

Retraction

Retracted: Three-Dimensional DV-Hop Localization Algorithm Based on Hop Size Correction and Improved Sparrow Search

Wireless Communications and Mobile Computing

Received 12 December 2023; Accepted 12 December 2023; Published 13 December 2023

Copyright © 2023 Wireless Communications and Mobile Computing. 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.

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] P. Gou, Z. Yu, X. Hu, and K. Miao, "Three-Dimensional DV-Hop Localization Algorithm Based on Hop Size Correction and Improved Sparrow Search," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 1540110, 19 pages, 2022.

Research Article

Three-Dimensional DV-Hop Localization Algorithm Based on Hop Size Correction and Improved Sparrow Search

Pingzhang Gou , Zhaoyang Yu , Xinyue Hu , and Kai Miao 

College of Computer Science and Engineering, Northwest Normal University, China

Correspondence should be addressed to Pingzhang Gou; goupz@nwnu.edu.cn

Received 30 December 2021; Revised 30 March 2022; Accepted 13 April 2022; Published 11 May 2022

Academic Editor: Ali Kashif Bashir

Copyright © 2022 Pingzhang Gou 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.

Acquiring precise localization information of sensor nodes is very important in wireless sensor networks. The 3DDV-hop localization algorithm suffers from large localization errors and high energy consumption. In order to improve positioning accuracy and reduce energy consumption, a 3DDV-hop node localization algorithm (3D-HC SSA) based on hop size correction and improved sparrow search optimization is proposed. The algorithm redefines the amendment factor and reduces the cumulative error caused by the hop counts in the traditional algorithm. A maximum distance similar link method based on a similar path search is proposed to find the most similar known node path pair from the target node to the noncoplanar known node link and correct the hop size between multihop counts. The sparrow search algorithm is improved by using the k -means clustering and sine cosine search strategy, which solves the problem that the traditional sparrow algorithm is easy to fall into the local optimum, accelerates the convergence speed, corrects the position deviation of the target node, and improves the positioning accuracy. Experiments demonstrate that the 3D-HC SSA algorithm can improve positioning accuracy and reduce energy consumption. Compared with the 3DDV-hop algorithm, 3D-GAIDV-hop algorithm, and HCLSO-3D algorithm, the 3D-HC SSA positioning accuracy is significantly improved.

1. Introduction

Wireless sensor networks (WSNs) is a multihop counts wireless network that incorporates nodes to sense, collect, and process target node information, enabling access to objective physical information, extending information acquisition capabilities, and providing the most direct and accurate information in next-generation wireless networks [1]. In next-generation technology, Non-orthogonal Multiple Access (NOMA) allocates different power to known nodes (a node whose own position is known) and target nodes which have different residual energy in WSNs, saving the known node energy consumption problem by using the Successive Interference Cancellation (SIC) technique. Therefore, this technique can be extended to be used in next-generation networks for larger-scale scenarios [2]. Localization information is one of the basic elements of sensor network monitoring data and in many cases is the basis of the application. Monitoring data without localization information is often worthless, for example, monitoring events such

as forest fires and the presence of enemy vehicles on the battlefield, all of which need to determine the specific localization of the event, to achieve the localization, and to track the monitoring target. Therefore, the estimation of target node location is a key issue in wireless sensor networks [3]. It is a challenging work about how to design a localization algorithm with low hardware requirements and low communication and computing overhead. Researchers have proposed some WSNs localization algorithms, which can be broadly classified into two categories: range-based and range-free algorithms [4]. The range-based algorithm calculates the localization of target nodes by measuring the distance or angle information between nodes. Although the effect is more accurate, it requires complex and expensive hardware equipment [5]. The main algorithms based on the range-based approach are the Time of Arrival (TOA) method [6, 7], Time Difference of Arrival (TDOA) method [8], and Angle of Arrival (AOA) method [9]. The localization method based on the range-free approach does not need distance and angle information but only carries out

node localization according to some information such as network connectivity. Without additional complex equipment, the positioning accuracy is lower than that of the algorithm based on the range-based method. The typical localization methods based on the range-free algorithm include the centroid algorithm [10], approximate triangle interior point test algorithm [11, 12], and DV-hop localization algorithm [13].

The DV-hop localization algorithm was proposed by Drağoş et al. [14]. It is a localization method of WSNs that directly measures the distance among nodes without additional hardware equipment [15]. In terms of implementation, the DV-hop localization algorithm only relies on the connectivity of the whole network, so the implementation of the DV-hop localization algorithm is relatively simple and less costly to set up the network. However, cumulative errors occur during the hop size estimation stage, resulting in large errors between the expected and actual distances between the known node and the target node. In the localization stage, there is also a certain error between the actual position and the expected position of the target node. In the DV-hop localization algorithm, a swarm intelligence optimization algorithm is usually used to reduce the localization error of target nodes. Nowadays, swarm intelligence algorithms are commonly used in DV-hop localization algorithms to optimize algorithms to reduce the target node localization error; classical swarm intelligence algorithms such as particle swarm optimization algorithm (PSO) [16], genetic algorithm [17], ant colony optimization algorithm [18], grey wolf optimization algorithm [19], and bacterial foraging optimization algorithm [20]; and some novel optimization algorithms proposed in recent years, such as the sparrow search algorithm [21], the squirrel search algorithm [22], and the butterfly optimization algorithm [23]. Particle swarm optimization is a common algorithm used by researchers to optimize the DV-hop localization algorithm. In Kanwar and Kumar [24] by adjusting a few particle parameters and extending the target node to the selected known node, the objective function of the particle swarm optimization algorithm is established to provide better positioning accuracy in resource-constrained environment through iteration. However, the increase in parameters will lead to the complexity of the operation and increase the complexity of the algorithm. Singh and Sharma [25] first found the feasibility region of each target node and determined the initial position velocity of the particle in the feasibility region. Secondly, update the particles in the estimation of the particle fitness function. Finally, the optimal position of target nodes can be determined after the iteration is completed by setting the number of iterations. Singh and Sharma [26] used the PSO algorithm to further correct the proposed two-dimensional hyperbolic algorithm to determine the localization of target nodes. Firstly, find out the survival area of each target node and determine the initial position and velocity of the particles in the feasible area; secondly, estimate the fitness of each particle; and finally, obtain the optimal solution through iteration, which is the optimal position of the target node. However, this algorithm does not optimize the PSO algorithm, which will result in slow convergence; the algorithm is easy to fall into a local optimum during the solution process; and the algorithm complexity and space complexity

will also increase. In Zhang et al. [27], the PSO algorithm is used to further correct the proposed two-dimensional hyperbolic algorithm to determine the localization of unknown nodes. Firstly, find out the survival region of each unknown node, and determine the initial position and velocity of particles in the feasible region; secondly, the fitness of each particle is estimated; and finally, the optimal solution obtained through iteration is the optimal localization of unknown nodes. The algorithm improves the positioning accuracy but increases the number of operations during the iterative process, which leads to an increase in the complexity of the algorithm. Fang et al. [28] used the exponentially decreasing function to improve the inertia weights because the inertia weight values in the PSO algorithm affect the network optimization speed effect, and the improved PSO algorithm can correct the position of the target node more effectively, thus improving the positioning accuracy. However, the method only optimizes the algorithm itself, but not the global optimization of the algorithm, so the algorithm is prone to problems such as local optimization.

Other swarm intelligence algorithms optimize DV-hop localization algorithms. Cai et al. [29] proposed weight convergence analysis of the DV-hop localization algorithm based on the genetic algorithm (MW-GADV-hop). Since the traditional weight model does not analyze the convergence and ignores the relationship between weight and error, this algorithm establishes a weight model based on error variation and proves the convergence of the model, which improves the positioning accuracy. However, some results in the weight model with error variation do not converge to 1/4 communication radius, so the convergence of the model is insufficient, which will eventually affect the positioning accuracy. Kanwar and Kumar [30] proposed a DV-hop based range-free localization algorithm for WSNs using runner-root optimization (RRADV-hop). The rotary root algorithm is used to determine the fitness function, and the one with the smallest value of the fitness function through iteration is the corrected target node coordinate. The rotation root optimization algorithm is not initialized, leading to slow convergence of the algorithm and affecting the positioning accuracy. Mohanta and Das [31] proposed a class topper optimization-based improved localization algorithm in a wireless sensor network (CTODV-hop). A two-dimensional hyperbolic algorithm is used to calculate the target node localization, and by the class topper optimization approach, the efficiency and reliability of the algorithm are improved, the localization error is reduced, and the complexity of the algorithm is decreased. Jacob et al. [32] proposed a modified search and rescue optimization-based node localization technique in WSNs (MSRODV-hop). By the MSRO-NLT method, the problem of solving the target node coordinates is transformed into the problem of solving the minimum in mathematics; this method reduces the number of iterations of the algorithm. However, the algorithm only optimizes the local optimum and does not solve the problem of fast convergence, and it is difficult to cover all solution sets completely.

We combine the above studies, a 3DDV-hop localization algorithm based on hop size correction and improved sparrow

search is proposed, which further reduces the hop size and localization error and improves the positioning accuracy. The main work of this article is as follows:

- (1) The maximum distance similar link method is proposed. Based on the traditional similar path search algorithm, a pair of known node paths most similar to the link from the target node to a specific noncoplanar known node is found, and the multihop size between the known node and the target node is modified, which reduces the hop size calculation overhead and improves the positioning accuracy of the algorithm
- (2) The K -means clustering algorithm is introduced to solve the problem that the traditional sparrow search algorithm is easy to fall into the local optimum when initializing. In the optimization scrounger position update stage, in order to avoid the poor effect of the scrounger position update stage in finding the optimal, the positive cosine search strategy is introduced to enhance the convergence of the sparrow search algorithm and improve the convergence accuracy. The fitness function is redesigned so that the fitness value can correct the target node position deviation after iteration

The remainder of this article is summarized as follows. The second part summarizes the related work. The third part states the 3DDV-hop localization algorithm, the traditional sparrow search algorithm, and the improved sparrow search algorithm. The fourth part mentions the localization algorithm of 3D-HCSSA. The fifth part analyzes the experimental results. The sixth part draws conclusions.

