Peer-to-Peer Networking and Applications
https://doi.org/10.1007/s12083-024-01710-1

An IMOA DV-Hop localization algorithm in WSN based on hop count and hop distance correction

Xiuwu Yu[1] · Wei Peng[1] · Zixiang Zhou[1] · Ke Zhang[1] · Yong Liu[1]

Received: 21 November 2023 / Accepted: 8 April 2024

Abstract

Wireless sensor networks (WSNs) have been widely used in environmental monitoring due to their low cost advantages. In WSNs monitoring, the location information is significant, because data collected by sensor nodes is valuable only if the locations of nodes are known. DV-Hop algorithm is a popular localization algorithm in WSNs monitoring. However, DV-hop has low localization accuracy due to its imperfect hop count, hop distance and location calculation mechanism. Therefore, in order to improve its localization accuracy, we improve the three stages of DV-hop respectively: Firstly, the anchor node broadcasts in three types of communication radius to reduce hop count error. Secondly, we utilize local average hop distance to reduce the hop distance calculation error. Finally, we use the heuristic algorithm MOA to calculate node positions. Meanwhile, we utilize the good point set, t-distribution and Levy flight to improve the global optimization ability of MOA. In simulation experiments, we use Matlab2018a to verify algorithm performance. The simulation results show that the proposed algorithm outperforms the comparison algorithm in different communication radius, number of anchor nodes, and total number of nodes. It performs optimally in both localization efficiency and accuracy, and has better robustness.

Keywords Wireless sensor network · DV-Hop · Three communication radius · Local average hop distance · Mayfly optimization algorithm·

1 Introduction

Wireless Sensor Networks (WSNs) are intelligent private networks composed of hundreds or thousands of low-power,multi-functional, and cost-effective sensor nodes deployed in a monitoring region. These nodes utilize self-organizing wireless communication manner to collaboratively perform specific functions through multi-hop routing. They collect and transmit multiple types of data to the observer in real-time [1, 2]. Owing to these advantages, WSNs have found a wide range of applications in industries such as environmental monitoring, military operations, traffic management, object tracking, and safety production [3, 4]. In WSNs, it is essential for sensor nodes to have accurate location information. This is because the data collected by the sensor nodes is valuable only if the locations of nodes are known. As a result, localization has become a significant research area in WSNs [5].

Wireless sensor networks consist of two types of sensor nodes: anchor nodes and unknown nodes. Anchor nodes are equipped with a Global Positioning System (GPS) module and can determine their own location information. However, only a small percentage of sensor nodes have GPS due to its high cost. The remaining nodes, referred to as unknown nodes, rely on anchor nodes and localization algorithms to determine their locations. Over the past two decades, various localization techniques have been developed to accurately determine the locations of unknown nodes. These techniques can be broadly categorized into two types: range-based algorithms and range-free algorithms [6, 7].

Range-based localization techniques utilize measurements such as Received Signal Strength Indicator (RSSI) [8], Time of Arrival (ToA) [9], Time Difference of Arrival (TDoA) [10], and Angle of Arrival (AoA) [11] to determine the location of nodes based on the distances or angles

This article is part of the Topical Collection: *1- Track on Networking and Applications*
Guest Editor: Vojislav B. Misic

✉ Wei Peng
2977681356@qq.com

[1] School of Resource & Environment and Safety Engineering, University of South China, Hengyang 421001, China

Published online: 27 May 2024

between them. However, implementing these techniques requires additional hardware facilities, which leads to a significant increase in overall costs. On the other hand, range-free localization algorithms do not rely on distance or angle measurements but instead utilize the connectivity between nodes. Some common range-free localization algorithms include Distance Vector-Hop (DV-Hop) [12], Approximate Point in Triangle (APIT) [13], Centroid [14], Amorphous [15], etc. These range-free algorithms do not require extra hardware and consume less energy. However, it is important to note that they generally exhibit lower accuracy compared to range-based techniques [16].

In WSNs, accurately localizing sensor nodes is a significant challenge. Therefore, we conducted a study to enhance the DV-Hop algorithm for WSN localization. Our primary objective is to reduce localization errors by modifying the original DV-Hop algorithm's calculation model for hop count and local hop distance. Moreover, we integrated the improved Mayfly Optimization Algorithm (MOA) into the third stage of DV-Hop to optimize the localization process and improve its accuracy. The main contributions of our work can be summarized as follows:

1. The anchor nodes broadcast with three different communication radii to improve the calculation of minimum hop count value between nodes.
2. The error weight of anchor nodes within a two-hop range of the unknown node is incorporated to improve the average hop distance of unknown nodes.
3. We use anchor nodes within two hops of unknown nodes along with the improved MOA to estimate unknown node coordinates.
4. We conducted simulation experiments to demonstrate the advantages of our proposed algorithm in terms of convergence rate and localization accuracy compared to DV-Hop, GADV-Hop [25], PSODV-Hop [26], and SSADV-Hop [27].

The remaining sections of this work are structured as follows: In Section 2, we provide a thorough literature review and introduce the DV-Hop algorithm and Mayfly Optimization Algorithm. In Section 3, we outline the enhancements made to the DV-Hop algorithm and the MOA, and we present our novel DV-Hop algorithm. In Section 4, we present the results of a simulation experiment and provide a detailed analysis of the algorithm's performance. Finally, we conclude our work in Section 5, summarizing the findings and discussing the implications of our research.

2 Related works

2.1 Literature review

DV-Hop is a popular range-free localization algorithm used in WSNs for estimating node locations based on network connectivity. However, one of its main limitations is its relatively low accuracy in localization. To overcome this challenge, researchers have developed enhanced versions of DV-Hop that integrate various techniques aimed at improving its performance.

For example, In [17], the authors proposed a modification to the DV-Hop algorithm, where the unknown nodes' average hop size was replaced with the mean of its neighbor anchors' average hop size. This modification has the potential to enhance accuracy. In [18], Kumar and Lobiyal introduced a new DV-Hop algorithm that included an improvement term in the calculation of anchor nodes' hop size. They further utilized unconstrained optimization techniques to minimize localization error. Literature [19] suggested an advanced DV-Hop localization algorithm aimed at reducing the inherent error in estimating the distance between anchor nodes and unknown nodes. Hu and Li [20] proposed a modified DV-Hop algorithm that improved the average hop distance of unknown nodes by considering the weighted average hop distances of anchor nodes. Moreover, researchers have explored recursive variants of DV-Hop to enhance efficiency. Messous et al. [21] utilized RSSI-based distance calculation between anchor nodes and their one-hop neighboring sensor nodes to improve DV-Hop. Once a sensor node is located, it acts as an anchor to locate other unknown nodes. Similarly, Xiao et al. [22] developed a weighted DV-Hop method based on RSSI, employing unconstrained optimization for position calculation of unknown nodes. Kaur et al. [23] presented an enhanced approach that combines the weighted centroid and DV-Hop algorithm, resulting in an improved weighted centroid DV-Hop localization approach. Messous et al. [24] proposed a novel improved recursive DV-Hop algorithm, leveraging recursive position computation and optimized average hop distance to enhance the DV-Hop algorithm. These algorithms have demonstrated advancements over the original DV-Hop. However, the problem of large errors in localization still persists.

