, the Wilcoxon statistical test proposed by Derrac et al. is used to verify whether the algorithm has a significant improvement, and the significance level is 5%. The test principle is as follows: when $P < 0.05$, there is a significant difference between the two algorithms. When $P = 0.05$, the performance of the algorithm is considered to be comparable. When $P > 0.05$, the performance of the algorithm in this paper is considered to be poor. In Table 4, the P-values of VELSO's Wilcoxon rank-sum test and other algorithms are shown among the 16 test functions. As shown in Table 4, $P < 0.05$ is the focus of attention. From Table 4, it can be seen that the algorithm in this paper has significant advantages over other algorithms in functions F_1-F_6, F_{12}-F_{13}, and F_{15}. On other functions, the advantages of the algorithm in this paper are not obvious only for individual algorithms, so we can verify from Table 4 that the VELSO algorithm has better performance and higher convergence accuracy.

Combining the above content, we can know that under these 16 test functions, the algorithm in this paper has great advantages over other algorithms in terms of stability, optimization accuracy, or running time, which also proves that the algorithm proposed in this paper is meaningful.

VII. APPLICATION

A. THEORETICAL ANALYSIS

DV-Hop localization algorithm [39] is a ranging-free localization algorithm. It relies on the information interaction between nodes to locate unknown nodes. With the characteristics of simple implementation and low hardware requirements, it has been widely used in wireless sensor networks (WSN) [40]. There are three steps to realize the DV-Hop localization algorithm:

The anchor node broadcasts its location information to the nodes within the communication range. The receiving node records the minimum number of hops to the anchor node, ignores the remaining hop information, and the receiving node forwards the hop information by adding 1.

According to the recorded information, the anchor node calculates the average jump distance through formula (26)

$$Hopsize_i = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i} hop_{ij}} \tag{26}$$

In formula (26), *Hopsize* represents the average hop distance of anchor nodes, *hop* represents the minimum hop number between anchor nodes, and (x, y) represents the location of anchor nodes.