2. Related Work

The 3DDV-hop algorithm mainly locates nodes by network connectivity and topology structure. It has low requirements on hardware devices and simple calculation, but nodes are distributed in different areas, resulting in different node densities. The average hop size of target nodes depends on one known node in the whole region. When there are too many known nodes in the region, the hop size of nodes will be wasted, and one known node cannot reflect the network environment of WSNs. If the known node is close to the target node and there is an error in the average hop size itself, the error will be directly transmitted to the target node; then, the error accuracy within the whole network decreases.

In the 3DDV-hop localization algorithm, various optimization algorithms and improvement methods can reduce the node localization error. Gou et al. [33] proposed a three-dimensional localization algorithm (HCLSO-3D) based on hop correction and lion swarm optimization in WSNs. The fitness function values of each type of lion are rearranged in ascending order, and the position of the lion with the smallest fitness value is the optimal position of the target node. The position of the lion king with the lowest fitness value was the optimal localization of the target node. However, this article

only improves the fitness function and does not initialize the lion swarm algorithm, which leads to problems such as local optimization and slow convergence speed. Kaushik et al. [34] proposed an improved 3DDV-hop localization algorithm (I3D-DVLAIN) based on neighbouring node information. By adopting a new method of solving the equations, subtract the last distance equation from all the other distance equations and then divide the equations by the maximum distance equation. Therefore, the error propagation is reduced, and the mathematical analysis of error propagation proves that the method can effectively improve the positioning accuracy. Due to the large number of parameters and related equations in the operation of this method, the calculation difficulty and algorithm complexity are increased during the calculation. In Kanwar and Kumar [35] firstly, the localization of the target node is corrected by the hyperbolic method. Then, the optimal target node localization was obtained by iterating through single-objective and multiobjective functions, respectively, which improved the robustness of the algorithm. However, there are many function variables and complex process in the iterative process, which increases the algorithm overhead. Cheng et al. [36] introduced that the ratio of the number of common neighbour nodes belonging to two nodes that are neighbour nodes to each other to the number of all neighbour nodes of the two nodes is used to replace the volume ratio of the area where the corresponding nodes are located, reversely solve the relation between the continuous hop count and the node ratio, and use the parameter-corrected relationship to calculate the continuous hop count, thereby reducing the distance estimation error, improving the positioning accuracy, but also increasing the algorithm complexity. Cai et al. [37] proposed a multiobjective three-dimensional DV-hop localization algorithm (N2-3DDV-hop) based on NSGA-II. Firstly, a multitarget model and NSGA-II were added to analyze the limitations of the traditional single-target localization model. Secondly, a multiobjective localization model is proposed in combination with the NSGA-II algorithm, and the target node coordinates are derived by iterating according to the model. Due to the large number of parameters and related equations in the algorithm, the calculation difficulty and algorithm complexity are increased. Sharma and Kumar [38] proposed using the genetic algorithm to improve the range-free localization of three-dimensional WSNs (3D-GAIDV-hop). Firstly, the genetic algorithm is introduced to accelerate the convergence speed. Secondly, the fitness function is redesigned; the optimal position of the target node coordinates is the one with the smallest adaptation value by iteration. The minimum fitness value is the optimal localization of the target node coordinate, and the positioning accuracy is improved. However, there is the problem that the algorithm is difficult to cover all solution sets.

Based on the above research, this article is mainly aimed at the inaccurate problem of hop count division, hop size setting, and target node coordinate calculation of the 3DDV-hop localization algorithm and proposes a 3DDV-hop localization algorithm based on hop size correction and improved sparrow search. With the HCSSA localization algorithm (3D-DV-hop based on hop size correction and improved sparrow search algorithm), this article divides the optimal hop count by redefining the amendment factor and proposes the maximum

distance similar link method to correct the multihop size between nodes. K -means clustering and sine-cosine search strategy are introduced to improve the traditional sparrow search algorithm to correct the coordinate deviation of target nodes. In conclusion, the 3D-HCSSA localization algorithm improves the node positioning accuracy and reduces the algorithm energy consumption.

3. 3DDV-Hop Localization Algorithm and Sparrow Search Algorithm

3.1. 3DDV-Hop Localization Algorithm. The 3DDV-hop (three-dimensional distance vector-hop) localization algorithm is a distributed localization algorithm based on the range-free approach [39]. The algorithm consists of three phases.

Step 1. Calculate the minimum hop count of the target node and each known node. The known node floods its information group to the neighbour node, including the localization information of the known node and the hop count with an initial value of zero. By receiving the minimum counts of hop for each known node, ignoring the group from the same known node with a larger count of hop, the hop count is automatically added by 1 and finally forwarded to the neighbour nodes.

Step 2. Calculate the distance between the target node and the known node. The known nodes calculate the average hop size by each known node based on the minimum counts of hop of other known nodes recorded in Step 1, and broadcast the average hop size information in the network to estimate the average hop size:

$$\text{AveHopsize}_I = \frac{\sum_{I \neq J} \sqrt{(x_I - X'_J)^2 + (y_I - Y'_J)^2 + (z_I - Z'_J)^2}}{\sum_{I \neq J} h_{IJ}}, \quad (1)$$

where $(x_I, y_I, z_I), (X'_J, Y'_J, Z'_J)$ are the coordinates of known nodes I and J , respectively; h_{IJ} is the hop count between the two known nodes; and AveHopsize_I is the average hop size of known node I . Each known node uses equation (1) to estimate the average actual distance per hop based on the localization information and the number of hop counts away from other known nodes recorded in the first phase.

The distance from the known node to the target node is given by the following equation:

$$\text{HS}_{Iu} = \text{AveHopsize}_I \times \text{Hop}_{Iu}, \quad (2)$$

where Hop_{Iu} is the hop count between the known node I and the target node u .

Step 3. Calculate the coordinates of target nodes. After deriving the expected distances between four and more target nodes to the known node in a three-dimensional space, the

coordinates of the target nodes are calculated by the great likelihood estimation method.

3.2. Sparrow Search Algorithm. The sparrow search algorithm (SSA) is a swarm intelligence optimization algorithm inspired by sparrow foraging and antipredation behaviour proposed by Xue and Shin [21] in 2020. There are two roles in SSA: producer and scrounger. Producers forage and provide guidance, while the scrounger obtains food through the producer. In order to better obtain quality food, members of the population monitor each other's behaviour, and in order to increase their own predation rate, scroungers compete for food with the high intake sparrows. The localization of the sparrow population can be expressed as

$$\Pi = \begin{bmatrix} \Pi_{1,1} & \Pi_{1,2} & \cdots & \cdots & \Pi_{1,d} \\ \Pi_{2,1} & \Pi_{2,2} & \cdots & \cdots & \Pi_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Pi_{n,1} & \Pi_{n,2} & \cdots & \cdots & \Pi_{n,d} \end{bmatrix}, \quad (3)$$

where n represents the number of the sparrow population and d is the optimal dimension.

The fitness of each sparrow is

$$\text{Fitness}_{\Pi} = [\zeta(\Pi_1), \zeta(\Pi_2) \cdots \zeta(\Pi_n)]^T, \quad (4)$$

where $\zeta(\Pi_1), \zeta(\Pi_2) \cdots \zeta(\Pi_n)$ are the fitness values of each sparrow, respectively.

Define producer updated localization:

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot \text{iter}_{\max}}\right), & \text{if } R_2 < ST, \\ x_{i,j}^t + Q \cdot L, & \text{if } R_2 \geq ST, \end{cases} \quad (5)$$

where t represents the current iteration number and iter_{\max} represents a constant of the maximum iteration number. $x_{i,j}$ represents the position information of the i th sparrow in dimension j . $\alpha \in (0, 1]$ is a random number. $R_2 \in [0, 1]$ and $ST \in [0.5, 1]$ represent the warning value and the safe value, respectively. Q is a random number. L represents a $1 \times d$ matrix. When $R_2 < ST$, it represents that there are no predators around the foraging environment at this time, and the producer can perform extensive search operations. When $R_2 \geq ST$, indicating the detection of a predator and alerting members, all sparrows fled to a safe place before foraging.

The updated localization of the scrounger is defined as follows:

$$x_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{x_{\text{worst}}^t - x_{i,j}^t}{i^2}\right), & \text{if } i > \frac{n}{2}, \\ x_p^{t+1} + |x_{i,j}^t - x_p^{t+1}| \cdot A^+ \cdot L, & \text{otherwise,} \end{cases} \quad (6)$$

where X_p is the producer optimal localization and X_{worst} is the global worst position. A represents a matrix of $1 \times d$,

where the elements have random values of -1 or 1 , and $A^+ = A^T(AA^T)^{-1}$. $i > n/2$ represents that the i th scrounger with a low fitness value has not obtained food, is in a hungry state, and needs to forage for food in order to forage for more food in other places.

SSA has strong local search ability and fast convergence speed but weak global search ability and weak operation of jumping out of the local optimum. Therefore, this article initializes the initial coordinates of known nodes and target nodes and proposes a method to further optimize the SSA when figuring out the target node coordinates.

3.3. Improved Sparrow Search Algorithm. This article improves the sparrow search algorithm by introducing K -means clustering and sine-cosine search strategy, using mathematical methods and clustering algorithm to optimize the sparrow search algorithm, solving the problem that the sparrow population is prone to fall into the local optimum when initializing, and improving the convergence accuracy and optimization effect of the traditional SSA. The improved SSA is applied to the third phase of the 3DDV-hop localization algorithm calculating target node coordinates. Through simulation experiments, the improved SSA accelerates the convergence speed and solves the problem that the algorithm is easy to fall into the local optimum and corrects the deviation of target node coordinates. In summary, the improved SSA can solve the problem of position deviation when calculating the target node coordinates and improve the positioning accuracy.

3.3.1. K -means Clustering Initialization. Since the SSA uses the random distribution principle in population initialization, the principle suffers from the problem that the sparrow population initialization cannot completely cover the solution space and there are coverage voids, which makes it difficult to traverse various cases of the solution set in the population and leads to the problem that the algorithm is prone to fall into the local optimum under complex multi-peaked functions. Therefore, in this article, the K -means clustering method is used to solve the deficiency that the algorithm is prone to fall into the local optimum when the population is initialized [40].