Recently, researchers have attempted to improve the localization accuracy of the DV-Hop algorithm by incorporating heuristic algorithms to overcome its limitations. In [25], Ouyang et al. introduced an upgraded DV-Hop algorithm based on an adaptive genetic algorithm (GA) to calculate the coordinates of unknown nodes. However, it did not result in any significant improvements in the average hop size and hop count. Chen and Zhang [26] revised the average hop distance and integrated particle swarm optimization (PSO) to optimize the location determined by the 2D hyperbolic algorithm. Lei et al. [27] proposed an improved Sparrow Search Algorithm (SSA) DV-Hop algorithm that used a double communication radius mechanism to modify the minimum hop count and utilized SSA for node position estimation. Cui et al. [28] introduced a new approach that combined an oriented cuckoo search algorithm with DV-Hop to reduce localization error. Similarly, Li et al. [29] presented an enhanced cat swarm optimization (CSO) algorithm that was integrated into DV-Hop to minimize the localization error of unknown nodes. Building on these efforts, Chen et al. [30] proposed a chicken swarm optimization-based approach to reduce localization error. Additionally, Shi et al. [31] suggested an improved hybrid algorithm that combined PSO and simulated annealing (SA) to reduce localization error in DV-Hop. However, it is important to note that heuristic algorithms have limitations, such as slow convergence rates and susceptibility to local optima, which can lead to inaccurate localization results. To address these challenges and further improve the accuracy of DV-Hop localization, a novel algorithm called IMOADV-Hop was proposed by combining the strengths of the improved DV-Hop and the MOA. Simulation results have shown that IMOADV-Hop outperforms the original DV-Hop and other improved DV-Hop algorithms in terms of localization accuracy, convergence rate, and energy efficiency.

2.2 DV-Hop algorithm

The DV-Hop localization algorithm comprises three phases:

Phase 1: Anchor nodes initiate a distributed process to acquire the minimum hop count. Each anchor node broadcasts a packet consisting of its coordinates, ID, and hop count (initially set to 0). Unknown nodes within the communication radius of the anchor nodes receive the packet and record the information. They then transmit the packet to their neighboring nodes while increasing the hop count by one. If multiple hop count information is received from the same anchor node, only the minimum hop count information is retained. This process continues iteratively until all unknown nodes receive and processe the packets from all anchor nodes. Ultimately, all unknown nodes obtain the minimum hop count to each anchor node.

Phase 2: Calculate the distance between the unknown node and all anchor nodes.

Firstly,each anchor node's average hop distance is calculated by Eq. (1).

$$Hopsize_i = \frac{\sum_{i \neq j} \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}}{\sum_{i \neq j} h_{ij}} \tag{1}$$

where h_{ij} represents the minimum hop count between anchor node i and j. The coordinates of anchor node i and j are denoted by (x_i,y_i) and (x_j,y_j) respectively. Additionally, $Hopsize_i$ refers to the average hop distance of anchor node i.

Once an anchor node computes its own average hop distance, it shares this information with the rest of the network via broadcasting. After that, an unknown node u adopts the first received average hop distance as its own, and then proceeds to calulate its distance to each anchor node using Eq. (2). This equation is derived from the minimum hop count information recorded in phase 1.

$$d_{iu} = Hopsize_i \times h_{iu} \tag{2}$$

Phase 3: In this Phase, the coordinates of the unknown node are determined by estimating the distance between the unknown node and each anchor node, based on the information obtained in phases 1 and 2. If the network consists of at least three anchor nodes ($m \geq 3$) anchor nodes, the distance from the unknown node to each anchor node is represented as d_1, d_2, …, d_m, respectively. The coordinates of the unknown node can be calculated using Eq. (3).

$$\begin{cases} \left(x - x_1\right)^2 + \left(y - y_1\right)^2 = d_1^2 \\ \left(x - x_2\right)^2 + \left(y - y_2\right)^2 = d_2^2 \\ \vdots \\ \left(x - x_m\right)^2 + \left(y - y_m\right)^2 = d_m^2 \end{cases} \tag{3}$$

The coordinates of each anchor node are represented as (x_1,y_1), (x_2,y_2), …, (x_m,y_m), where (x_1,y_1) corresponds to anchor node 1, (x_2,y_2) corresponds to anchor node 2, and so on. The coordinates (x, y) denote the coordinates of the unknown node.

2.3 Mayfly optimization algorithm

The Mayfly Optimization Algorithm is inspired by the social behavior of mayflies [32]. It uses the state updates and mating behavior observed in mayflies to find the optimal solution within the given solution space. The algorithm follows the following main steps.

2.3.1 Regeneration of male mayflies

Male mayflies display movement patterns that resemble to particles in PSO. Their new locations are determined by their current positions and velocities. The position of the *i*-th mayfly in the *t*-th iteration is represented as X_i^t, and its position update formula is given by:

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{4}$$

where, V_i is the velocity of mayfly, and the updating formula of speed is:

$$V_i^{t+1} = \begin{cases} gV_i^t + a_1 e^{-\beta r_p^2}\left(P - X_i^t\right) + a_2 e^{-\beta r_g^2}\left(G - X_i^t\right), \; f\left(Y_i\right) > f(X_i) \\ gV_i^t + dr, \; others \end{cases} \tag{5}$$

where g is the inertia weight coefficient; a_1 and a_2 are attraction coefficients; P and G are the historical and global optimal locations, respectively; β is the visibility coefficient that controls the visibility range of mayfly; r_p and r_g represent the Euclidean distance between the current mayfly's position and P, G, respectively; d is dance coefficient, with $d = d_0\delta^t$, where d_0 is the initial value of mating; t is the number of iterations; r and δ are random numbers in the range (0,1).

2.3.2 Regeneration of female mayflies

In the MOA, each female mayfly is paired with a male mate in a sequential manner. The position of the *i*-th female mayfly in the *t*-th iteration is represented by Y_i^t. The position update formula for the female mayflies can be expressed as follows:

$$Y_i^{t+1} = Y_i^t + V_i^{t+1} \tag{6}$$

The speed of the female mayfly is updated by:

$$V_{ij}^{t+1} = \begin{cases} gV_{ij}^t + a_2 e^{-\beta r_m^2}\left(X_{ij}^t - Y_{ij}^t\right) \; f\left(Y_i\right) > f\left(X_i\right) \\ gV_{ij}^t + fl \cdot r \qquad\qquad\qquad others \end{cases} \tag{7}$$

where g represents the inertia weight coefficient; a_2 is the attraction coefficient; β is the visibility coefficient; r_m is the distance between male and female mayflies; fl is the random walk coefficient; r is a random number in the interval of (0,1).

2.3.3 Mayfly mating

In the MOA, female mayflies choose their male mates based on their quality during the mating phase. The selection process is done sequentially, starting with the best female mayfly being paired with the best male mayfly followed by the second-best female being paired with the second-best male, and so on. This sequential pairing continues until all female mayflies have successfully paired up with male mayflies.

$$off_1 = L \cdot male + (1 - L) \cdot female \tag{8}$$

$$off_2 = (1 - L) \cdot male + L \cdot female \tag{9}$$

The male and female mayflies' offspring are labeled as off_1 and off_2, respectively. L is a random number in the range of [-1,1], and the terms *male* and *female* represent the mayflies.

2.3.4 Fitness function

When estimating the position of unknown nodes using MOA, the fitness function of each mayfly individual is crucial. The fitness function is defined as follows:

$$f(x, y) = \sum_{k=1}^{n} \left| \sqrt{\left(x - x_k\right)^2 + \left(y - y_k\right)^2} - d_k \right| \tag{10}$$

where (x,y) represents the estimated coordinate of the unknown node, while (x_k,y_k) represents the coordinate of an anchor node, and d_k stands for the estimated distance between the unknown node and the k-th anchor node. The fitness function is considered optimal when it reaches its minimum value, indicating that the estimated coordinates of the unknown node are closest to the actual coordinates.

3 Proposed IMOADV-Hop algorithm

3.1 Minimum hop count optimization

The DV-Hop algorithm is a popular method for localization in WSNs. However, it has a limitation related to the calculation of hop counts. When an unknown node is located within the communication radius of an anchor node, the hop count between them is recorded as one. This can result in significant errors in localization, particularly when multiple unknown nodes use the same minimum hop count to estimate their distance from the anchor node. To address this shortcoming, a new approach is proposed in this paper, which aims to enhance the accuracy of the minimum hop count in the DV-Hop algorithm. The proposed method utilizes additional topology information to refine the hop count calculation process. By incorporating this extra information, the algorithm can achieve a more precise estimation of the node's distance to the anchor node.