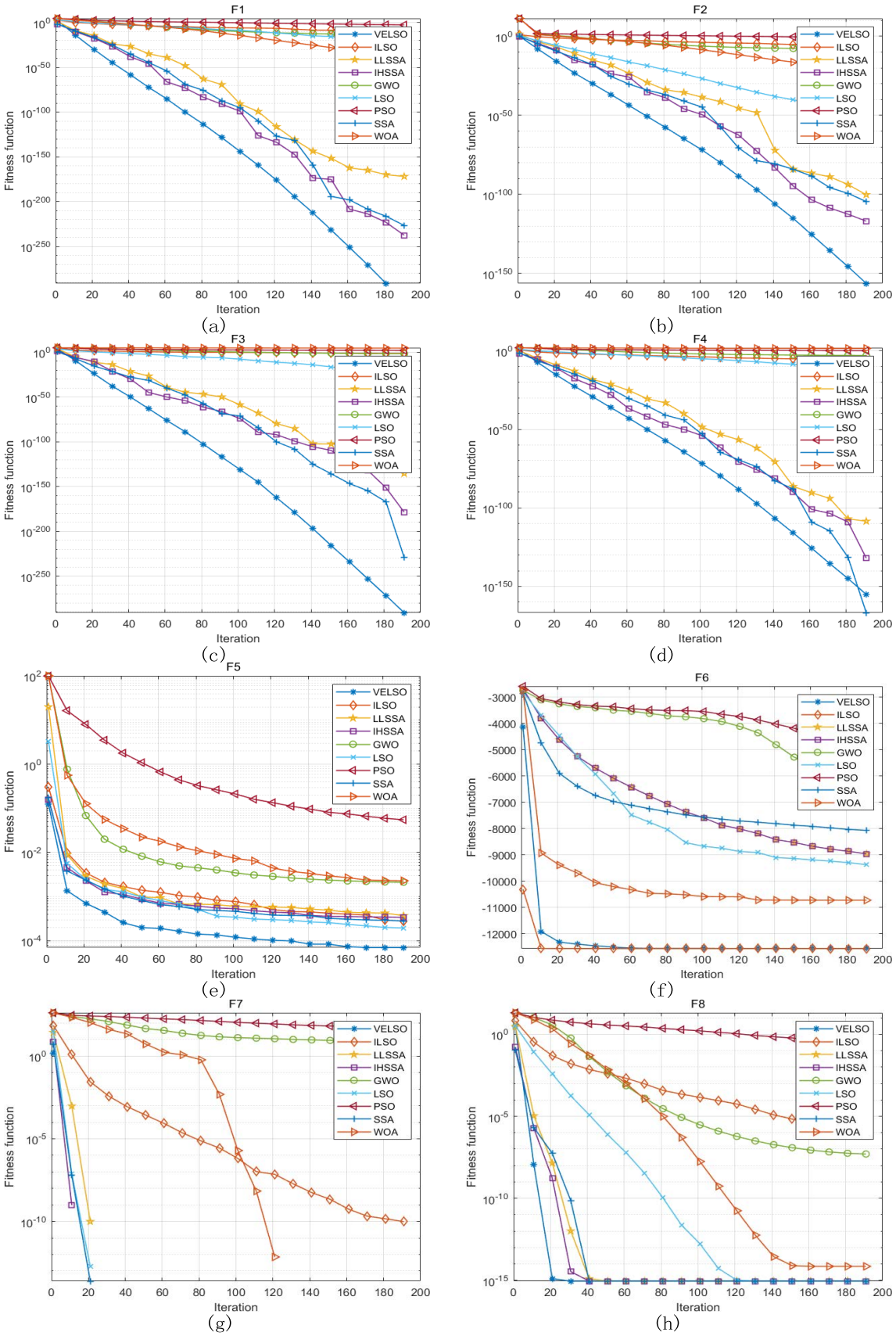

FIGURE 7. Curve convergence diagrams of 9 algorithms under 16 test functions, including F_1 (a), F_2 (b), F_3 (c), F_4 (d), F_5 (e), F_6 (f), F_7 (g), F_8 (h), F_9 (i), F_{10} (j), F_{11} (k), F_{12} (l), F_{13} (m), F_{14} (n), F_{15} (o), F_n16 (p).