By choosing a similarity measurement method, the two-sample producer and scrounger are divided separately. Among them, the producer represents the known node coordinate, and the scrounger represents the target node coordinate. This partitioning makes the data within the same category as similar as possible. Using the Euclidean distance as the measurement standard, the samples with similar distances are divided into different class clusters until the set of the two samples initially divided is obtained. The mean value of each class cluster sample was taken as the centre of the next cluster, and the distance between the remaining samples and the new cluster centre was calculated and classified. Iterate repeatedly until the clustering criterion function converges or reaches the number of iterations and finally improves the global search ability of SSA.

$w = \{\Theta, \Xi\}$ is divided into two clusters: w_1, w_2 , where $w_i = (\Theta_i, \Xi_i)^T$, c_1, c_2 are the two initial clustering centers, and the methods used in the clustering process are as follows:

Euclidean distance between samples Θ, Ξ :

$$d(\Theta, \Xi) = \sqrt{(\Theta - \Xi)^T(\Theta - \Xi)}. \quad (7)$$

Average distance from sample Θ to all samples:

$$m = \frac{1}{n} \sum d(\Theta, \Xi). \quad (8)$$

Sample variance:

$$\text{var} = \frac{1}{n-1} \sum [d(\Theta, \Xi) - m]^2. \quad (9)$$

Average distance of data set samples:

$$d = \frac{2}{n(n-1)} \sum \sum d(\Theta, \Xi). \quad (10)$$

Sum of squares of error:

$$S = \sum_{i=1}^2 \sum (\Theta - c_i)^2. \quad (11)$$

Assuming a population size of 100, Figures 1 and 2 are the populations initialized by random distribution and K -means clustering, respectively. After the comparison between Figures 1 and 2, it is obvious that the population distribution of the latter is more uniform than that of the former, which fully covers the knowledge space. Therefore, the algorithm has better ergodicity and improves the global search ability.

3.3.2. Sine-Cosine Search Strategy. The sine and cosine search strategy was proposed by Mirjalili in 2016 [41], and its main idea is to enhance the convergence of the algorithm by iteratively continuously optimizing the solution set of the objective function through the mathematical method sine and cosine function properties for the local and global search of the population. In the SSA, the scrounger position update is mainly influenced by the discoverer, which leads to the disadvantage of poor algorithm search effect, and in order to improve the convergence accuracy and search effect, this article uses an improved sine and cosine search strategy to optimize the producer position [41].

The improved sine and cosine search strategy expression can be expressed as

$$\phi(q+1) = \begin{cases} \phi(q) + r_1 \cdot \sin(r_2) \cdot D_{r_3} < 0.5, \\ \phi(q) + r_1 \cdot \cos(r_2) \cdot D_{r_3} \geq 0.5, \end{cases} \quad (12)$$

where r_2, r_3 are random factors and the corresponding value ranges are $(0, 360^\circ)$ and $[0, 1]$, respectively. Among them, r_2 determines the moving distance in the iterative process, r_3 is

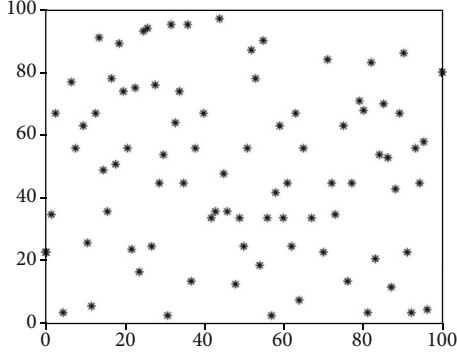


FIGURE 1: Randomly distributed initial population.

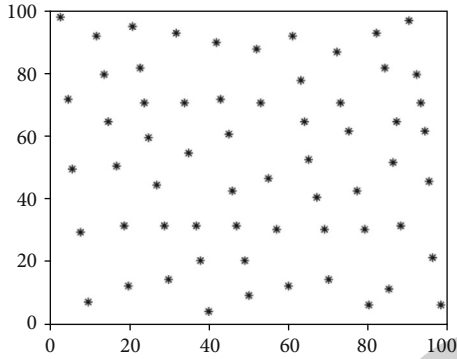


FIGURE 2: K-means cluster initialization population.

the random weight coefficient assigned to the current optimal solution to enhance (when $r_3 > 1$) or weaken (when $r_3 < 1$) the influence of the optimal solution on the defined distance, and r_1 represents the control factor, which is a linear decreasing function that controls the fluctuation amplitude of the sine and cosine function and also determines the movement direction of the iteration. If $r_1 < 1$, the next solution will move from the current solution to the region of the optimal solution; if $r_1 > 1$, it will move in the reverse direction. The expression of the control factor is as follows:

$$r_1 = \varepsilon \left(1 - \frac{t}{T} \right), \quad (13)$$

where ε is a constant and T represents the maximum number of iterations.

The process of seeking optimization of the sine-cosine optimization algorithm mainly includes two phases: global search and local search. As the number of iterations t increases, r_1 demonstrates a linear decreasing trend. Therefore, the convergence speed of the algorithm further accelerated the convergence rate. The optimized position of the scrounger is iteratively calculated by using the sine-cosine search strategy (equation (13)) and the control factor (expressions (14) and (15)).

$$D = \{|r_4 \cdot x_i(t) - x_i(t)| i = 1, 2 \dots d\}, \quad (14)$$

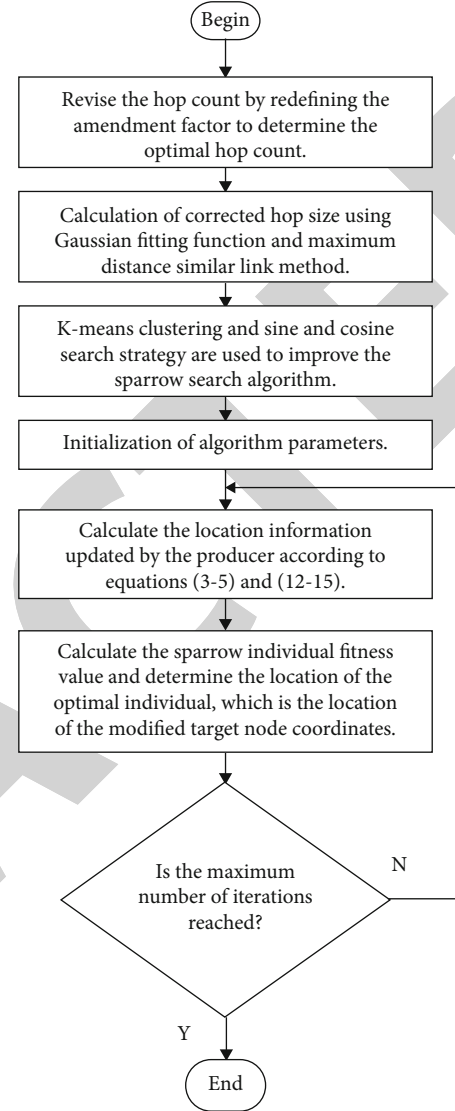


FIGURE 3: Flow chart of the 3D-HCSSA algorithm.

where r_4 is the random factor on $[0, 1]$, which represents the switching condition of the iterative equation of the sine and cosine function.

$$r_3 = \varepsilon - \varepsilon \cdot \frac{t}{M}, \quad (15)$$

where M represents the number of iterations and ε is a constant.

4. 3D-HCSSA Localization Algorithm

The 3D-HCSSA algorithm in this article is based on the hop size correction and the improved sparrow algorithm to optimize the 3DDV-hop algorithm. The 3DDV-hop algorithm is improved and optimized from three aspects: hop count division, hop size correction, and optimization of target node coordinates. The simulation results demonstrate that the improved algorithm proves that its positioning accuracy has

Algorithm 1: Finalization method to solve the sparrow optimization algorithm to solve the coordinate value
Input: Known node coordinates, the distance from the target node to the corresponding known node.
output Expected coordinates of target nodes.
Initialize a population of n sparrows and define its relevant parameters: M, Maximum number of iterations; PD, Population Number of Producers; SD, the number of sparrows who perceive the danger; R₂, Alert value; N, Total Population Quantity.
//Initial parameter Value Represents the initialization parameter value
1. **While**(t<G) **// The iteration termination condition is judged**
2. Sort the fitness values to find the current individual best and worst.
3. R₂=rand (1)
4. **for** i=1: PD
5. The updated producer localization is calculated by equation (3);
6. **end for**
7. **for** i=(PD+1): N
8. The updated scrounger positions are calculated by equations (12-15);
9. **end for**
10. **for** i=1:SD
11. The sparrow position is updated by equation (6);
12. **end for**
13. Get the current latest position;
14. Update if the latest position is better than the previous one;
15. t = t + 1;
16. **end while**
17. **return** X_{best}, f_g.

ALGORITHM 1: Sparrow optimization algorithm to calculate the target node coordinate value.

TABLE 1: Parameter settings.

Algorithm	Parameter
SSA	ST = 0.8 PD = 0.2SD = 0.3
LSSA	ST = 0.8 PD = 0.2SD = 0.3
HCSSA	ST = 0.8 PD = 0.2SD = 0.3

been improved from many aspects, and the node energy consumption is greatly reduced, resulting in a reduction in the overall energy consumption of the algorithm.

4.1. Calculate the Hop Count of the Known Node. In the 3DDV-hop algorithm, the distance between the target node within one hop of the known node and the known node in the actual network is different, which will result in lower localization accuracy. Therefore, it is necessary to optimize and improve the partition of hop count. By making a difference between the expected hop count of a known node and the perfect hop count, the ratio of the difference to the expected hop count redefines the amendment factor to reduce the cumulative error caused by hop count division. The specific steps are as follows.

Step 1. Set the perfect hop count. Suppose the real distance between known nodes i and j is d_{ij} ; the ratio of real distance between known nodes i , j and the maximum communication radius R is defined as the perfect hop count H_{ij} [42]:

$$H_{ij} = \frac{d_{ij}}{R}. \quad (16)$$

Step 2. Set the amendment factor. The expected hop count of

a known node makes a difference with the perfect hop count, and the ratio of the difference to the expected hop count is

$$\varphi_{ij} = \frac{h_{ij} - H_{ij}}{h_{ij}}, \quad (17)$$

where h_{ij} is the expected counts of hop between known nodes i, j . φ_{ij} represents the amount of correction between the expected counts of hop and the perfect counts of hop which can reflect the difference between the expected counts of hop and the perfect counts of hop. Use φ_{ij} to redefine the amendment factor ω_{ij} as

$$\omega_{ij} = 1 - \left(\frac{h_{ij} - H_{ij}}{h_{ij}} \right)^2. \quad (18)$$

Step 3. The corrected optimal hop count can be calculated by the amendment factor as

$$h'_{ij} = \omega_{ij} h_{ij}, \quad (19)$$

where h'_{ij} represents the optimal hop count.