Figure 1 illustrates the improved minimum hop count model proposed in this paper for localization in WSNs. To achieve a more precise minimum hop count, the proposed

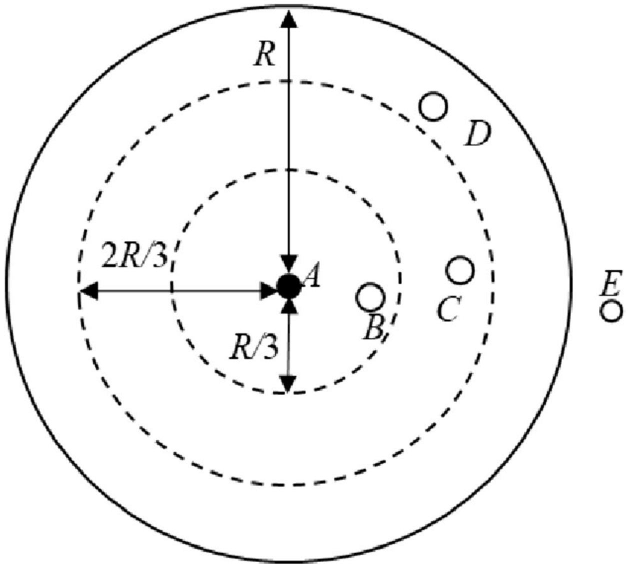


Fig. 1 Minimum hop count calculation model

approach utilizes a multi-hop broadcasting scheme. The anchor node (Node *A*) is surrounded by several unknown nodes (Nodes *B*, *C*, *D*, and *E*) located within the communication radius *R* of the anchor node. The process begins with the anchor node A initiating the broadcasting with a radius of *R*/3. At this stage, only node *B* is within range to receive the broadcast. Node *B* records the hop count as 1/3.

In the second broadcast, anchor node *A* increases the radius to 2*R*/3. Both nodes *B* and *C* receive the broadcast. However, node *B* discards the second broadcast and continues to forward the minimum hop count value of 1/3 to its neighboring nodes. In contrast, node *C* records the hop count as 2/3 and forwards it to its neighbors.

In another round of broadcasting, anchor node *A* increases the radius to *R*. Nodes *B* and *C* discard the broadcast information, while node *D* receives it and records the hop count as 1. It then continues to forward the broadcast information. Finally, node *E* receives information from nodes *B*, *C*, and *D*, and it selects the smallest hop count value, which is 1/3 from node *B*. It then adds 1 to it, resulting in a minimum hop count of 4/3 to the anchor node.

By employing this approach, each unknown node selects the smallest hop count value to the anchor node, effectively reducing the localization error that can arise when multiple unknown nodes utilize the same minimum hop count value.

3.2 Average hop distance optimization

In WSNs, it is important to accurately calculate the distance between unknown nodes and anchor nodes. To achieve this, it's crucial to consider the local topological relationships within the network. In this paper, we propose a new method that builds upon the concept of local average hop distance (LAHD) [20]. Our approach takes into account the node distribution surrounding the unknown node and assigns weights for the LAHD values of anchor nodes within a two-hop range. This allows us to estimate the node's distance to the anchor node more accurately. The process of LAHD optimization is outlined as follows:

Firstly, all the anchor nodes' LAHD within two hops of unknown node is:

$$LAHD_i = \frac{\sum\limits_{i \neq j} \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}}{\sum\limits_{i \neq j} h_{ij}} \tag{11}$$

where *i* is the any anchor node within two hops from the unknown node; $LAHD_i$ is the local average hop distance of anchor node *i*, *j* is any anchor node within two hops from anchor node *i*; h_{ij} is the hop count between *i* and *j*.

The actual distance d_{ij} between anchor node *i* and *j* is:

$$d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{12}$$

The estimated distance d_{ij} between anchor node *i* and *j* is:

$$d'_{ij} = LAHD_i \times h_{ij} \tag{13}$$

Then, the mean distance error e_i of anchor node *i* is:

$$e_i = \frac{\sum\limits_{i \neq j} \left| d'_{ij} - d_{ij} \right|}{\sum\limits_{i \neq j} h_{ij}} \tag{14}$$

The smaller e_i is, the better the anchor node can accurately represent the distribution of nodes close to the unknown node. To increase the influence of the anchor node in this situation, it is necessary to assign it a higher weight. The weight can be normalized as:

$$w_i = \frac{1/e_i}{\sum\limits_{i-1}^{n} 1/e_i} \tag{15}$$

where w_i is the LAHD's weight of the anchor node *i*, then the formula for calculating the LAHD of unknown node *u* is as follows:

$$LAHD_u = \sum_{i=1}^{n} w_i \cdot LAHD_i \tag{16}$$

Therefore, the calculated distance from unknown node *u* to anchor node *i* is:

$$d_{ui} = LAHD_u \cdot h_{ui} \tag{17}$$

where d_{ui} represents the estimated distance between unknown node u and anchor node i, and h_{ui} represents the optimized minimum hop count between unknown node u and anchor node i.

After obtaining the coordinates and distances to multiple anchor nodes within a two-hop range, the unknown node u can use the improved MOA method outlined in Section 3.3 for position estimation.

3.3 Improved mayfly optimization algorithm

3.3.1 Good point set

The original MOA is often criticized for its slow convergence rate. This can be attributed to its random population initialization strategy. This strategy may lead to poor diversity, quality, and uniform traversal of the solution space. To overcome this limitation, we propose incorporating a good point set [33] with excellent ergodicity as an initialization method for the mayfly population in the MOA.

Assuming that V_D is a D-dimensional unit cube, r belongs to V_D, and $P_\mathrm{n}(k)$ fulfills the following condition:

$$P_n(k) = \left\{ \left(\left\{ r_1^{(n)} \cdot k \right\}, \left\{ r_2^{(n)} \cdot k \right\}, \ldots, \left\{ r_D^{(n)} \cdot k \right\} \right),\ 1 \leqslant k \leqslant n \right\} \tag{18}$$

If the criteria showed in Eq. (19) is met by the deviation of $P_n(k)$, it can be referred to as a good point set.

$$\varphi(n) \leqslant C(r, \varepsilon) n^{-1+\varepsilon} \tag{19}$$

where $\mathrm{C}(r,\varepsilon)n^{-1+\varepsilon}$ is a constant that only depends on r and ε; n denotes the number of point in the set, and r is defined as $r = \{2\cos(2\pi k/p)\}$,where p is the smallest prime number that satisfies $(p\text{-}3)/2 \geq n$. After that, we can map the points in the good point set to the search space.

$$x_i(j) = \left\{ r_j^{(i)} \cdot k \right\} \left(ub_j - lb_j \right) + lb_j \tag{20}$$

where, ub_j and lb_j represent the upper and lower bounds of the j-th dimension of the search space.

In our study, we conducted a performance comparison between two population initialization methods, the good point set method and the random method, within a two-dimensional search space ranging from 0 to 50. Figure 2 presents a visual representation of the results of this comparison, highlighting the differences between the two methods. The results clearly show that the initial population generated by the good point set method displays a more uniform distribution across the search space compared to the random method. This indicates a higher level of diversity in the initial population.