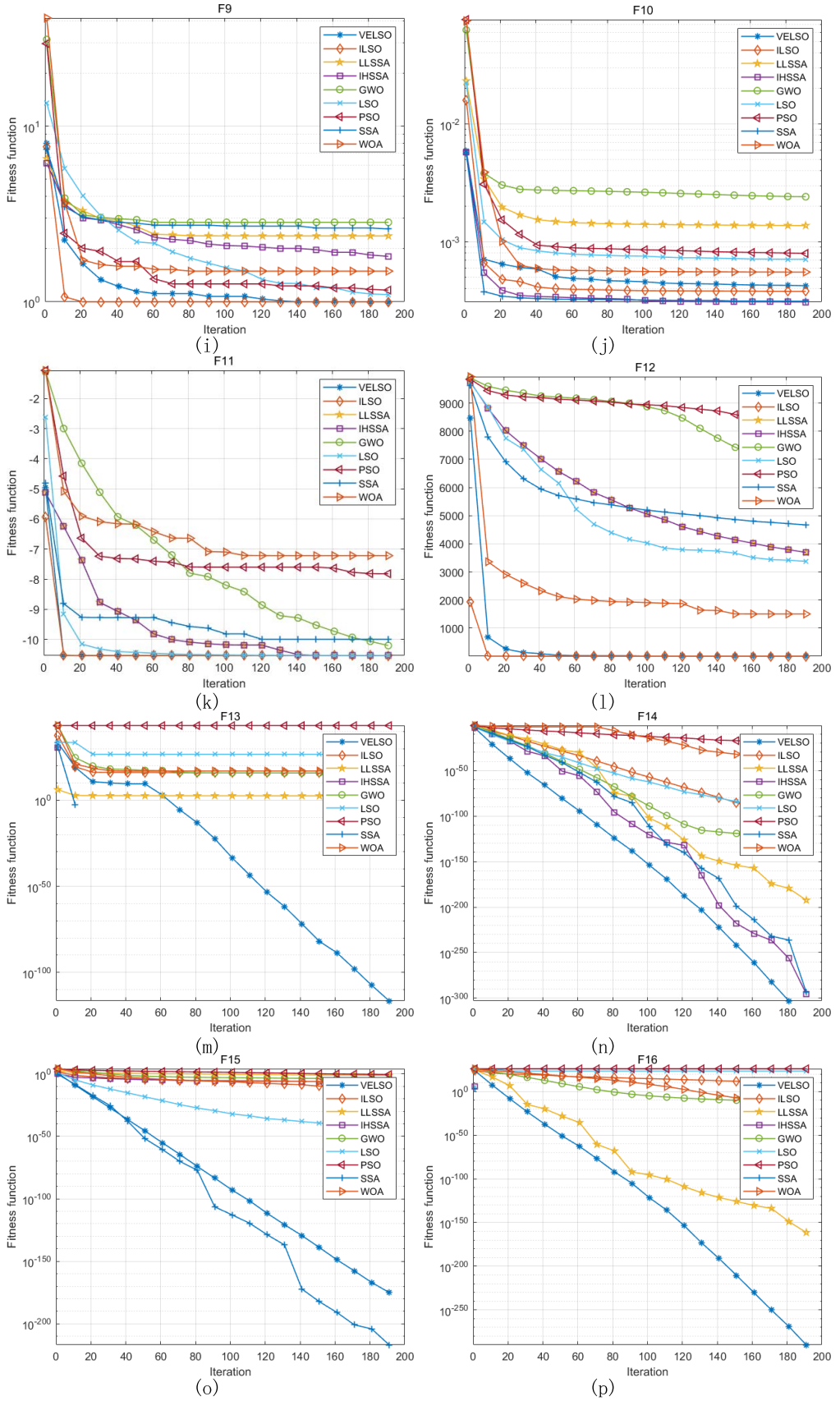

FIGURE 7. *(Continued.)* **Curve convergence diagrams of 9 algorithms under 16 test functions, including F_1 (a), F_2 (b), F_3 (c), F_4 (d), F_5 (e), F_6 (f), F_7 (g), F_8 (h), F_9 (i), F_{10} (j), F_{11} (k), F_{12} (l), F_{13} (m), F_{14} (n), F_{15} (o), F_n16 (p).**

TABLE 4. P value and Wilcoxon rank.

Function	GWO	LSO	PSO	SSA	WOA	ILSO	LLSSA	IHSSA
F1	1.21E-12	1.21E-12	1.21E-12	0.04190	1.21E-12	1.21E-12	6.61E-05	0.0110
F2	3.02E-11	3.02E-11	3.02E-11	3.88E-07	3.02E-11	3.02E-11	3.02E-11	3.01E-11
F3	1.21E-12	1.21E-12	1.21E-12	0.0110	1.21E-12	1.21E-12	3.45E-07	1.46E-04
F4	3.02E-11	3.02E-11	3.02E-11	3.32E-10	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F5	3.02E-11	4.11E-07	3.02E-11	1.29E-06	3.47E-10	8.48E-09	6.53E-08	9.26E-09
F6	3.02E-11	3.02E-12	3.02E-12	3.02E-12	8.99E-11	3.96E-08	3.02E-11	4.08E-11
F7	1.21E-12	N/A	1.21e-12	N/A	N/A	1.66E-11	N/A	N/A
F8	1.21E-12	N/A	1.21E-12	N/A	9.41E-12	1.21E-12	N/A	N/A
F9	N/A	2.82E-04	1.45E-04	N/A	0.0274	4.56E-10	N/A	4.06E-05
F10	N/A	2.41E-05	6.47E-08	1.04E-08	0.0176	4.99E-05	2.90E-06	1.67E-09
F11	5.47E-11	1.94E-08	N/A	8.99E-08	1.36E-08	1.82-10	7.33E-11	1.78E-11
F12	4.46E-11	4.46E-11	4.46E-11	4.46E-11	4.46E-11	1.72E-05	4.46E-11	4.46E-11
F13	4.13E-11	4.13E-11	4.13E-11	7.27E-08	4.13E-11	4.13E-11	4.13E-11	7.27E-08
F14	2.26E-12	2.26E-12	2.26E-12	N/A	2.26E-12	2.26E-12	8.03E-04	N/A
F15	4.46E-11	4.46E-11	4.46E-11	3.81E-11	4.46E-11	4.46E-11	4.46E-11	4.46E-11
F16	2.26E-12	2.26E-12	2.26E-12	N/A	2.26E-12	2.26E-12	0.0124	N/A

The anchor node forwards the calculated average hop distance to the network, the unknown node saves the average hop distance information of the nearest anchor node, and then obtains the distance from the unknown node to different anchor nodes by using the formula (27). Finally, the location of the unknown node is obtained by the least square method

$$d_{ij} = Hopsize \cdot hop_{ij} \tag{27}$$

In formula (27), d represents the distance from the unknown node to the anchor node. *hop* represents the minimum number of hops to the anchor node saved by the unknown node, and *Hopsize* is the average hop distance information saved by the unknown node.