The corrected optimal hop count obtained through the above steps can be used for the calculation of the next section.

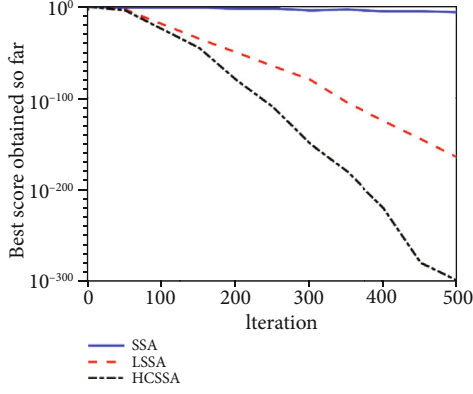
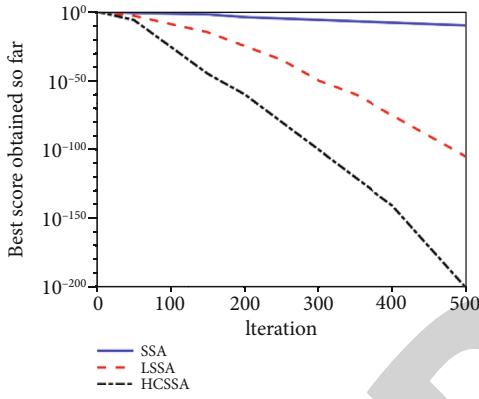
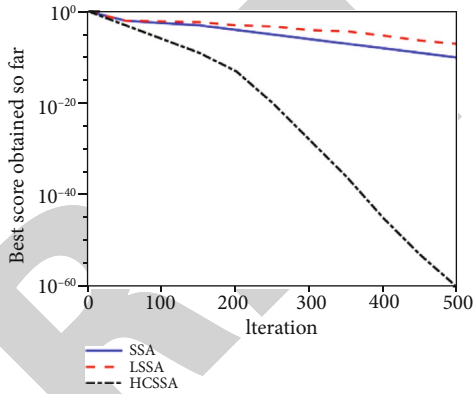
4.2. Hop Size Correction. This section mainly amends the one-hop size and multihop size between the known node and target node.

TABLE 2: 20 benchmark functions.

Number	Function	Dimensions	Interval	Minimum value
1	$F_1(x) = \sum_{i=1}^n (10^6)^{(i-1)/(n-1)} x_i^2$	30	[-100,100]	0
2	$F_2(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	[-100,100]	0
3	$F_3(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30	[-100,100]	0
4	$F_4(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30,30]	0
5	$F_5(x) = \sum_{i=1}^n (x_i + 0.5)^2$	30	[-100,100]	0
6	$F_6(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0,1]$	30	[-1.28,1.28]	0
7	$F_7(x) = \sum_{i=1}^n ix_i^4 + \text{rand}[0,1]$	30	[-1.28, -1.28]	0
8	$F_8(x) = \sum_{i=1}^n x_i \sin(x_i) + 0.1x_i $	30	[-100,100]	0
9	$F_9(x) = \sum_{i=1}^n x_i^2$	30	[-10,10]	0
10	$F_{10}(x) = \sum_{i=1}^n [x_i^2 - 10 \cos 2\pi x_i + 10]$	30	[-5.12,5.12]	0
11	$F_{11}(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	[-500,500]	-418.9829n
12	$F_{12} = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + y_i = 1 + x_i + \frac{1}{4}$	30	[-50,50]	0
13	$F_{13} = 418.9829n - \sum_{i=1}^n x_i \sin(\sqrt{ x_i })$	30	[-500,500]	0
14	$F_{14}(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$	30	[-4.5,4.5]	0
15	$F_{15}(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)) + 19.8(x_2 - 1)((x_4 - 1))$	30	[-10,10]	0
16	$F_{16}(x) = \left(0.02 + \sum_{i=1}^{25} (1/i) + (x_1 - a_{1i})^6 + (x_2 - a_{2i})^6 \right)^{-1}$ $a = \begin{pmatrix} -32 - 16.01632 - 32 \dots 0.1632 \\ -32 - 32 - 32 - 32 - 16 \dots 32.3232 \end{pmatrix}$	30	[-65.536,65.536]	0.998

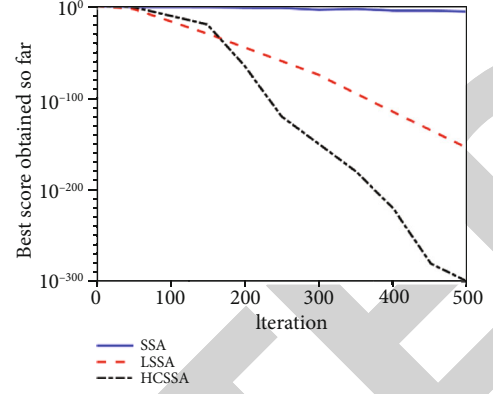
TABLE 2: Continued.

Number	Function	Dimensions	Interval	Minimum value
17	$F_{17}(x) = \sum_{i=1}^n x_i^2 $	30	[-1,1]	0
18	$F_{18}(x) = (x_i + 0.5)^2$	30	[-100,100]	0
19	$F_{19}(x) = \max_i x_i , i \in [1, m]$	30	[-100,100]	0
20	$F_{20}(x) = \sum_{i=1}^n ix_i^2$	30	[-10,10]	0

FIGURE 4: A roundabout way of F_1 .FIGURE 5: A roundabout way of F_2 .FIGURE 6: A roundabout way of F_3 .

4.2.1. Correction of One-Hop Size. In the 3DDV-hop localization algorithm, the same average hop size is used between the known node and the target node to calculate, which not only increases the hop size error but also increases the energy cost. Therefore, in one-hop size, the distance value is derived as the hop size by proposing that through the signal strength value of the Received Signal Strength Indicator (RSSI).

Since RSSI is based on the radio wave loss between the transmitting end and the receiving end, the transmission loss is converted into distance according to a specific signal

FIGURE 7: A roundabout way of F_4 .

model [43], and the most widely used logarithmic-normal distribution model is

$$\text{RSS}(d)[\text{dBm}] = P_{\text{tr}} - P_{\text{loss}}(d') - 10\alpha \log_{10} \frac{d}{d'} + x_{\xi}, \quad (20)$$

where $\text{RSS}(d)$ represents the signal strength received by the target node from a known node, d represents the distance between the known node and the target node, P_{tr} represents the transmission signal energy, $P_{\text{loss}}(d')$ represents the signal power loss at the reference distance d' , α represents the path loss index, whose value depends on the transmission medium, and d' represents the reference distance. x_{ξ} represents the noise, which is a Gaussian random variable with a mean of 0. ξ is the standard deviation.

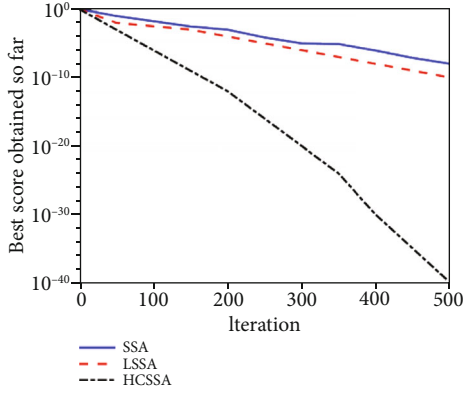
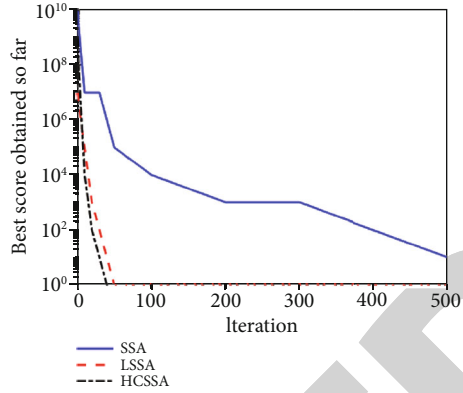
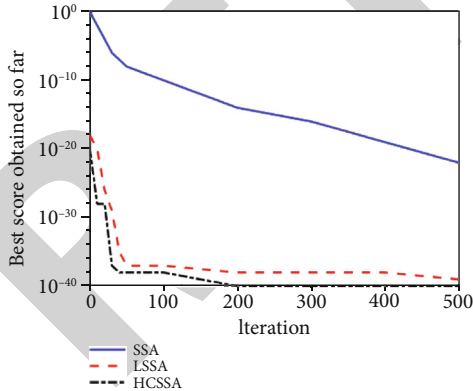
According to equation (20), it can be concluded that the hop size between the known node and the target node is

$$d = \left(10d'\right)^{\left(\text{RSS}(d)[\text{dBm}] - P_{\text{tr}} + P_{\text{loss}}(d') - x_{\xi}\right)/10\alpha}. \quad (21)$$

However, the RSSI values received by target nodes will be lost to a certain extent at a certain distance [44], resulting in deviations in the distance values, resulting in a cumulative error in the derived distance value. Therefore, we use the Gaussian fitting function to further correct the distance value. The collected RSSI values are processed through the Gaussian fitting function, the abnormal data are screened, and the processed RSSI values are substituted into equation (22) to obtain a more accurate internode distance. The optimized RSSI value is closer to the real hop size value. The Gaussian fitting function is as follows [45]:

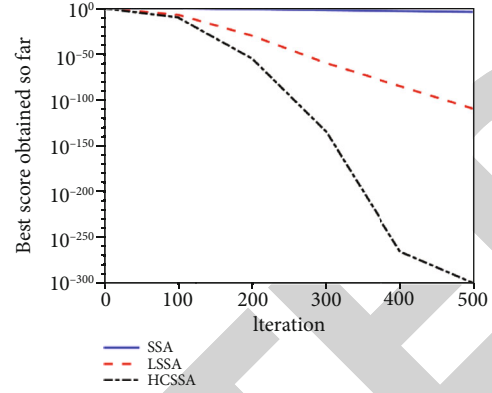
$$f(x_i) = x_0 + \frac{A}{\sigma\sqrt{\pi/2}} \times e^{-2((x_i - \mu)^2/\sigma^2)}, \quad (22)$$

where mean $\mu = \sum_{i=1}^N \text{RSSI}_i / N$, standard deviation $\sigma = \sqrt{\sum_{i=1}^N (\text{RSSI}_i - \mu)^2 / (N - 1)}$, x_0 and A are undetermined coefficients, which are determined by the relationship between the known node position and the signal value, x_i

FIGURE 8: A roundabout way of F_5 .FIGURE 9: A roundabout way of F_6 .FIGURE 10: A roundabout way of F_7 .

represents the i th signal distance value, and N is the total value of the received RSSI.