3.3.2 Adaptive Levy flight

In the MOA velocity update, the coefficient of inertial weight (g) is usually reduced linearly with time, while the attraction coefficients (a_1 and a_2) remain constant. However, this linear approach may not be effective in balancing local and global search, which can lead to slow convergence of the algorithm. To improve the convergence rate of the algorithm, we propose an adaptive strategy that involves dynamically adjusting the inertia weight coefficient using a nonlinear approach.

$$g = \begin{cases} g_{\min} + \frac{(g_{\max} - g_{\min})(f_i - f_{\min})}{f_{avg} - f_{\min}} & , f_i \leqslant f_{avg} \\ g_{\max} & , f_i > f_{avg} \end{cases} \tag{21}$$

where $g_{\max}$ and $g_{\min}$ are the maximum and minimum inertia weight coefficients respectively; f_i is the fitness value of mayfly i; $f_{\min}$ and f_{avg} are the minimum and average fitness value, respectively. The impact of fitness on the behavior of mayflies can be observed from the Eq. (12), which indicates that the weight of the mayfly is maximal when its fitness is greater than the average fitness. This increases its activity and enables it to explore the search space more effectively. On the other hand, when the fitness value of a mayfly is below the average, the weight of its speed is reduced. This prompts the mayfly to prioritize movement towards more advantageous positions in the search space.

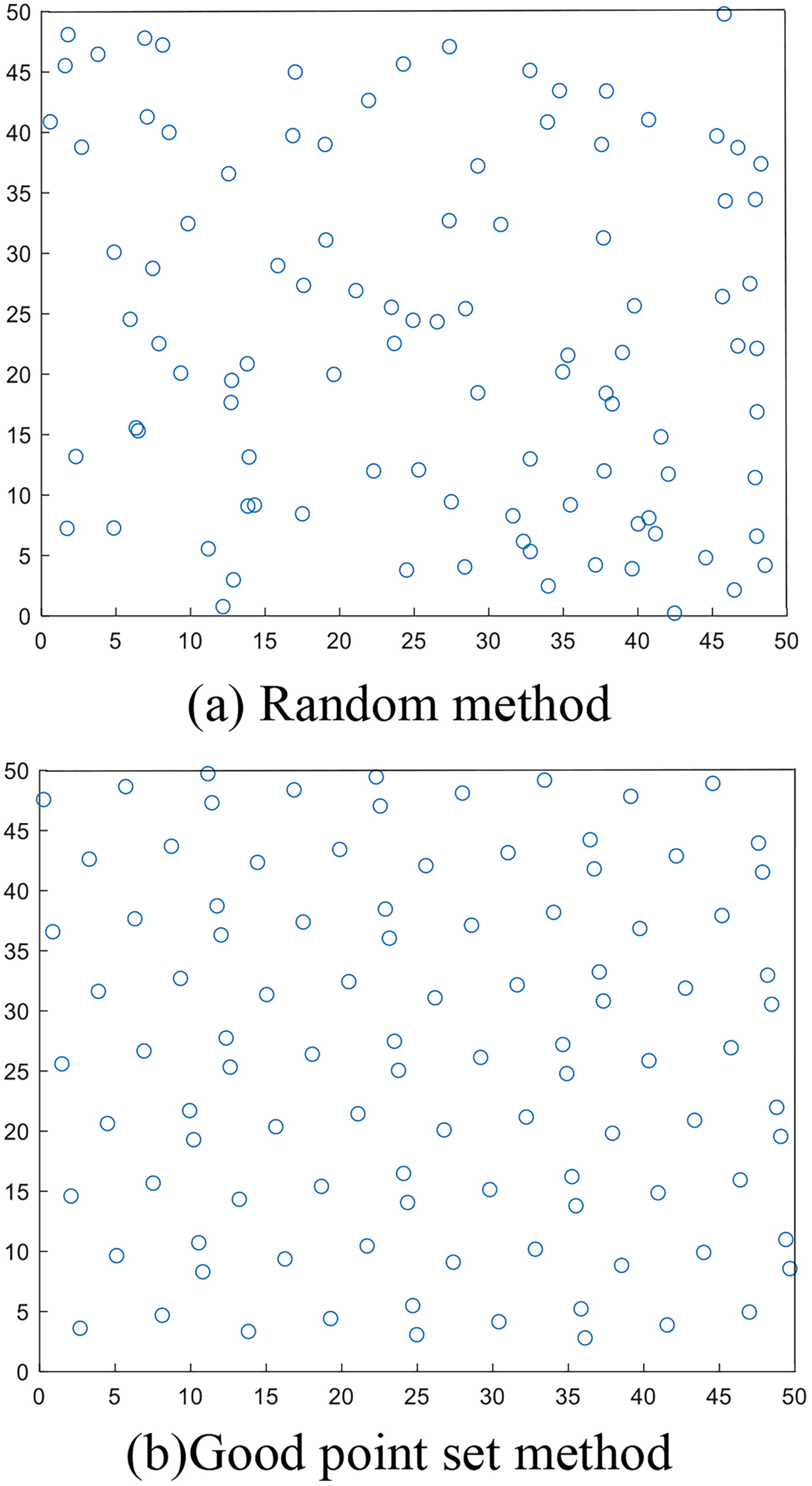
In the proposed solution, a dynamic attraction coefficient is incorporated along with the dynamic adjustment of the inertia weight coefficient to achieve a balance between local and global search capabilities in the algorithm.

$$\begin{cases} a_1 = 2.5 - 2\left(\frac{t}{t_{\max}}\right)^2 \\ a_2 = 0.5 - 2\left(\frac{t}{t_{\max}}\right)^2 \end{cases} \tag{22}$$

where t represents the current iteration number, and $t_{\max}$ represents the maximum iteration number. It is noteworthy that the adaptive attraction coefficient dynamically changes as the iteration number increases. This dynamic adjustment allows the algorithm to exhibit strong local search ability during the early stages of optimization while transitioning to a strong global search ability in the later stages.

In the algorithm, the local optimization ability is primarily determined by the position update of the mayflies, while the global optimization ability is mainly influenced by the mating behavior of mayflies. The velocity update formula used by the mayflies indicates that the algorithm inherently has a strong local search ability but a relatively weaker global search ability. To overcome this limitation and enhance the algorithm' global optimization ability, the Levy flight strategy is employed.

Fig. 2 Comparison of initial population method



(a) Random method

(b)Good point set method

Levy flight [34] is a type of stochastic search mechanism that takes inspiration from the flight patterns of certain animals, including birds and insects. It involves introducing random and long-range movements into the search process, which helps the algorithm explore the search space more extensively and potentially discover better global solutions. The length of each step in Levy flight is determined by formula (23).

$$S = \frac{\mu}{|\upsilon|^{1/\beta}} \tag{23}$$

where $\mu \sim N(0,\sigma^2)$; $\upsilon \sim N(0,\sigma^2)$.

$$\sigma = \left\{ \frac{\Gamma(1+\beta)\cdot sin(\pi\beta/2)}{\Gamma\left[(1+\beta)/2\right]\cdot 2^{(\beta-1)/2}\cdot\beta} \right\}^{1/\beta} \tag{24}$$

where Γ is the gamma function, $\Gamma(n) = (n-1)!$, and β takes 1.5.

The velocity update based on adaptive weight coefficients and Levy flight is:

$$\begin{aligned} V_i^{t+1} = & g\cdot V_i^t + a_1 e^{-\beta r_p^2}(P - X_i^t) \\ & + a_2 e^{-\beta r_g^2}\left(G - X_i^t\right) + S\cdot(G - r) \end{aligned} \tag{25}$$

where g is the inertia weight in formula (21); a_1 and a_2 are the attraction coefficients in formula (22); S is the Levy flight step size in formula (23); and r is a random number in (0,1).

3.3.3 Adaptive *t*-distribution variation

The use of an adaptive *t*-distribution mutation operator in the MOA algorithm is aimed at improving its ability to escape from local optima and enhance overall optimization performance [35]. This operator introduces perturbations and mutations to the current best solution, preventing the algorithm from getting stuck in local optima.