In recent years, many scholars have studied the DV-Hop algorithm mainly from three aspects:

1) Optimize hop count information

(2) Corrected average jump distance

(3) Using an intelligent algorithm instead of the least square method to solve the location information of unknown nodes

(1), (2) is to reduce ranging error, and (3) is to reduce positioning error. For the DV-Hop algorithm, reducing ranging error is the basis and premise of reducing positioning error. Therefore, in order to prove the practicability of the algorithm in this paper, reducing the ranging error between node positioning from two aspects (1) and (2) are prepared first. Then, we use the VELSO algorithm to locate the unknown node. And prove the availability of the VELSO algorithm through comparative experiments.

In order to reduce the error of the DV-Hop algorithm in the minimum number of hops, we introduce a multi-hop division of nodes from the document [41] to refine the number of hops from nodes within the communication radius to communication nodes. At the same time, in order to prevent the division of the multi-communication radius from being too cumbersome, we divide the level of node communication radius according to the formula (28).

$$R = \frac{1}{n}r, \frac{2}{n}r, \frac{3}{n}r, \cdots, r \tag{28}$$

In formula (28), R is the pre-propagation radius of multi-radius communication, n is the reciprocal rounding of the proportion of anchor nodes, and r is the maximum communication radius.

In the stage of calculating the average jump distance of anchor nodes, the minimum error means square criterion in literature [42], [43] is introduced to solve the average jump distance of anchor nodes. The formula is as follows:

$$Hopsize_i = \frac{\sum_{i=1,j\neq i}^{n} d_{ij} \cdot hop_{ij}}{\sum_{i=1,j\neq i}^{n} hop_{ij}^2} \tag{29}$$

In formula (29), *Hopsize* represents the average hop distance of anchor nodes, d represents the distance between different anchor nodes, and n is the number of anchor nodes. *hop* represents the minimum number of hops between different anchor nodes.

The average hop distance of the unknown node of the traditional DV-Hop algorithm is obtained from the anchor node closest to the unknown node, which results in different unknown nodes obtaining the same average hop distance, and affects the ranging accuracy. Therefore, a correction factor is introduced from the literature [44], [45] to correct the average hop distance of the unknown node. The specific formula is as follows:

$$\omega_i = \frac{1/hop_{ui}}{\sum_{i=1}^{n} 1/hop_{ui}} \tag{30}$$

In formula (30), *hop* represents the number of hops from the unknown node to the anchor node, ω represents the correction factor for the average jump distance of an unknown node, which is set based on the distance from the unknown node to the anchor node. Then, the average hop distance of unknown nodes is solved according to the formula (31).

$$Hopsize_u = \sum_{i=1}^{n} \omega_i \cdot Hopsize_i \tag{31}$$

After obtaining the average hop distance of the unknown node, the distance from the unknown node to the anchor node can be obtained according to the formula (27).

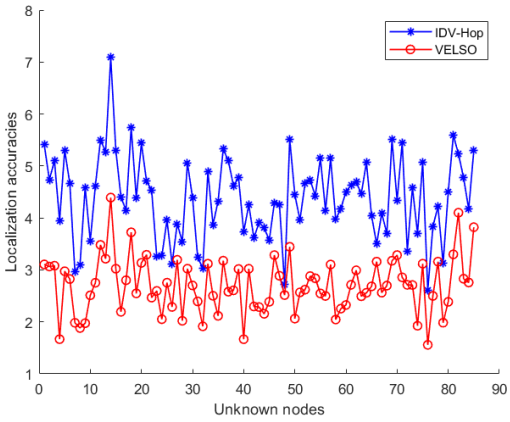

FIGURE 8. Comparison diagram of ranging error.

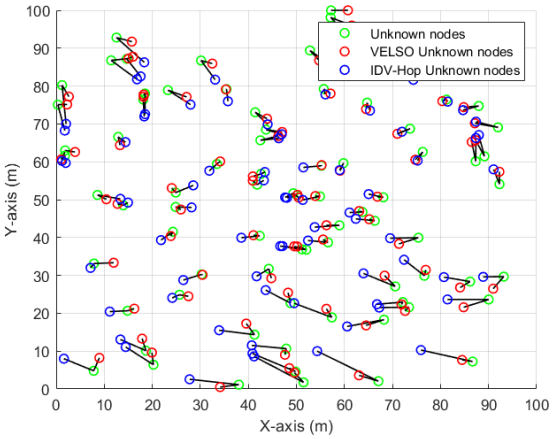

FIGURE 9. Positioning point diagram.