Most of the RSSI values in the Gaussian function are distributed between $[\mu - \sigma, \mu + \sigma]$, and the average value of this interval represents that the measured and actual values are close to each other. After being filtered by Gaussian filtering, the currently observed RSSI value output is

FIGURE 11: A roundabout way of F_8 .

$$\text{RSSI}' = \frac{1}{n} \sum_{i=1}^n x_i(x_i \in [\mu - \sigma, \mu + \sigma]), \quad (23)$$

where n represents the number of RSSI values in $[\mu - \sigma, \mu + \sigma]$.

The ranging distance corrected by Gaussian fitting is the corrected distance of one-hop size:

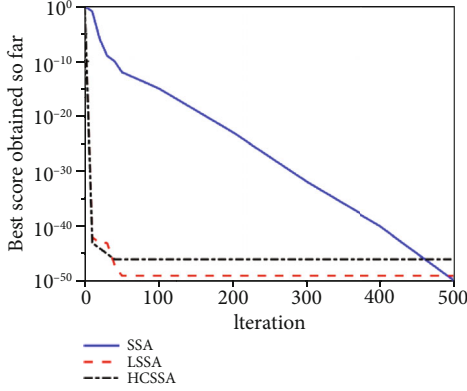
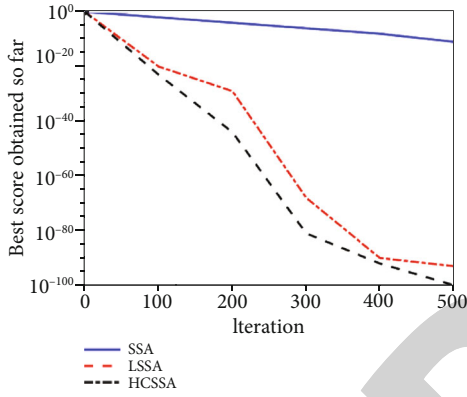
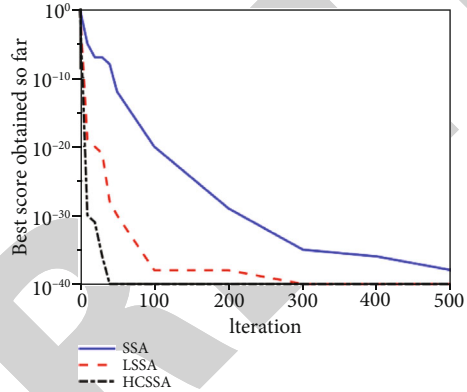
$$d_N = \left(10d'\right)^{(\text{RSSI}' - P_{\text{tr}} + P_{\text{loss}}(d') - x_{\xi})/10\alpha}. \quad (24)$$

4.2.2. Correction of Multihop Size. In this article, when calculating the multihop size between a known node and a target node, the maximum distance similar link method is proposed based on the classical similar path search algorithm, which corrects the average hop size between the target node and the known node by finding the most similar pair of known node paths for the link from the target node to the noncoplanar known node. As the corrected average hop size value between the known node and the multihop target node, the optimal distance is obtained by using this average hop size multiplied by the optimized optimal hop size.

Step 1. Determine the maximum similarity link. After the exchange of the node information, each node keeps the information of each known node, as well as the identification of other nodes in the path to the minimum hop count experienced by each known node. The similarity link (SL) factor represents the path from the target node to a specific known node [46] and the degree of similarity between the paths from other known nodes to a specific known node. The calculation method is as follows:

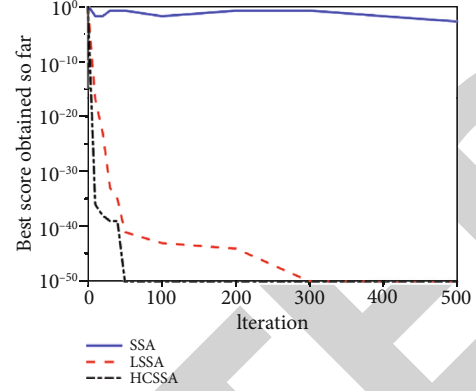
$$\text{SL}(\kappa, \omega) = \frac{n(\kappa \cap \omega)}{\sqrt{n(\kappa) \times n(\omega)}}, \quad (25)$$

where $\text{SL}(\kappa, \omega)$ represents the similarity between the path κ and path ω , κ represents target node to the specific noncoplanar known nodes by the minimum hop count between nodes on the link set of identification, ω represents the residual known node in the network to the specific noncoplanar between known nodes, after the minimum hop count by a

FIGURE 12: A roundabout way of F_9 .FIGURE 13: A roundabout way of F_{10} .FIGURE 14: A roundabout way of F_{11} .

collection of node id on a link, and $n(\cdot)$ is the number of elements in set.

The similarity link factor SL is used to calculate the path from the target node to a specific noncoplanar known node and the similarity degree between the paths from other known nodes to a specific noncoplanar known node. The greater the value of SL, the higher the similarity degree. The path between the known nodes with the maximum value is determined as the possible path. When the calculated SL values with multiple paths are the same, compare

FIGURE 15: A roundabout way of F_{12} .

the Euclidean distance between each known node pair in the same SL value path. When the distance between two points of the node is larger, the calculated energy consumption of the node path is smaller, and the calculated average hop size is closer to the actual each hop size. Therefore, the path with the largest Euclidean distance of known node is selected as the result.

Step 2. Calculate the average hop size of similar links with maximum distance. The path obtained by Step 1 is the path between two known nodes, including specific known nodes. The Euclidean distance between nodes is calculated by using the coordinates of the two known nodes. The calculated distance is divided by the optimized minimum hop count between the two nodes, which is the average hop size used by the target node.

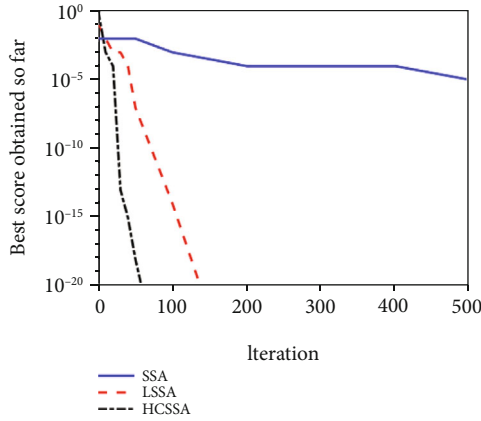
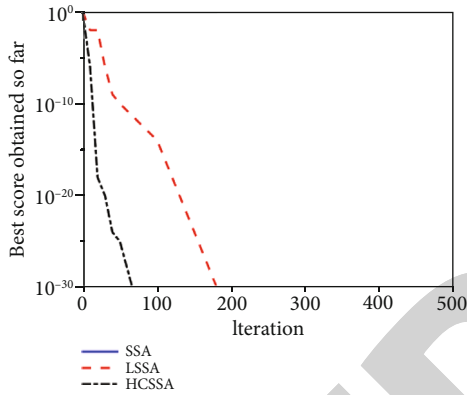
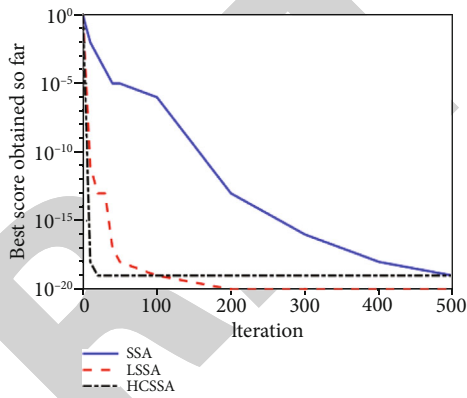
Step 3. Calculate the multihop size between the known node and the target node. To obtain the distance to the noncoplanar known node, each target node uses the calculated average hop size multiplied by the optimal hop count.

4.3. Calculate Target Node Coordinates

4.3.1. Determine Fitness Value. The sine and cosine function search method effectively enhances the convergence of the sparrow algorithm, improves the convergence accuracy and the optimization effect, and improves the iterative efficiency. For this reason, this article redesigns the fitness function so that the fitness value after iteration can better correct the position deviation of the target node:

$$\text{fitness} = \frac{1}{n} \cdot \sum_{j=1}^n \left| \sqrt{(x_I - X'_j)^2 + (y_I - Y'_j)^2 + (z_I - Z'_j)^2} - D_{I,u} \right|, \quad (26)$$

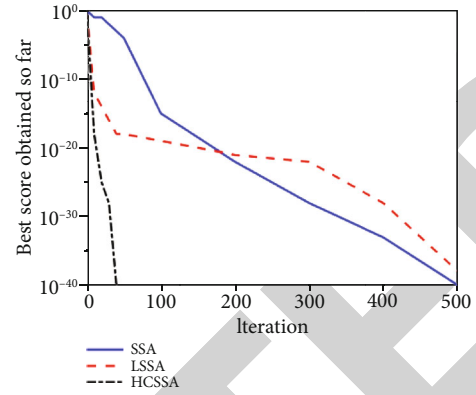
where (x_I, y_I, z_I) , (X'_j, Y'_j, Z'_j) are the coordinates of the target node and known node, respectively. $D_{I,u}$ is the distance from the target node u to the known node I .

FIGURE 16: A roundabout way of F_{13} .FIGURE 17: A roundabout way of F_{14} .FIGURE 18: A roundabout way of F_{15} .

4.3.2. 3D-HCSSA Algorithm Flow. The localization process of the 3DDV-hop algorithm based on 3D-HCSSA can be described as follows:

Step 1. The algorithm revises the hop counts by redefining the amendment factor to determine the optimal hop counts.

