

Improved DV-Hop Localization Algorithm Based on Bat Algorithm in Wireless Sensor Networks

Yuan Liu, Junjie Chen and Zhenfeng Xu

Southeast University

Nanjing 210096, Jiangsu - China

[e-mail: liuyseu@gmail.com, inschenjj@seu.edu.cn, 284913151@qq.com]

*Corresponding author: Junjie Chen

*Received August 10, 2016; revised October 21, 2016; accepted November 30, 2016;
published January 31, 2017*

Abstract

Obtaining accurate location information is important in practical applications of wireless sensor networks (WSNs). The distance vector hop (DV-Hop) is a frequently-used range-free localization algorithm in WSNs, but it has low localization accuracy. Moreover, despite various improvements to DV-Hop-based localization algorithms, maintaining a balance between high localization accuracy and good stability and convergence is still a challenge. To overcome these shortcomings, we proposed an improved DV-Hop localization algorithm based on the bat algorithm (IBDV-Hop) for WSNs. The IBDV-Hop algorithm incorporates optimization methods that enhance the accuracy of the average hop distance and fitness function. We also introduce a nonlinear dynamic inertial weight strategy to extend the global search scope and increase the local search accuracy. Moreover, we develop an updated solutions strategy that avoids premature convergence by the IBDV-Hop algorithm. Both theoretical analysis and simulation results show that the IBDV-Hop algorithm achieves higher localization accuracy than the original DV-Hop algorithm and other improved algorithms. The IBDV-Hop algorithm also exhibits good stability, search capability and convergence, and it requires little additional time complexity and energy consumption.

Keywords: Wireless sensor networks, distance vector hop, bat algorithm, optimization methods, localization performance

This work was supported by the National Science and Technology Support Project of China under Grant No.2014BAD08B03, the Science and Technology Special Fund Project of Northern Jiangsu Province under Grant No.BN2014085, the Agricultural Science and Technology Support Project of Jiangsu Province under Grant No.BE2014312, the Three New Aquaculture Project of Jiangsu Province under Grant No.Y2016-3.

1. Introduction

Wireless sensor networks (WSNs) enable the sensing of environmental information and have wide applications in various fields such as the military, medical care, environmental monitoring, disaster monitoring and home automation [1, 2]. The vast majority of these practical applications demand increasingly precise location information for the sensor node where an event occurred [3]. However, design an algorithm that can locate nodes with low energy consumption, high accuracy and good stability and practicability is difficult.

The existing localization algorithms for WSNs can be categorized into range-based localization algorithms and range-free localization algorithms. In range-based localization algorithms, the distance or angle information of a node from a known location is directly acquired and used for location estimation. The angle of arrival (AOA), the radio signal strength indicator (RSSI), the time of arrival (TOA) and the time difference of arrival (TDOA) [4] are common range-based localization algorithms. In contrast, in range-free localization algorithms, knowing absolute information such as distance, angle and other physical measurements is unnecessary because of the low hardware requirements. Typical range-free localization algorithms include the centroid [5], the convex optimization [6], the lateration scheme [7], the multidimensional scaling map (MDS-MAP) [8], the cluster-based MDS (CMDS) [9], the approximate point-in-triangulation test (APIT) [10], the distance vector hop (DV-Hop) [11] and the expected hop progress (EHP) [12]. DV-Hop, which has been widely applied in various applications, is a useful localization algorithm that has low-cost, low energy consumption and is simple to implement. However, its localization accuracy is always unsatisfactory, and its localization performance is susceptible to the network distribution. When the DV-Hop algorithm is employed in networks with non-uniformly distributed sensor nodes, the algorithm's inaccurate measurement of the average hop distance produces large distance error, which accumulates and seriously affects the DV-Hop algorithm's multi-objective optimization process. In such circumstances, the DV-Hop algorithm can yield results that include a large localization error.

A series of optimal theories and methods have been applied to improve the DV-Hop algorithm, e.g., the expected hop [12], the weight algorithm [13, 14], the collinearity normalized theory [15], the RSS measurements [16, 17], the network geometry [18, 19] and the new multi-objective optimization algorithms [20]. These algorithms mainly focus on increasing the accuracy of the estimated distance in the DV-Hop algorithm and ignore the potential localization error produced during the process of the multi-objective optimization. In recent years, some intelligent optimization algorithms such as particle swarm optimization (PSO) [21-23] and the genetic algorithm (GA) [24, 25] have been used to reduce the localization error in multi-objective optimization methods based on the DV-Hop algorithm. However, these intelligent optimization algorithms fall short of a balanced performance that includes high localization accuracy, good convergence and stability and low complexity.

The bat algorithm (BA) has shown better global search capability compared to other intelligent optimization algorithms [26-28]. The BA is a novel swarm intelligent optimization algorithm with a simple model. Therefore, the DV-Hop based on the bat algorithm (BDV-Hop) promises to provide a better-balanced localization algorithm for WSNs. Although the BA has an outstanding capability for solving multi-objective optimization problems, it has poor stability and is prone to premature convergence. Motivated to enhance the localization accuracy and improve the stability and convergence of BDV-Hop algorithm, we propose an

improved BDV-Hop (IBDV-Hop) algorithm for WSNs where the sensor nodes are randomly deployed at an arbitrary density.