According to references [42], [46], when using the VELSO algorithm to locate unknown nodes, we need to transform the node location problem into a function optimization problem. Therefore, the fitness function is shown in the formula (32).

$$f(x, y) = \sum_{i=1}^{m} (\sqrt{(x_i - x)^2 + (y_i - y)^2} - d_i)^2 \tag{32}$$

B. EXPERIMENT

In order to verify the practicability and effectiveness of this algorithm, this algorithm is integrated with the above improved DV-Hop algorithm (IDV-Hop). In MATLAB 2020a, set the total number of nodes to 100, the number of anchor nodes to 20, and the maximum communication radius to 20m, and run for 30 times. Firstly, according to formula (33), the broken line diagram of the ranging error between the VELSO algorithm and improved IDV-Hop is obtained, as shown in Figure 8, and the positioning point diagram of nodes is shown in Figure 9.

$$La = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{33}$$

where, (x, y) is the real position of the unknown node and (x_i, y_i) is the positioning position of the unknown node.

It can be seen from Fig. 8 and Fig. 9 that replacing the least square method with the VELSO algorithm effectively reduces the ranging error accumulated by the least square method. Therefore, in order to better show the feasibility of the VELSO algorithm, we test the average positioning error of the VELSO algorithm. The calculation formula for average positioning error is shown in (34).

$$Error = \frac{\sum_{i=1}^{U} \sqrt{(x - x_i)^2 + (y - y_i)^2}}{U \cdot R} \tag{34}$$

In formula (34), U is the number of unknown nodes, R is the communication radius of the node, (x, y) is the actual position of the unknown node, and (x_i, y_i) is the positioning position of the unknown node.

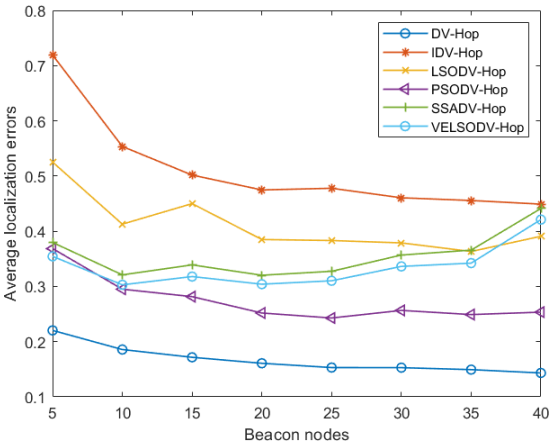

FIGURE 10. Anchor node and average positioning error.

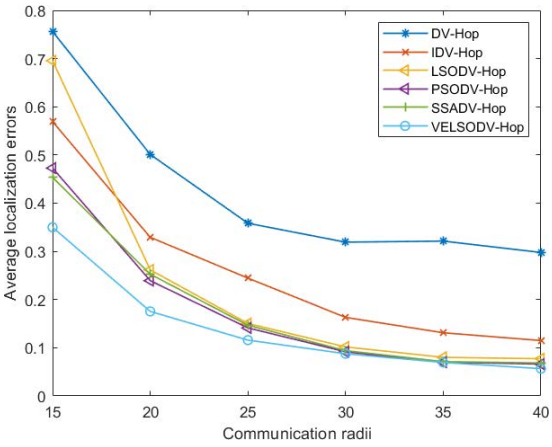

FIGURE 11. Communication radius and average positioning error.

Figure 10 shows the relationship between the average positioning error and the number of anchor nodes when the communication radius is 20 and the total number of nodes is 100. Figure 11 shows the relationship between the communication radius and the average positioning error when the ratio of anchor nodes is 0.2 and the total number of nodes

TABLE 5. Average positioning error under different positioning algorithms.

Variable	Algorithm	Average localization errors
Beacon nodes	DV-Hop	0.5116
	IDV-Hop	0.4112
	LSODV-Hop	0.2749
	PSODV-Hop	0.3564
	SSADV-Hop	0.3362
	VELSODV-Hop	0.1671
Communication radii	DV-Hop	0.4257
	IDV-Hop	0.2587
	LSODV-Hop	0.2278
	PSODV-Hop	0.1811
	SSADV-Hop	0.1802
	VELSODV-Hop	0.1425
Number nodes	DV-Hop	0.3688
	IDV-Hop	0.2891
	LSODV-Hop	0.2518
	PSODV-Hop	0.2436
	SSADV-Hop	0.2127
	VELSODV-Hop	0.1199