The *t*-distribution, also known as the student distribution, is a probability distribution that depends on the degrees of freedom, represented as *n*. The probability density function (PDF) of the t-distribution can be written as follows:

$$f(x,n) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}\,\Gamma\left(\frac{n}{2}\right)}\left(1+\frac{x^2}{n}\right)^{\frac{-(n+1)}{2}} \tag{26}$$

To improve the algorithm's ability to avoid getting stuck in local optima, the *t*-distribution mutation operator can be utilized to mutate the current optimal individual. The following method can be used to apply this operator:

$$X_{new} = X_{best} + X_{best}\cdot t(iter) \tag{27}$$

where *iter* represents the current number of iterations, and *t(iter)* is a random number generated from the *t*-distribution with *iter* degrees of freedom. At the beginning of the algorithm, the *t*-distribution behaves like the Cauchy distribution, which is known for its strong disturbance capability. However, as the number of iterations increases, the *t*-distribution gradually approaches a Gaussian distribution, which results in a weaker disturbance capability. This adaptive adjustment of the algorithm's local optimization ability allows it to effectively escape from local optima.

3.4 IMOADV-Hop algorithm process

Step1: After the nodes deployment in the monitoring area, the anchor node periodically broadcasts its ID and coordinates within the communication radius.

Step2: The unknown node within a two-hop range receives and stores the broadcasted information from the anchor nodes, including their IDs and coordinates. The

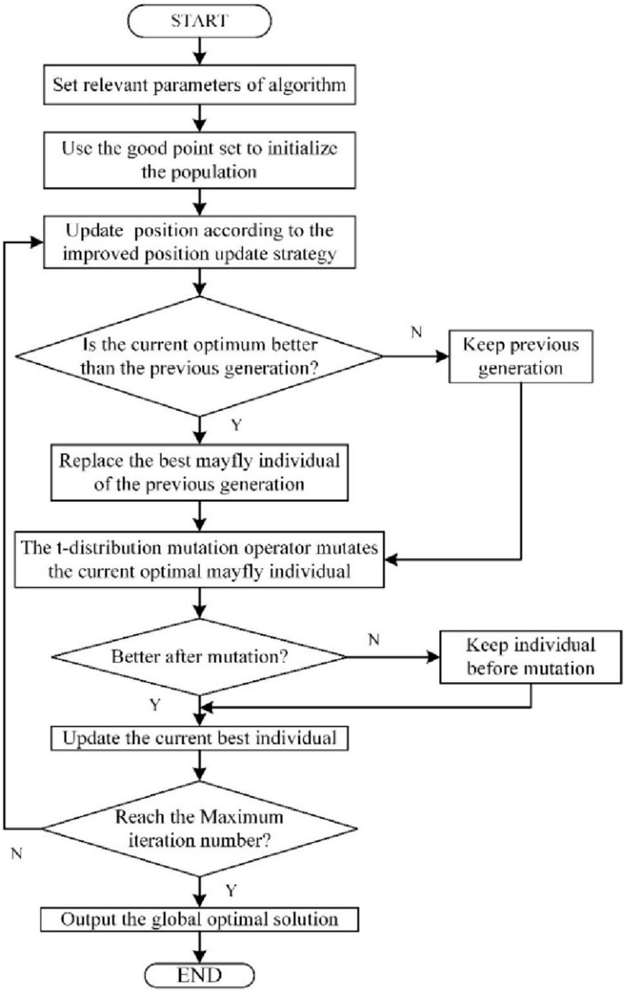

START
Set relevant parameters of algorithm
Use the good point set to initialize the population
Update position according to the improved position update strategy
Is the current optimum better than the previous generation?
N
Keep previous generation
Y
Replace the best mayfly individual of the previous generation
The t-distribution mutation operator mutates the current optimal mayfly individual
Better after mutation?
N
Keep individual before mutation
Y
Update the current best individual
Reach the Maximum iteration number?
N
Y
Output the global optimal solution
END

Fig. 3 The algorithm flow of IMOA

unknown node then uses an enhanced mechanism involving hop count and average hop distance to calculate the distance to each anchor node.

Step3: The unknown node employs the improved MOA to determine its own coordinates.

The algorithm flow is shown in Fig. 3.

4 Simulation results and analysis

To assess the effectiveness of the proposed algorithm, we compared it to four other algorithms: DV-Hop, GADV-Hop, PSODV-Hop, and SSADV-Hop. The evaluation was carried out using the MATLAB2018a simulation platform. The simulation environment is a square of 100m × 100m area. The performance of the algorithm is measured using the normalized average positioning error in Eq. (28).

$$E = \frac{1}{nR}\sum_{1}^{n}\sqrt{(x-\hat{x})^2 + (y-\hat{y})^2} \tag{28}$$

where n is the total number of unknown nodes, R is the communication radius, (x,y) is the real coordinates of unknown nodes, $(\hat{x}, \hat{y})$is the coordinates of unknown nodes calculated by the algorithm.

4.1 Convergence simulation

To assess the performance of the improved Mayfly Optimization Algorithm (IMOA), we utilize Eq. (10) as the objective function for comparative analysis. Figure 4 presents the relationship between iteration time and the average fitness of IMFO, PSO, GA, and SSA. The results demonstrate that IMFO achieves convergence to 1.95 in approximately 5.5 s, while GA converges to 2.75 in approximately 8 s, PSO converges to 2.09 in approximately 9 s, and SSA converges to 1.95 in approximately 10.5 s. These findings suggest that IMFO exhibits higher convergence accuracy and rate, positioning it as a promising approach for enhancing localization accuracy.

4.2 Unknown node localization error

In this simulation, the total number of nodes present is 120. These nodes are randomly distributed across the simulation area, which includes 30 anchor nodes. The communication radius is set to 30 m. The distribution of nodes is shown in Fig. 5. The localization results obtained from the five algorithms are presented in Fig. 6.

In the figures, the anchor nodes are depicted with red stars, the real positions of the unknown nodes are indicated with black circles, and the estimated positions of the unknown nodes are represented with blue circles. The estimated positions of the unknown nodes are connected to their real positions with blue lines.

Based on the presented figures, it can be concluded that the IMOADV-Hop algorithm has significantly lower localization errors compared to SSADV-Hop, PSODV-Hop, GADV-Hop, and DV-Hop. Specifically, in Fig. 6 (f), the box width of IMOADV-Hop is narrower than the four comparison algorithms, indicating that the localization results obtained from IMOADV-Hop are more stable.