Step 2. Firstly, the Gaussian fitting function is used to reduce the error between the measured and actual values of the one-hop size between the known node and the target node; sec-

FIGURE 19: A roundabout way of F_{16} .

ondly, the average hop size between the known node and the target node of this path is corrected according to the maximum distance similar link method proposed in this article, and the distance between the known node and the target node is calculated after the correction.

Step 3. In the three-dimensional space, $100\text{ m} \times 100\text{ m} \times 100\text{ m}$ -means clustering and sine and cosine search strategy are used to improve the sparrow search algorithm to optimize the localization of target nodes. Set the number of sparrow populations, and set the number of iterations and the localization of the population in initialization.

Step 4. Calculate the fitness values of all sparrow individuals through the fitness function, sort the fitness values of all sparrow individuals, and record the current optimal and worst sparrow positions.

Step 5. The entire iterative process iterates according to equations (3), (4), (5), and (12)–(15), updates the population position and calculates the new fitness value, and records the fitness value and position of the best individual after the update.

Step 6. The algorithm loops and calculates Step 5, and after iterating to the maximum, the global optimal position of the producer is obtained, that is, the optimal value of the coordinates of the target node and the algorithm.

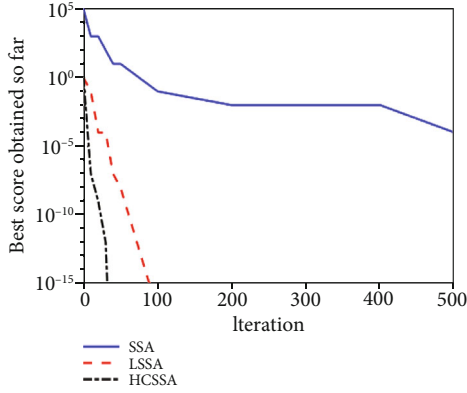
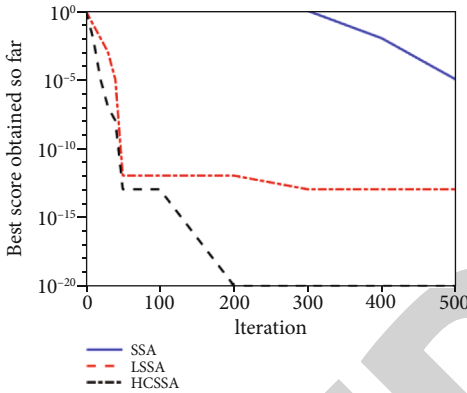
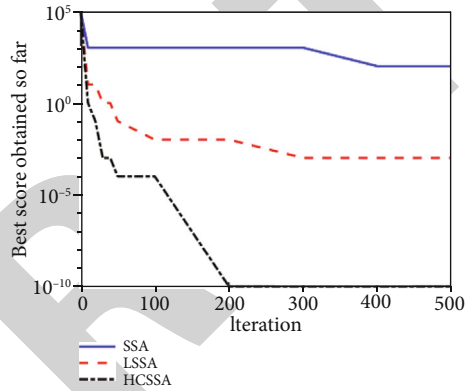
The flow chart of the 3D-HCSSA algorithm is shown in Figure 3.

Algorithm 1 is the pseudo-code of the spark search algorithm used to calculate the location of the target node.

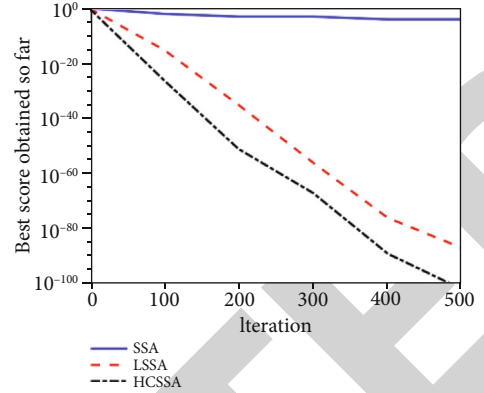
5. Simulation Results and Analysis

5.1. Evaluation of the Performance. This article compared HCSSA with SSA [21] and LSSA [47]. The parameters are set as shown in Table 1.

This section analyzes the performance of the improved sparrow search algorithm with 20 benchmark functions; as shown in Table 2, most of the benchmark functions come from the literatures [20, 42].

FIGURE 20: A roundabout way of F_{17} .FIGURE 21: A roundabout way of F_{18} .FIGURE 22: A roundabout way of F_{19} .

The algorithm in this article is compared with SSA and LSSA by 20 benchmark functions; Figures 4–23 show the details of the convergence curves of the SSA, the LSSA, and the HCSSA on the benchmark functions. From the figure, the iterative functions F_6 , F_{13} , F_{14} , F_{16} , F_{17} , F_{18} , F_{19} , and F_{20} find the optimal solution before the number of iterations is reached in the iterative process. From this iterative process, the convergence speed of the HCSSA is better than the other two algorithms, and the algorithm is

FIGURE 23: A roundabout way of F_{20} .

less likely to fall into a local optimum. The convergence curves of each function are shown in Figures 4–23.

5.2. Simulation Analysis. In order to verify the performance of the proposed algorithm in localization, the 3DDV-hop localization algorithm, 3D-GAIDV-hop localization algorithm, and HCLSO-3D localization algorithm were analyzed separately. The relationship between the number of known nodes, the communication radius, and the total number of nodes and energy consumption is compared to analyze the positioning accuracy of the four algorithms.

5.2.1. Experimental Parameter Setting. To verify the performance of the algorithm 3D-HCSSA in localization in this article, simulation experiments are conducted using MATLAB. 100 simulation experiments were conducted in the same environment, and the average value is taken as the final result of simulation experiments. $100\text{ m} \times 100\text{ m} \times 100\text{ m}$ is set as the simulation area, and 100 target nodes and 30 known nodes are taken. The specific experimental parameters are shown in Table 3. The random distribution of nodes in the 3D space is shown in Figure 24.

5.2.2. The Localization Performance of the Model Is Evaluated by Means of the Average Localization Error Metric.

$$\gamma_{\text{ALE}} = \frac{\sum_{I=1}^n \sqrt{(x_I - X'_I)^2 + (y_I - Y'_I)^2 + (z_I - Z'_I)^2}}{NR}, \quad (27)$$

where γ_{ALE} represents the average localization error, (x_I, y_I, z_I) represents the actual coordinate value of the target node I , (X'_I, Y'_I, Z'_I) is the expected coordinate value of the node I , N represents the total number of target nodes, and R represents the communication radius of the known node.

5.2.3. Analysis of the Number of Known Nodes and the Average Localization Error. From Figure 25, it can be seen that the average localization error of the algorithm in this article is smaller when the known nodes are the same. When the proportion of known nodes is 30%, the average localization error of the 3D-HCSSA algorithm is the smallest at 8.5%. At this time, the average localization errors of 3DDV-hop, 3D-GAIDV-

TABLE 3: Network environment and parameter setting.

Parameter	Value
The network area	100 m × 100 m × 100 m
Total number of nodes	100-300
Number of known nodes	10-30
Communication radius	25-45
Maximum iteration	300

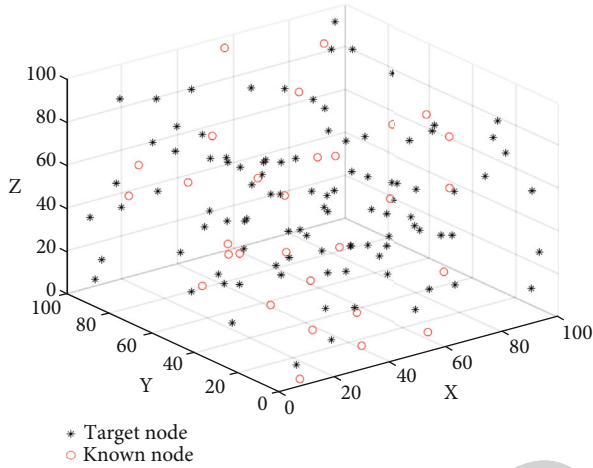


FIGURE 24: Random distribution of nodes in the three-dimensional space.

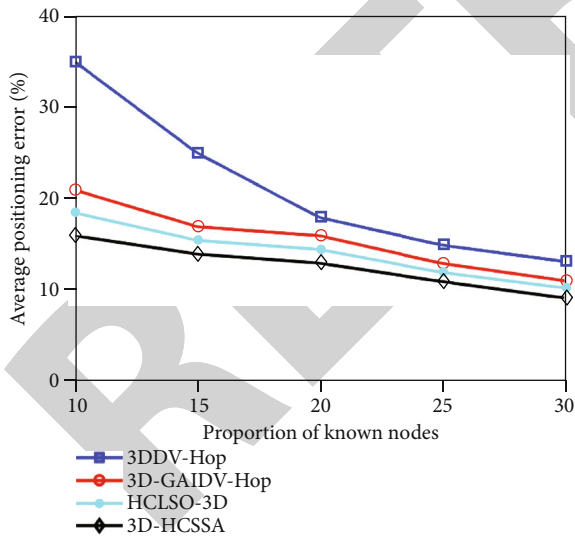


FIGURE 25: Compare and analyze the localization errors of different numbers of known nodes.

hop, and HCLSO-3D are 12.5%, 11.3%, and 9.8%, respectively, and the average localization errors of the 3D-HCSSA algorithm reduces compared with these three algorithms by 47.06%, 32.94%, and 15.29%, respectively. Therefore, when the number of known nodes is constant and the average positioning error curve is shown in the figure, the 3D-HCSSA algorithm has

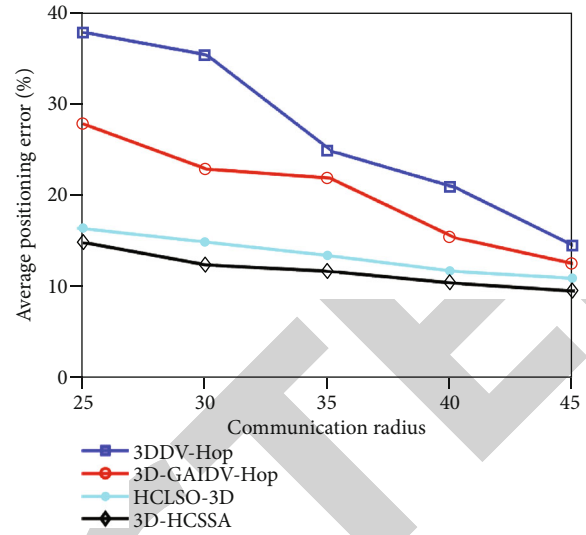


FIGURE 26: Compare the localization errors of different communication radii.

the advantage of high positioning accuracy compared with the other three algorithms.