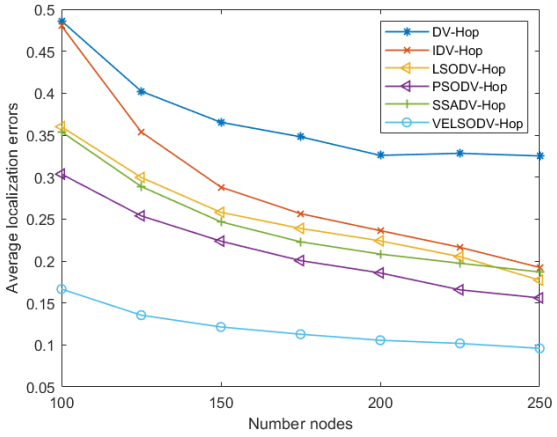

FIGURE 12. Total number of nodes and average positioning error.

is 100. Figure 12 shows the relationship between the total number of nodes and the average positioning error when the ratio of anchor nodes is 0.2, the communication radius is 20. Table 5 shows the average positioning error of each algorithm under different conditions.

As can be seen from Figure 10-12 and Table 5, when the anchor node is a variable, the communication radius is a variable and the total number of nodes is a variable, the average positioning error of the VELSODV-Hop algorithm is 0.1671, 0.1425, and 0.1191. It can be seen that the VELSO algorithm can effectively reduce the positioning error and improve the positioning accuracy compared with other algorithms. However, due to the mechanism of the DV-Hop location algorithm, it cannot reduce the location error to 0, so it can only try to reduce the location error to an acceptable range. Therefore, it is proved that this algorithm is feasible and effective in the DV-Hop locations.

VIII. SUMMARIZE

Aiming at the defects of the lion swarm algorithm, a variable speed elastic collision lion swarm algorithm is proposed in this paper. Firstly, the strategy based on the variable spiral search is used to improve the interaction mode of lioness hunting and increase the flexibility of the global search of the algorithm. Then the improved refraction learning strategy is used to increase the exploration ability and diversity of the population. Furthermore, the variable speed elastic collision strategy is used to increase the probability of the population jumping out of the local optimum, improve the convergence accuracy while maintaining the convergence speed of the lion swarm algorithm, avoid its falling into the local optimum, and improve the efficiency and accuracy of the algorithm search. Under the comparison of 16 test functions, it is verified that the VELSO algorithm is generally superior to the classic algorithms such as PSO, GWO, WOA, SSA, and LSO. It also has more advantages than several algorithms such as the SSA variant and the LSO variant. Through the analysis of the effectiveness of the algorithm, it is verified that the algorithm not only ensures the convergence speed but also improves the convergence accuracy and increases the search range. After calculation, it is proved that compared with the improvement of the effect, a little increase in time complexity is worth to be done. Finally, the application of the VELSO algorithm to the DV-Hop location proves the feasibility and practicability of the VELSO algorithm, which is better than other algorithms.

Due to the limitations of the algorithm in this paper, the effect is not ideal when faced with multi-objective testing

problems, so the research on the VELSO algorithm is far from over. In the future research process, on the basis of this algorithm, we will focus on the problem of solving multi-objective functions with the algorithm in this paper, and at the same time study the behavior pattern of the lion cubs in the lion group algorithm to obtain better accuracy and convergence efficiency and continue to study how to apply this algorithm to the localization of complex terrain. in order to obtain a good positioning effect.