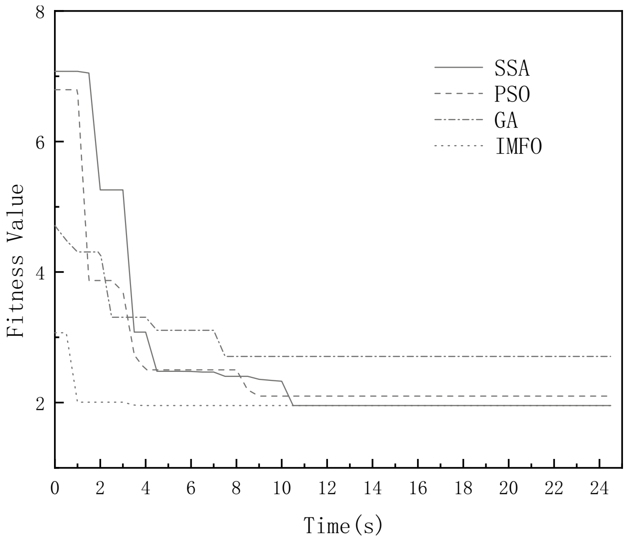


Fig. 4 Iteration curve of four algorithms

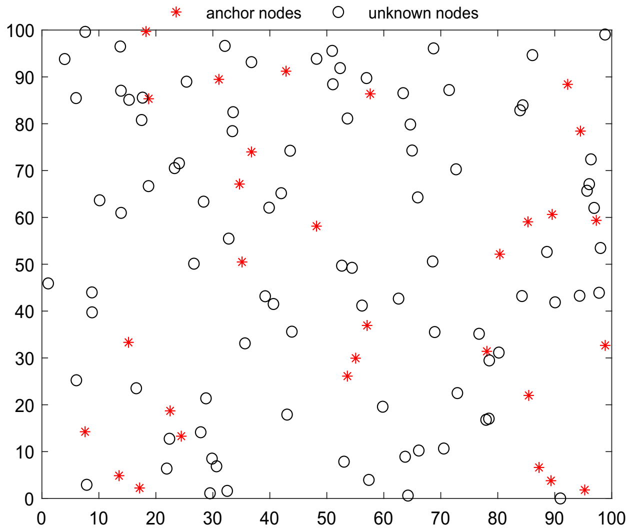
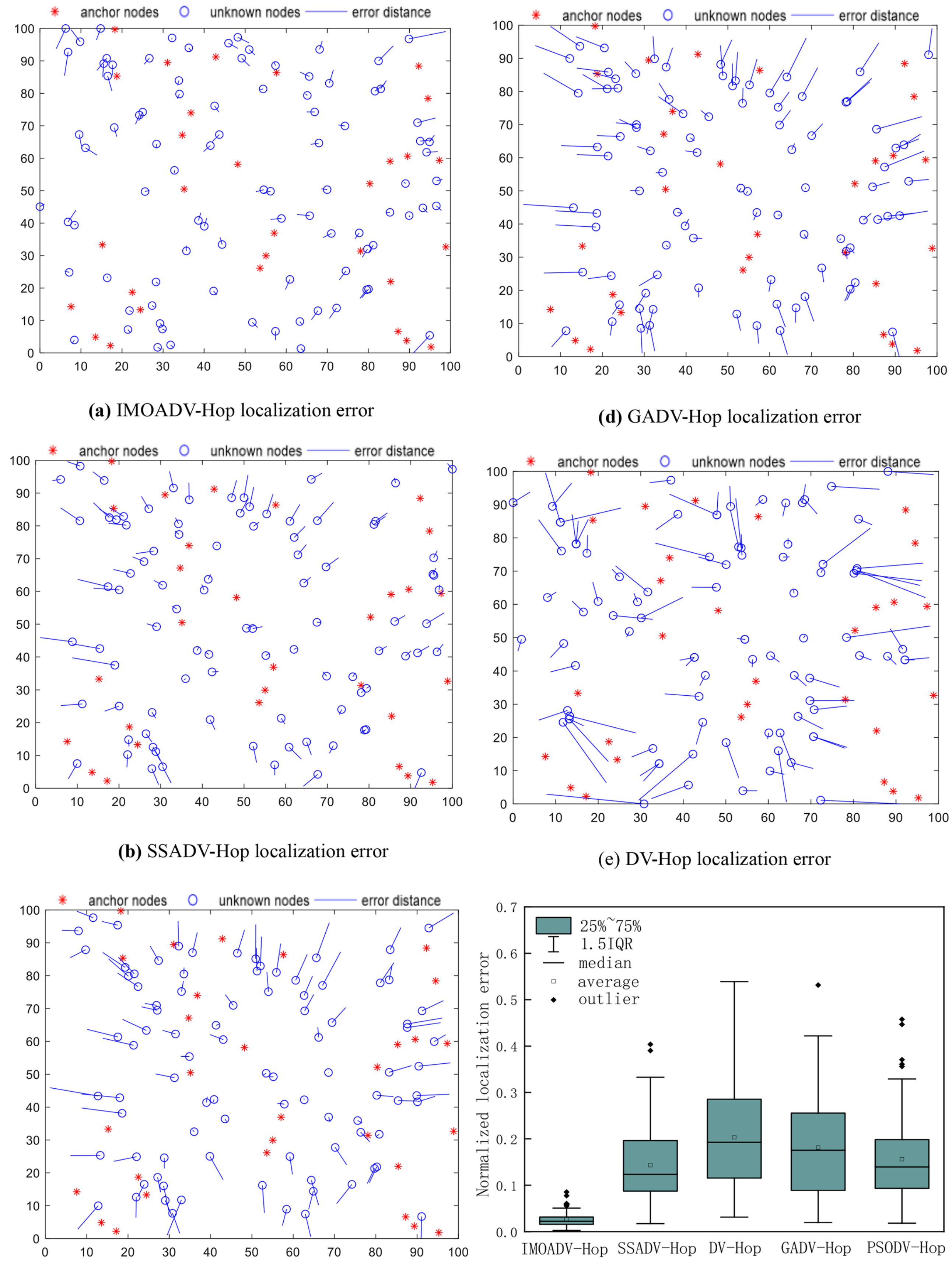


Fig. 5 Distribution of sensor node



(a) IMOADV-Hop localization error

(d) GADV-Hop localization error

(b) SSADV-Hop localization error

(e) DV-Hop localization error

(c) PSODV-Hop localization error

(f) Localization error boxplot

◀**Fig. 6** Localization error comparison

4.3 The effect of local hop and minimum communication radius

In this simulation, a total of 120 nodes are randomly distributed in the simulation area, which include 30 anchor nodes. The communication radius is set to 30m. As shown in Fig. 7, it can be noticed that the error rate of the five algorithms stops decreasing when the number of local hops exceeds 2. This implies that increasing the number of local hops may not always lead to a reduction in error. Similarly, in Fig. 8, it is evident that when the minimum communication radius is smaller than R/3, the localization error reaches a plateau and hardly changes further. Based on these observations, we conclude that the optimal number of local hops for this study is 2, and the minimum communication radius is set to R/3.

4.4 The effect of the anchor nodes number

In this section of the simulation, a total of 120 nodes are randomly deployed within the simulation area, with a communication radius of 30m. The number of anchor nodes is gradually increased from 20 to 50. The localization errors of the five algorithms are presented in Fig. 9.

As shown in Fig. 9, the localization errors of all five algorithms decrease as the number of anchor nodes increase. This improvement is due to the increased availability of favorable location information provided by a greater number of anchor nodes. Furthermore, the IMOADV-Hop algorithm consistently outperforms the other four algorithms across different numbers of anchor nodes. The normalized average localization error of the IMOADV-Hop algorithm is 0.087, which represents a reduction of 12.1%, 35.5%, 34.1%, and 47.3% compared to SSADV-Hop, PSODV-Hop, GADV-Hop, and DV-Hop, respectively.

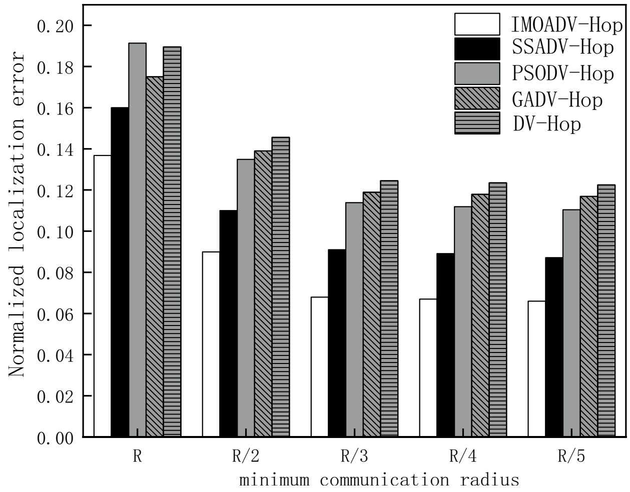


Fig. 8 The effect of minimum communication radius

4.5 The effect of communication radius

In this simulation, a total of 120 nodes are randomly deployed within the simulation area, out of which 30 anchor nodes were included. The communication radius is gradually increased from 20 to 50m. Figure 10 illustrates the comparison of localization errors of five algorithms under different communication radii. As shown in Fig. 10, all five algorithms showed a significant decrease in localization error as the communication radius increased. This improvement can