5.2.4. *Analysis of Different Communication Radii and Average Localization Errors.* Figure 26 demonstrates the curve of average localization error that varies with communication radius. The proportion of known nodes is 10% in the network, and the communication radius is increased from 25 m to 45 m. By changing the communication radius, the localization performance of the four algorithms is compared and analyzed. The average localization error of the 3D-HCSSA algorithm is always the smallest. As the communication distance increases, the known node communicates directly with more nodes, so the average localization error gradually tends to be stable. The average localization error of the 3D-HCSSA algorithm is 9.6% minimum at the communication radius equal to 45 m. In this case, the average localization errors of 3DDV-hop, 3D-GAIDV-hop, and HCLSO-3D algorithms are 14.3%, 11.5%, and 11.2%, respectively. Compared with these three algorithms, the average localization errors of the 3D-HCSSA algorithm are reduced by 48.96%, 19.79%, and 16.67%, respectively. In summary, we can see by the change curve of communication radius from the figure that the average positioning error of the 3D-HCSSA algorithm is the smallest in different communication radius ranges, so the proposed algorithm in this article has the feature of high positioning accuracy.

5.2.5. *Total Number of Nodes and Average Localization Error Analysis.* Figure 27 demonstrates the curve of average localization error that varies with the total number of nodes. Sensor nodes are randomly deployed throughout the network with a communication radius of 30 m. The performance of the four algorithms demonstrates that the average localization error decreases gradually with the curve. The reason for this is that the network connectivity becomes better as the node density increases. Therefore, when the total number of nodes reaches a certain value, the average localization

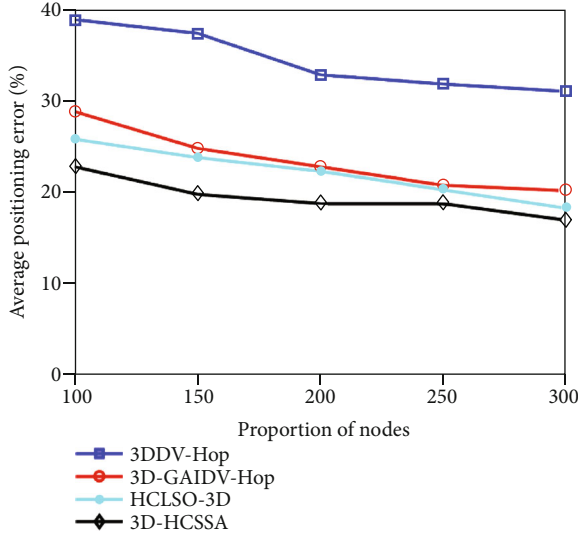


FIGURE 27: Compare and analyze the total number of nodes and the average localization error.

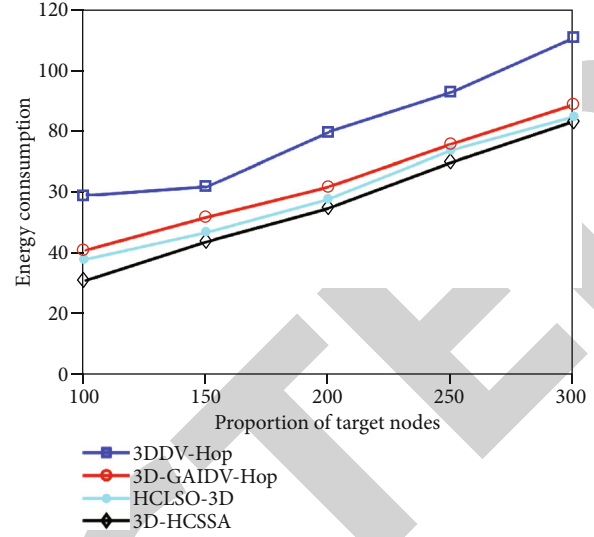


FIGURE 29: Compare and analyze the number of target nodes and energy consumption.

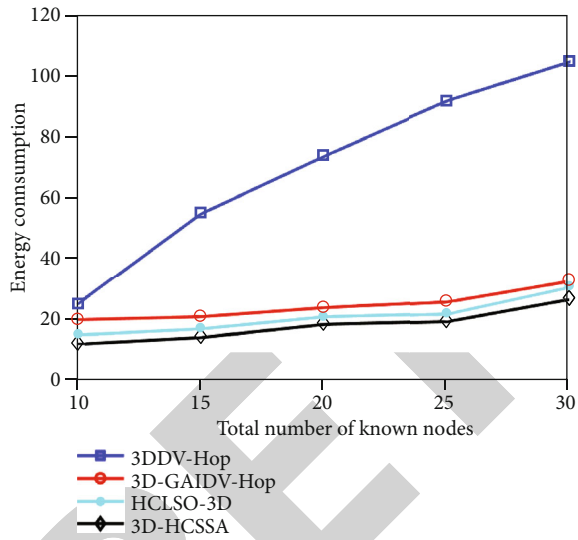


FIGURE 28: Compare and analyze the number of known nodes and energy consumption.

error of all four algorithms gradually stabilizes and does not change significantly. The average localization error of the 3D-HCSSA algorithm is always the smallest under the same conditions. When the total number of nodes reaches 300, the average localization error of the 3D-HCSSA algorithm is the smallest at 19.2%. In this case, the average localization errors of 3DDV-hop and 3D-GAIDV-hop and HCLSO-3D algorithms are 31.2%, 22.4%, and 19.9%, respectively. The 3D-HCSSA algorithm reduces the average localization error by 62.5%, 16.67%, and 3.64%, respectively, by comparison. Therefore, compared with the other three algorithms, the average positioning error of this algorithm is significantly reduced and the positioning accuracy is improved.

5.2.6. Influence of the Number of Known Nodes on Energy Consumption. Figure 28 demonstrates the curve of network energy consumption that varies with the number of known nodes. The energy consumption of the four algorithms increases gradually by randomly deploying sensor nodes throughout the network. The reason for this is that the increase in the number of known nodes leads to an increase in computation time. The energy consumption of the 3D-HCSSA algorithm is always the smallest throughout the process. When the number of known nodes is 30, the minimum energy consumption of 3D-HCSSA is 28. In this case, the energy consumption of 3DDV-hop, 3D-GAIDV-hop, and HCLSO-3D is 103, 35, and 33, respectively. Compared with the three algorithms, the energy consumption of 3D-HCSSA is reduced by 72.82%, 25%, and 20%, respectively. The known nodes consume large power in the network, which leads to the problem of high energy consumption of the whole network. It can be seen from the energy consumption curve in the figure that as the number of known nodes increases, the energy consumption is also increased, but the algorithm proposed in this paper has the advantage of low energy consumption compared with the other three algorithms.

5.2.7. Influence of the Number of Target Nodes on Energy Consumption. Figure 29 demonstrates the curve of network energy consumption that varies with the target number of nodes. The performance of the four algorithms are compared by randomly deploying sensor nodes within the entire network. The energy consumption of the four algorithms gradually increases as the total number of nodes increases, and the increase in the number of target nodes leads to an increase in computation time. The energy consumption of the 3D-HCSSA algorithm is always the smallest throughout the experiment. The energy consumption of the 3D-HCSSA algorithm is the smallest when the target node is 300 at 82. In this case, the energy consumption of 3DDV-hop, 3D-GAIDV-hop, and HCLSO-3D algorithms is 113, 90, and 87, respectively.

TABLE 4: Average running time of the four algorithms.

Algorithms	Average running time per second
3DDV-hop	32.3
3D-GAIDV-hop	49.2
HCLSO-3D	51.6
3D-HCSSA	53.4

Compared with the three algorithms, the energy consumption of the 3D-HCSSA algorithm is reduced by 37.8%, 9.76%, and 6.09%, respectively.

5.2.8. Analysis of Calculation Cost. The average running time of the algorithm is analyzed as an index of computational cost. Table 4 demonstrates the average running time of 3DDV-hop and 3D-GAIDV-hop and HCLSO-3D and 3D-HCSSA algorithms under the same experimental conditions. As can be seen from the experimental results, the average running time of the 3D-HCSSA localization algorithm is about 1.65 times that of the 3DDV-hop algorithm, about 1.08 times that of the 3D-GAIDV-hop algorithm, and about 1.03 times that of the HCLSO-3D algorithm. Because the 3D-HCSSA algorithm uses an intelligent algorithm to solve the target node coordinates, which results in the increase in the amount of calculation, the small increase in time consumption of the algorithm can be ignored, but the positioning accuracy is significantly improved.

5.2.9. Time Complexity Analysis of the Localization Algorithm. Since WSNs are usually limited by resources, in addition to positioning accuracy, the complexity of a localization algorithm is also one of the factors that must be considered. Suppose the number of nodes in the whole WSNs is n and the number of nodes is known to be m . The time complexity of computing the minimum counts of hop between nodes is $o(n^3)$ in the 3DDV-hop algorithm, the time complexity of calculating the actual distance between nodes is $o(n)$, the time complexity of calculating the hop size from the target node to the known node is $o(n \times m)$, and the time complexity of calculating the localization of the target node through the maximum likelihood estimation method is $o((n - m)^4)$. Similarly, the complexity of 3D-GAIDV-hop and HCLSO-3D mentioned in this article increases $o(n)$, respectively, on the basis of the original 3DDV-hop algorithm. The time complexity of the proposed algorithm is also increased by $o(n)$ on the basis of the 3DDV-hop algorithm. Therefore, the time complexity of the 3D-HCSSA algorithm does not increase significantly, but the positioning accuracy is better improved.