DATA AVAILABILITY

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

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] S. Liu, Y. Yang, and Y. Zhou. "A swarm intelligence algorithm-lion swarm optimization." *Pattern Recognit. Artif. Intell.*, vol. 31, no. 5, pp. 431–441, 2018.
[2] C. Ouyang, D. Zhu, and F. Wang, "A learning sparrow search algorithm," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–23, Aug. 2021, doi: 10.1155/2021/3946958.
[3] G.-C. Liao, "Fusion of improved sparrow search algorithm and long short-term memory neural network application in load forecasting," *Energies*, vol. 15, no. 1, p. 130, Dec. 2021.
[4] Y. Lin, L. Zhao, X. Liu, W. Yang, X. Hao, and L. Tian, "Design optimization of a passive building with green roof through machine learning and group intelligent algorithm," *Buildings*, vol. 11, no. 5, p. 192, May 2021.
[5] Z. Han, X. Tian, X. Dong, and F. Xie, "Swarm intelligent algorithm for re-entrant hybrid flow shop scheduling problems," *Int. J. Simul. Process Model.*, vol. 14, no. 1, pp. 17–27, 2019.
[6] X.-W. Yu, L.-P. Huang, Y. Liu, K. Zhang, P. Li, and Y. Li, "WSN node location based on beetle antennae search to improve the gray wolf algorithm," *Wireless Netw.*, vol. 28, no. 2, pp. 539–549, Feb. 2022.
[7] A. M. Anter, M. A. Elaziz, and Z. Zhang, "Real-time epileptic seizure recognition using Bayesian genetic whale optimizer and adaptive machine learning," *Future Gener. Comput. Syst.*, vol. 127, pp. 426–434, Feb. 2022.
[8] A. M. Anter, D. Oliva, A. Thakare, and Z. Zhang, "AFCM-LSMA: New intelligent model based on Lévy slime mould algorithm and adaptive fuzzy C-means for identification of COVID-19 infection from chest X-ray images," *Adv. Eng. Inform.*, vol. 49, Aug. 2021, Art. no. 101317.
[9] A. M. Anter, D. Gupta, and O. Castillo, "A novel parameter estimation in dynamic model via fuzzy swarm intelligence and chaos theory for faults in wastewater treatment plant," *Soft Comput.*, vol. 24, no. 1, pp. 111–129, Jan. 2020.
[10] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[11] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
[12] J. Xue, "Research and application of a novel swarm intelligence optimization technique: Sparrow search algorithm," M.S. thesis, Dept. Control Sci. Eng., Donghua Univ., Shanghai, China, 2020.
[13] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, 1995, pp. 1942–1948.
[14] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci. (MHS)*, 1995, pp. 39–43.
[15] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, "Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm," *Int. J. Mach. Learn. Cybern.*, vol. 11, no. 7, pp. 1501–1529, Jul. 2020.
[16] M. Li, G. Xu, Q. Lai, and J. Chen, "A chaotic strategy-based quadratic opposition-based learning adaptive variable-speed whale optimization algorithm," *Math. Comput. Simul.*, vol. 193, pp. 71–99, Mar. 2022.
[17] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009.
[18] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, 2016.
[19] H. A. Shehadeh, "A hybrid sperm swarm optimization and gravitational search algorithm (HSSOGSA) for global optimization," *Neural Comput. Appl.*, vol. 33, no. 18, pp. 11739–11752, Sep. 2021.
[20] F. Ji and M. Jiang, "Three-dimensional DV-hop localization based on improved lion swarm optimization algorithm," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Aug. 2020, pp. 40–45.
[21] Z. Yang and C. Wei, "Prediction of equipment performance index based on improved chaotic lion swarm optimization–LSTM," *Soft Comput.*, vol. 24, no. 13, pp. 9441–9465, Jul. 2020.
[22] Z. Dai, M. Jiang, X. Li, D. Yuan, and X. Zhou, "Reinforcement lion swarm optimization algorithm for tool wear prediction," in *Proc. Global Rel. Prognostics Health Manag. (PHM-Nanjing)*, Oct. 2021, pp. 1–7.
[23] D. Zhu, Y. Ma, M. Wang, J. Yang, Y. Yin, and S. Liu, "LSO-FastSLAM: A new algorithm to improve the accuracy of localization and mapping for rescue robots," *Sensors*, vol. 22, no. 3, p. 1297, Feb. 2022.
[24] Z. Wu and Z. Xie, "A multi-objective lion swarm optimization based on multi-agent," *J. Ind. Manag. Optim.*, vol. 19, no. 2, p. 1447, 2023.
[25] C. Huang, D. Yuan, H. Zhang, and A. Zheng, "Job shop scheduling in discrete manufacturing based on improved hybrid lion swarm optimization," in *Proc. IEEE 16th Conf. Ind. Electron. Appl. (ICIEA)*, Aug. 2021, pp. 438–443.
[26] Z. Daoqing and J. Mingyan, "Parallel discrete lion swarm optimization algorithm for solving traveling salesman problem," *J. Syst. Eng. Electron.*, vol. 31, no. 4, pp. 751–760, Aug. 2020.
[27] W. Qiao, H. Lu, G. Zhou, M. Azimi, Q. Yang, and W. Tian, "A hybrid algorithm for carbon dioxide emissions forecasting based on improved lion swarm optimizer," *J. Cleaner Prod.*, vol. 244, Jan. 2020, Art. no. 118612.
[28] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," in *Proc. Int. Conf. Comput. Intell. Modelling, Control Autom. Int. Conf. Intell. Agents, Web Technol. Internet Commerce (CIMCA-IAWTIC)*, vol. 1, 2005, pp. 695–701.
[29] S. Gupta and K. Deep, "A hybrid self-adaptive sine cosine algorithm with opposition based learning," *Expert Syst. Appl.*, vol. 119, pp. 210–230, Apr. 2019.
[30] C. Ouyang, D. Zhu, and Y. Qiu, "Lens learning sparrow search algorithm," *Math. Problems Eng.*, vol. 2021, pp. 1–17, May 2021.
[31] Z. Wang, X. Huang, and D. Zhu, "A multistrategy-integrated learning sparrow search algorithm and optimization of engineering problems," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–21, Feb. 2022.
[32] D. Oliva and M. A. Elaziz, "An improved brainstorm optimization using chaotic opposite-based learning with disruption operator for global optimization and feature selection," *Soft Comput.*, vol. 24, no. 18, pp. 14051–14072, 2020.
[33] Y. Sun and Y. Gao, "A multi-objective particle swarm optimization algorithm based on Gaussian mutation and an improved learning strategy," *Mathematics*, vol. 7, no. 2, p. 148, Feb. 2019.
[34] M. Guo, J.-S. Wang, L. Zhu, S. Guo, and W. Xie, "Improved ant lion optimizer based on spiral complex path searching patterns," *IEEE Access*, vol. 8, pp. 22094–22126, 2020.
[35] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: An optimization method for continuous non-linear large scale problems," *Inf. Sci.*, vol. 183, no. 1, pp. 1–15, Jan. 2012.
[36] D. Zhu, Z. Huang, L. Xie, and C. Zhou, "Improved particle swarm based on elastic collision for DNA coding optimization design," *IEEE Access*, vol. 10, pp. 63592–63605, 2022.
[37] X. Zhang, Z. Wang, and Z. Lu, "Multi-objective load dispatch for microgrid with electric vehicles using modified gravitational search and particle swarm optimization algorithm," *Appl. Energy*, vol. 306, Jan. 2022, Art. no. 118018.
[38] F. Farivar and M. A. Shoorehdeli, "Stability analysis of particle dynamics in gravitational search optimization algorithm," *Inf. Sci.*, vols. 337–338, pp. 25–43, Apr. 2016.
[39] L. Jian Yin, "A new distance vector-hop localization algorithm based on half-measure weighted centroid," *Mobile Inf. Syst.*, vol. 2019, pp. 1–9, Jan. 2019.
[40] A. Hassan, A. Anter, and M. Kayed, "A robust clustering approach for extending the lifetime of wireless sensor networks in an optimized manner with a novel fitness function," *Sustain. Comput., Informat. Syst.*, vol. 30, Jun. 2021, Art. no. 100482.