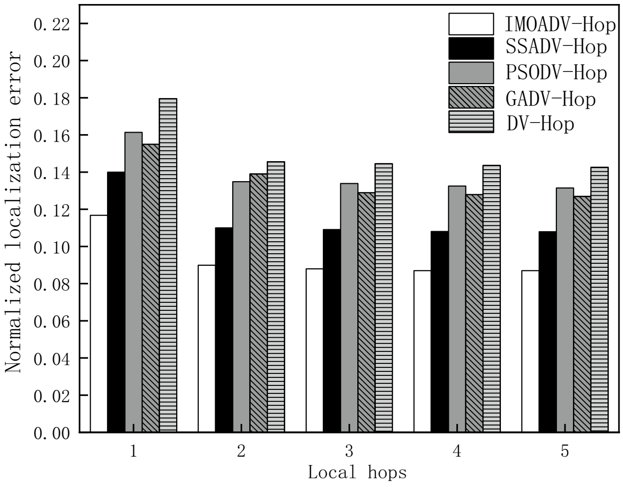


Fig. 7 The effect of the local hop

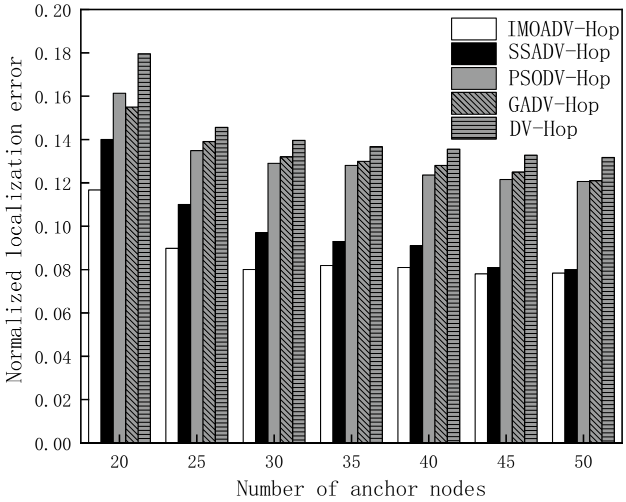


Fig. 9 The number of anchor nodes

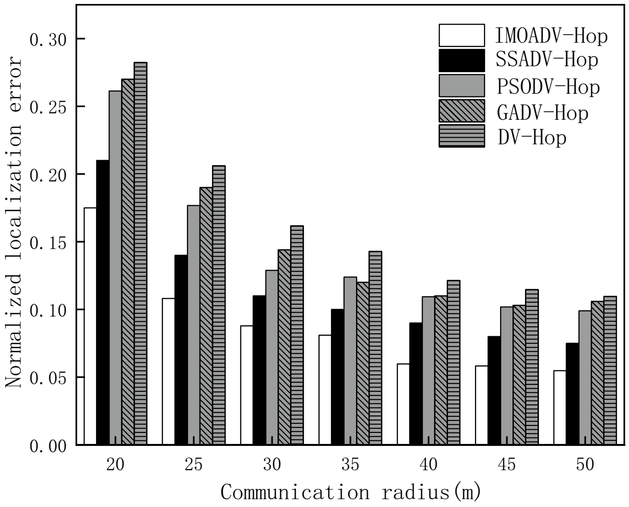


Fig. 10 The effect of communication radius

be attributed to the increased network connectivity resulting from the larger communication radius, which in turn provides more anchor node information for localization. Consequently, the localization error decreases. For different communication radii, the IMOADV-Hop algorithm displayed the lowest normalized average localization error of 0.088. This represents a reduction of 23.4%, 40.5%, 40.9%, and 45.3% compared to SSADV-Hop, PSODV-Hop, GADV-Hop, and DV-Hop, respectively.

4.6 The effect of total nodes number

In this part of the simulation, the node communication radius is set to 30m and the proportion of anchor nodes is 30%. The total number of nodes increases gradually from 100 to 200. The localization errors of the five algorithms are presented in Fig. 11.As shown in the Figure, the localization errors of all five algorithms decrease as the total number of nodes increases. Moreover, the IMOADV-Hop algorithm consistently achieves a lower normalized average localization error than the other four algorithms. The normalized average localization errors for the five algorithms are 0.078, 0.113, 0.128, 0.139, and 0.172, respectively. Compared to the four comparison algorithms, the IMOADV-Hop algorithm reduces the average localization error by 39.1%, 30.9%, 43.9%, and 54.7%, respectively.

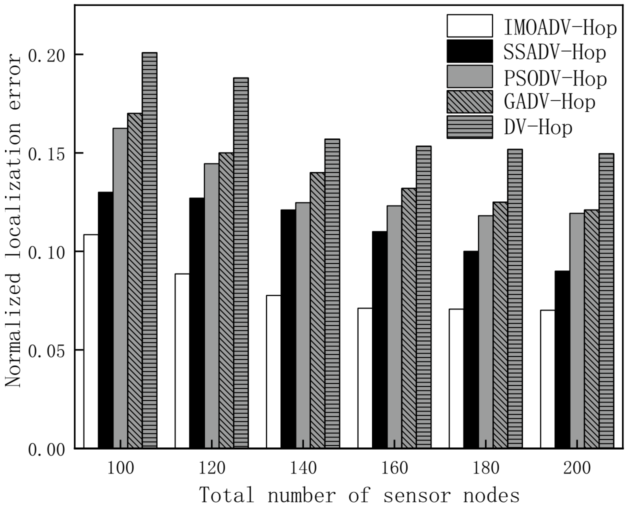


Fig. 11 The effect of total nodes number

5 Conclusion

In this study, a novel DV-Hop localization algorithm called IMOADV-Hop is proposed to improve the localization accuracy of DV-Hop. The algorithm introduces several improvements to overcome the limitations of the original DV-Hop. Firstly, the IMOADV-Hop algorithm utilizes three different communication radii for anchor nodes to broadcast information. This approach aims to obtain a more accurate minimum hop count, which is critical for distance estimation in DV-Hop. By using multiple communication radii, the algorithm can enhance the precision of distance calculations. Secondly, the algorithm optimizes the weights of anchor nodes within a two-hop range of the unknown node. This optimization process aims to improve the average hop distance calculation, which is another crucial aspect of DV-Hop. By refining the weights, the algorithm aims to achieve more accurate localization results. Lastly, the IMOADV-Hop algorithm incorporates improvements to the MOA used for estimating the positions of unknown nodes. These enhancements include utilizing a good point set, Levy flight, and t-distribution. These enhancements aim to improve the accuracy and performance of the position estimation process. Simulation results demonstrate that the IMOADV-Hop algorithm outperforms the SSADV-Hop, PSODV-Hop, GADV-Hop, and DV-Hop algorithms in terms of localization accuracy. The proposed algorithm achieves higher precision and accuracy in estimating the positions of unknown nodes compared to the other algorithms. Although the proposed algorithm has high accuracy in simulation experiments, it has not been validated in actual environments. It is still debatable whether the parameters used in simulation have the same localization accuracy when applied to real environments. Therefore, how to choose appropriate parameters in practical environments is a interesting point for upcoming studies.

Author contributions Xiuwu Yu and Wei Peng wrote the main part of the paper. Zixiang Zhou, Ke Zhang and Yong Liu checked the paper.

Funding This work was in part supported by Hunan Provincial Natural Science Foundation of China (2024JJ5338); National Natural Science Foundation of China (No.11875164); University of South China Postdoctoral Research star-up Fund(230XQD053).

Data availability The data that supporting the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Ethics approval The research adhered to all applicable laws and regulations governing online research, data privacy, and human subject protection.

Consent to publish We agree to publish.

Competing interest The authors declare no competing interests.