6. Conclusion

Aiming at the obvious error problem of node localization in the 3DDV-hop algorithm, a sparrow search localization algorithm 3D-HCSSA based on hop size correction and improvement is proposed. The algorithm redefines the amendment factor based on the original hop count to reduce the hop count error and proposes the maximum distance similar link method to correct the multihop size between

nodes and uses K -means clustering and sine-cosine search strategy to improve the traditional sparrow search algorithm. If it is insufficient, optimize the position of the target node and improve the positioning accuracy. The simulation results demonstrate that the average positioning errors of the 3D-HCSSA algorithm are reduced by 52.84%, 23.13%, and 11.87%, respectively, compared with the 3DDV-hop algorithm, the 3D-GAIDV-hop algorithm, and the HCLSO-3D algorithm. Therefore, the algorithm improves the node positioning accuracy and reduces the energy consumption of the algorithm. For the future research directions, we will focus on these aspects: firstly, to explore a better way to build an integrated network with seamless coverage within the network by combining 5G/6G technology with WSNs to solve the solution set coverage problem in the sparrow optimization search algorithm, reduce coverage voids, and improve positioning accuracy, and secondly, how to constrain the residual energy of nodes and enhance the application of the algorithm in various heterogeneous networks.

Data Availability

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

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 71961028 and the Education Research Project of AFCEC, China, under Grant 2019-AFCEC-079.

References

- [1] C. Yuan, W. Chen, and D. Li, "A hierarchical matrix decomposition-based signcryption without key-recovery in large-scale WSN," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 5929828, 10 pages, 2018.
- [2] B. Sheng, "On element selection in STAR-RIS for NOMA transmission," *Mobile Information Systems*, vol. 2022, Article ID 1335829, 8 pages, 2022.
- [3] T. F. Chen, L. J. Sun, Z. Q. Wang, Y. Wang, Z. Zhao, and P. Zhao, "An enhanced nonlinear iterative localization algorithm for DV_hop with uniform calculation criterion," *Ad Hoc Networks*, vol. 111, article 102327, 2021.
- [4] X. Y. Liu and C. Liu, "Wireless sensor network dynamic mathematics modelling and node localization," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 1082398, 8 pages, 2018.
- [5] Y. F. Jia, K. X. Zhang, and L. Q. Zhao, "Improved DV-hop location algorithm based on mobile anchor node and modified hop count for wireless sensor network," *Journal of Electrical and Computer Engineering*, vol. 2020, Article ID 9275603, 9 pages, 2020.

- [6] J. Cheon, H. Hwang, D. Kim, and Y. Jung, "IEEE 802.15.4 ZigBee-based time-of-arrival estimation for wireless sensor networks," *Sensors*, vol. 16, no. 2, p. 203, 2016.
- [7] K. Rao, T. Kumar, and C. Venkatnaryana, "Selection of anchor nodes in time of arrival for localization in wireless sensor networks," *Proceedings of the International Conference on Soft Computing Systems*, vol. 397, pp. 45–57, 2016.
- [8] Y. Zhang, M. Wang, Z. Tan et al., "Design of wireless sensor network localization algorithm based on TDOA," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 70–75, Chengdu, China, 2019.
- [9] W. Wang, P. Bai, Y. Zhou, X. L. Liang, and Y. B. Wang, "Optimal configuration analysis of AOA localization and optimal heading angles generation method for UAV swarms," *IEEE Access*, vol. 7, pp. 70117–70129, 2019.
- [10] Y. Shi, H. Liu, W. Zhang, Y. Wei, and J. Dong, "Research on three-dimensional localization algorithm for WSN based on RSSI," in *Advances in Intelligent Systems and Computing*, vol. 928, pp. 1048–1055, Springer Nature, Switzerland, 2020.
- [11] A. Stanoev, S. Filiposka, V. In, and L. Kocarev, "Cooperative method for wireless sensor network localization," *Ad Hoc Networks*, vol. 40, pp. 61–72, 2016.
- [12] Z. Liu, X. Feng, J. J. Zhang, T. Li, and Y. L. Wang, "An improved GPSR algorithm based on energy gradient and APIT grid," *Journal of Sensors*, vol. 2016, Article ID 2519714, 7 pages, 2016.
- [13] G. S. Li, S. S. Zhao, J. H. Wu, C. L. Li, and Y. Liu, "DV-hop localization algorithm based on minimum mean square error in Internet of things," *Procedia Computer Science*, vol. 147, pp. 458–462, 2019.
- [14] N. Dragoş and B. Nath, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1/4, pp. 267–280, 2003.
- [15] B. F. Gumaida and J. Luo, "Novel localization algorithm for wireless sensor network based on intelligent water drops," *Wireless Networks*, vol. 25, no. 2, pp. 597–609, 2019.
- [16] G. Venter and J. S. Sobieski, "Particle swarm optimization," *AIAA Journal*, vol. 41, no. 8, pp. 1583–1589, 2003.
- [17] L. M. Schmitt, "Theory of genetic algorithms," *Theoretical Computer Science*, vol. 259, no. 1-2, pp. 1–61, 2001.
- [18] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [19] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [20] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.
- [21] J. K. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020.
- [22] J. Mohit, S. Vijander, and R. Asha, "A novel nature-inspired algorithm for optimization: squirrel search algorithm," *Swarm and Evolutionary Computation*, vol. 44, pp. 148–175, 2019.
- [23] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Computing*, vol. 23, no. 3, pp. 715–734, 2019.
- [24] V. Kanwar and A. Kumar, "DV-hop localization methods for displaced sensor nodes in wireless sensor network using PSO," *Wireless Networks*, vol. 27, no. 1, pp. 91–102, 2021.
- [25] S. P. Singh and S. C. Sharma, "Implementation of a PSO based improved localization algorithm for wireless sensor networks," *IETE Journal of Research*, vol. 65, no. 4, pp. 502–514, 2019.
- [26] S. P. Singh and S. C. Sharma, "A PSO based improved localization algorithm for wireless sensor networks," *Wireless Personal Communications*, vol. 98, no. 1, pp. 487–503, 2018.
- [27] H. Zhang, D. Y. Long, T. Qin, X. Wang, and J. Yang, "Improved DV-hop localization algorithm based on continuous hop," *Journal of Chinese Computer Systems*, vol. 42, no. 11, pp. 2388–2393, 2021.
- [28] W. S. Fang, Z. Wang, W. W. Wu, and Z. D. Hu, "PSO localization algorithm based on node density weighting and distance correction," *Transducer and Microsystem Technologies*, vol. 39, no. 7, pp. 127–133, 2020.
- [29] X. J. Cai, P. H. Wang, Z. H. Cui, W. Zhang, and J. Chen, "Weight convergence analysis of DV-hop localization algorithm with GA," *Soft Computing*, vol. 24, no. 23, pp. 18249–18258, 2020.
- [30] V. Kanwar and A. Kumar, "DV-hop-based range-free localization algorithm for wireless sensor network using runner-root optimization," *The Journal of Supercomputing*, vol. 77, no. 3, pp. 3044–3061, 2021.
- [31] T. K. Mohanta and D. K. Das, "Class topper optimization based improved localization algorithm in wireless sensor network," *Wireless Personal Communications*, vol. 119, no. 4, pp. 3319–3338, 2021.
- [32] S. S. Jacob, K. Muthumayil, M. Kavitha et al., "A modified search and rescue optimization based node localization technique in WSN," *Computers, Materials Continua*, vol. 70, no. 1, pp. 1229–1245, 2022.
- [33] P. Z. Gou, X. Z. Liu, M. Y. Sun, and B. He, "Three-dimensional localization algorithm based on hop correction and lion swarm optimization in WSNs," *Computer Engineering and Science*, vol. 43, no. 8, pp. 1405–1412, 2021.
- [34] A. Kaushik, D. K. Lobiyal, and S. Kumar, "Improved 3-dimensional DV-hop localization algorithm based on information of nearby nodes," *Wireless Networks*, vol. 27, no. 3, pp. 1801–1819, 2021.
- [35] V. Kanwar and A. Kumar, "Range free localization for three dimensional wireless sensor networks using multi objective particle swarm optimization," *Wireless Personal Communications*, vol. 117, no. 2, pp. 901–921, 2021.
- [36] J. Cheng, Y. L. Dong, J. X. Chen, and Z. H. Liu, "An improved three-dimensional DV-hop algorithm with continuous hop values," *Acta Electronica Sinica*, vol. 48, no. 11, pp. 2122–2130, 2020.
- [37] X. Cai, P. Wang, L. Du, Z. Cui, W. Zhang, and J. Chen, "Multi-objective three-dimensional DV-hop localization algorithm with NSGA-II," *IEEE Sensors Journal*, vol. 19, no. 21, pp. 10003–10015, 2019.
- [38] G. Sharma and A. Kumar, "Improved range-free localization for three-dimensional wireless sensor networks using genetic algorithm," *Computers and Electrical Engineering*, vol. 72, pp. 808–827, 2018.
- [39] S. Dong, X. G. Zhang, and W. G. Zhou, "A security localization algorithm based on DV-hop against Sybil attack in wireless sensor networks," *Journal of Electrical Engineering and Technology*, vol. 15, no. 2, pp. 919–926, 2020.

- [40] H. T. Ouyang, Y. X. Qiu, and D. Zhu, "A multi-strategy improved sparrow search algorithm," *Journal of Physics: Conference Series*, vol. 1848, no. 1, article 012042, 2021.
- [41] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.
- [42] T. Li, C. Wang, and Q. Na, "Research on DV-hop improved algorithm based on dual communication radius," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, 2020.
- [43] W. X. Xue and W. N. Qiu, "Improved Wi-Fi RSSI measurement for indoor localization," *IEEE Sensors Journal*, vol. 17, no. 7, pp. 2224–2230, 2017.
- [44] O. Cheikhrouhou, G. M. Bhatti, and R. Alroobaea, "A hybrid DV-hop algorithm using RSSI for localization in large-scale wireless sensor networks," *Sensors*, vol. 18, no. 5, pp. 1469–1482, 2018.
- [45] R. Mahapatra and N. Shet, "Localization based on RSSI exploiting gaussian and averaging filter in wireless sensor network," *Arabian Journal for Science and Engineering*, vol. 43, no. 8, pp. 4145–4159, 2018.
- [46] Q. Zhou and L. Leydesdorff, "The normalization of occurrence and co-occurrence matrices in bibliometrics using cosine similarities and Ochiai coefficients," *Journal of the Association for Information Science and Technology*, vol. 67, no. 11, pp. 2805–2814, 2016.
- [47] C. T. Ouyang, D. L. Zhu, and F. Q. Wang, "A learning sparrow search algorithm," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 3946958, 23 pages, 2021.