[41] X. Li, K. Wang, B. Liu, J. Xiao, and S. Han, "An improved range-free location algorithm for industrial wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2020, no. 1, pp. 1–13, Dec. 2020.
[42] M. G. A. E. Ghafour, S. H. Kamel, and Y. Abouelseoud, "Improved DV-hop based on squirrel search algorithm for localization in wireless sensor networks," *Wireless Netw.*, vol. 27, no. 4, pp. 2743–2759, May 2021.
[43] K. Chen, C. Wang, L. Chen, X. Niu, Y. Zhang, and J. Wan, "Smart safety early warning system of coal mine production based on WSNs," *Saf. Sci.*, vol. 124, Apr. 2020, Art. no. 104609.
[44] T. Chen, L. Sun, Z. Wang, Y. Wang, Z. Zhao, and P. Zhao, "An enhanced nonlinear iterative localization algorithm for DV_Hop with uniform calculation criterion," *Ad Hoc Netw.*, vol. 111, Feb. 2021, Art. no. 102327.
[45] X. Huang, "Multi-node topology location model of smart city based on Internet of Things," *Comput. Commun.*, vol. 152, pp. 282–295, Feb. 2020.
[46] A. Ouyang, Y. Lu, Y. Liu, M. Wu, and X. Peng, "An improved adaptive genetic algorithm based on DV-hop for locating nodes in wireless sensor networks," *Neurocomputing*, vol. 458, pp. 500–510, Oct. 2021.

WANGSHENG FANG was born in Ganzhou, Jiangxi, in 1963. He received the master's degree. He is also a Professor with the School of Information Engineering, Jiangxi University of Science and Technology. He is also mainly engaged in wireless sensor networks, digital watermarking, and gene expression programming.

WEI GUO was born in Fuyang, Anhui, in 1997. He received the master's degree, major in electronic information, engaged in intelligent optimization algorithm, wireless sensor networks, and other related research.

YUJIA LIU was born in Yichun, Jiangxi, in 1997. He is currently a Teacher at the School of Artificial Intelligence, Jiangxi University of Applied Sciences. He mainly engaged in intelligent algorithm and intelligent computing and other research.