References

1. Strumberger I, Minovic M, Tuba M, Bacanin N (2019) Performance of elephant herding optimization and tree growth algorithm adapted for node localization in wireless sensor networks. Sensors 19(11):2515–2544
2. Kim TH, Goyat R, Rai MK, Kumar G, Buchanan WJ, Saha R, Thomas R (2019) A novel trust evaluation process for secure localization using a decentralized blockchain in wireless sensor networks. IEEE Access 7:184133–184144
3. Du R, Gkatzikis L, Fischione C, Xiao M (2015) Energy efficient sensor activation for water distribution networks based on compressive sensing. IEEE J Sel Areas Commun 33(12):2997–3010
4. Boubrima A, Bechkit W, Rivano H (2017) Optimal WSN deployment models for air pollution monitoring. IEEE Trans Wireless Commun 16(5):2723–2735
5. Boukerche A, Oliveira HA, Nakamura EF, Loureiro AA (2007) Localization systems for wireless sensor networks. IEEE Wirel Commun 14(6):6–12
6. Tomic S, Beko M, Dinis R, Montezuma P (2017) Distributed algorithm for target localization in wireless sensor networks using RSS and AoA measurements. Pervasive Mob Comput 37:63–77
7. Nemer I, Sheltami T, Shakshuki E, Elkhail AA, Adam M (2021) Performance evaluation of range-free localization algorithms for wireless sensor networks. Pers Ubiquit Comput 25:177–203
8. Hoang MT, Yuen B, Dong X, Lu T, Westendorp R, Reddy K (2019) Recurrent neural networks for accurate RSSI indoor localization. IEEE Internet Things J 6(6):10639–10651
9. Zhao S, Zhang XP, Cui X, Lu M (2021) A new TOA localization and synchronization system with virtually synchronized periodic asymmetric ranging network. IEEE Internet Things J 8(11):9030–9044
10. Wu P, Su S, Zuo Z, Guo X, Sun B, Wen X (2019) Time difference of arrival (TDoA) localization combining weighted least squares and firefly algorithm. Sensors 19(11):2554–2567
11. Hao K, Xue Q, Li C, Yu K (2020) A hybrid localization algorithm based on Doppler shift and AOA for an underwater mobile node. IEEE Access 8:181662–181673
12. Cai X, Wang P, Du L, Cui Z, Zhang W, Chen J (2019) Multi-objective three-dimensional DV-hop localization algorithm with NSGA-II. IEEE Sens J 19(21):10003–10015
13. Liu J, Wang Z, Yao M, Qiu Z (2016) VN-APIT: Virtual nodes-based range-free APIT localization scheme for WSN. Wireless Netw 22:867–878
14. Blumenthal J, Grossmann R, Golatowski F, Timmermann D (2007) Weighted centroid localization in zigbee-based sensor networks. In 2007 IEEE International Symposium on Intelligent Signal Processing, pp 1–6
15. Zhao LZ, Wen XB, Li D (2015) Amorphous localization algorithm based on BP artificial neural network. Int J Distrib Sens Netw 11(7):657241–657246
16. Cao Y, Wang Z (2019) Improved DV-hop localization algorithm based on dynamic anchor node set for wireless sensor networks. IEEE Access 7:124876–124890
17. Chen H, Sezaki K, Deng P, So HC (2008) An improved DV-Hop localization algorithm with reduced node location error for wireless sensor networks. IEICE Trans Fundam Electron Commun Comput Sci 91(8):2232–2236
18. Kumar S, Lobiyal DK (2017) Novel DV-Hop localization algorithm for wireless sensor networks. Telecommun Syst 64:509–524
19. Kumar S, Lobiyal DK (2013) An advanced DV-Hop localization algorithm for wireless sensor networks. Wireless Pers Commun 71:1365–1385
20. Hu Y, Li X (2013) An improvement of DV-Hop localization algorithm for wireless sensor networks. Telecommun Syst 53:13–18
21. Messous S, Liouane H, Cheikhrouhou O, Hamam H (2021) Improved recursive DV-hop localization algorithm with RSSI measurement for wireless sensor networks. Sensors 21(12):4152–4166
22. Xiao H, Zhang H, Wang Z, Gulliver TA (2017) An RSSI based DV-hop algorithm for wireless sensor networks. In 2017 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), pp 1–6
23. Kaur A, Kumar P, Gupta GP (2019) A weighted centroid localization algorithm for randomly deployed wireless sensor networks. J King Saud Univ-Comput Inform Sci 31(1):82–91
24. Messous S, Liouane H, Liouane N (2020) Improvement of DV-Hop localization algorithm for randomly deployed wireless sensor networks. Telecommun Syst 73:75–86
25. Ouyang A, Lu Y, Liu Y, Wu M, Peng X (2021) An improved adaptive genetic algorithm based on DV-Hop for locating nodes in wireless sensor networks. Neurocomputing 458:500–510
26. Chen X, Zhang B (2012) Improved DV-Hop node localization algorithm in wireless sensor networks. Int J Distrib Sens Netw 8(8):213980–213986
27. Lei Y, De G, Fei L (2020) Improved sparrow search algorithm based DV-Hop localization in WSN. In 2020 Chinese Automation Congress (CAC), pp 2240–2244
28. Cui Z, Sun B, Wang G, Xue Y, Chen J (2017) A novel oriented cuckoo search algorithm to improve DV-Hop performance for cyber–physical systems. J Parallel Distribut Comput 103:42–52
29. Li J, Gao M, Pan JS, Chu SC (2021) A parallel compact cat swarm optimization and its application in DV-Hop node localization for wireless sensor network. Wireless Netw 27:2081–2101
30. Chen J, Zhang W, Liu Z, Wang R, Zhang S (2020) CWDV-Hop: A hybrid localization algorithm with distance-weight DV-Hop and CSO for wireless sensor networks. IEEE Access 9:380–399
31. Shi Q, Wu C, Xu Q, Zhang J (2021) Optimization for DV-Hop type of localization scheme in wireless sensor networks. J Supercomput 77(12):13629–13652
32. Zervoudakis K, Tsafarakis S (2020) A mayfly optimization algorithm. Comput Ind Eng 145:106559–106581
33. Yu XW, Huang LP, Liu Y, Zhang K, Li P, Li Y (2022) WSN node location based on beetle antennae search to improve the gray wolf algorithm. Wireless Netw 28(2):539–549
34. Zhang J, Wang JS (2020) Improved salp swarm algorithm based on levy flight and sine cosine operator. IEEE Access 8:99740–99771
35. Yang X, Liu J, Liu Y, Xu P, Yu L, Zhu L, Deng W (2021) A novel adaptive sparrow search algorithm based on chaotic mapping and t-distribution mutation. Appl Sci 11(23):11192–11213

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

Xiuwu Yu received the M.S. and Ph.D. degree from Wuhan University of Technology, China, in 2009 and 2012. He is currently a professor with the University of South China (USC), Hengyang, China. His research interests include localization, routing, data fusion and coverage control in wireless sensor networks. He has published more than 20 technical papers in the international journals and conferences.

Wei Peng obtained his B.S. from the University of South China, Hengyang, China in 2021. He is now working toward his Master Degree in the University of South China (USC), Hengyang. His research interest is wireless sensor network routing protocols and localization algorithms.

Zixiang Zhou is now working toward his Master Degree in the School of Resource & Environment and Safety Engineering, University of South China (USC), Hengyang, China. His research interest is wireless sensor network.

Ke Zhang is with the School of Resource & Environment and Safety Engineering, University of South China, Hengyang 421001, China. He is also a member of Key Lab Emergency Safety Technol & Equipment Nuclear, Hengyang 421001, China.

Yong Liu received the M.S. and Ph.D. degree from the University of South China, Hengyang, China in 2004 and 2015. He is currently a professor with the Shenzhen University, Shenzhen, China. His research area is mainly safety engineering. He has published more than 20 technical papers in the international journals and conferences.