In this paper, we first describe the basic theory and localization mode of the BDV-Hop algorithm and analyze its existing deficiencies. Then, to reduce the deficiencies, we propose the IBDV-Hop algorithm, which contains improvements in the calculation of the average hop distance, in its fitness function and in velocity correction and also features an updated solutions strategy. In addition, we carry out simulation experiments to evaluate the localization accuracy, stability, search capability and convergence of the IBDV-Hop algorithm compared with the original DV-Hop, the BDV-Hop and other improved DV-Hop-based algorithms. Moreover, we explore the impacts of varying the maximum iterations, the size of the bat group, the density of anchor nodes and the node communication range on the localization performance of the IBDV-Hop algorithm and the other tested algorithms. The main contributions of this paper are as follows.

- (1) Characterize and demonstrate the localization error caused by the average hop distance and fitness function in DV-Hop algorithm. Analyze the poor stability and convergence of the BDV-Hop algorithm.
- (2) Propose the IBDV-Hop algorithm, which has a higher localization accuracy, a more powerful search capability and better stability and convergence compared to the other algorithms mentioned in this paper. We also note that the IBDV-Hop algorithm has little additional time complexity and energy consumption compared with the DV-Hop algorithm.
- (3) Investigate the influences of the maximum number of iterations, the size of the bat group, the density of anchor nodes and the node communication range in the localization performance of the IBDV-Hop and other algorithms.

The rest of this paper is organized as follows. Section 2 presents the related works on the improvements to the DV-hop algorithm. Section 3 describes the BDV-Hop algorithm and its deficiencies. Section 4 introduces the IBDV-Hop algorithm. A theoretical analysis of the IBDV-Hop algorithm and simulation results and comparisons are illustrated in Section 5. Finally, Section 6 presents the conclusions.

2. Related Work

Extensive studies have been conducted to improve the DV-Hop algorithm. These improvements have focused on two aspects: increasing the accuracy of the calculation of the average hop distance for sensor nodes and designing better multi-objective optimization algorithms.

The localization error caused by the average hop distance calculation to the anchor nodes in the DV-Hop algorithm have a significant impact on the estimated locations of the unknown nodes. [12] proposes a range-free localization algorithm by using the expected hop progress (LAEP) in WSNs with sensor nodes randomly deployed. In LAEP algorithm, the expected hop progress from a network model is derived according to the network parameters. [13] presents an improved DV-Hop algorithm by calculating the average hop distance of all anchor nodes instead of the average hop distance of nearest anchor nodes. A threshold M is introduced and the average hop distance of unknown node is estimated through the weighted average hop distances of anchor nodes within M hop. A normalized collinearity DV-Hop algorithm is presented in [15]. The algorithm is adopted to the Voronoi diagram which divides the WSNs into several regions, and the anchor node information is obtained by each Voronoi polygon. The unknown node's position can be estimated with relatively higher accuracy through the

anchor node information and the collinearity condition. [16] suggests a DV-Hop algorithm with RSS measurements and thresholds to improve the connectivity of the network. The pairs of nodes whose RSS values exceed the threshold are considered as the neighbors nodes. A Sub-Square Weighted (SSW) DV-hop algorithm is used in [17]. The SSW DV-hop algorithm is based on the rectangle topology which contributes to dividing the network into small square fields by taking the short edge of the rectangle field as the length of the edge for square. The position of an unknown node is the product of the estimated position in sub-square fields. Two improved DV-Hop algorithms (Checkout DV-hop and Selective 3-Anchor DV-hop) are given in [19]. The Checkout DV-hop algorithm estimates the position of mobile node by nearest anchor nodes and adjusts the estimated position of a normal node. The Selective 3-Anchor DV-hop algorithm creates three best anchor nodes and the best position of candidate positions are used to calculate the location of unknown node. [20] puts forward three improved DV-Hop (iDV-Hop1, iDV-Hop2, and Quad DV-Hop) algorithms under the irregular topologies situation. In iDV-Hop1 algorithm, three reference points are formed by choosing the nearest anchor nodes of the unknown node. The iDV-Hop2 algorithm is modified by using anchor location as the third reference point. For Quad DV-Hop algorithm, the least squares problem is transformed to the quadratic programming problem.

In the original DV-Hop algorithm, the trilateration algorithm or the least squares method is applied to solve the multi-objective optimization problem and transform the location information for unknown nodes to coordinates. However, the fitness function of multi-objective optimization is sensitive to the distance error. Any error in the estimated average hop distance accumulates in the process of multi-objective optimization, which may result in quite large localization error. Hence, many studies have concentrated on improving the multi-objective optimization algorithms. A weighted least square algorithm is presented in [14] to reduce the inherent error of estimated distance between anchor nodes and the unknown node. [21] designs an improved DV-Hop algorithm by combining the DV-Hop with PSO algorithm. The PSO algorithm is used to minimize the localization error and revise the localization coverage rate and average localization accuracy. Another improved DV-Hop algorithm with PSO algorithm is proposed in [22] which optimizes the particle velocity, inertial weight, learning strategy and variation. The improved DV-Hop algorithm is able to avoid the local optimal solution and enhance the search speed in later iterative stage to increase localization accuracy. [23] also presents an improved DV-Hop algorithm with PSO algorithm to optimize the iterative process. An improved DV-Hop algorithm based on GA is proposed in [24]. The algorithm establishes a mathematical optimization model of the distance between the unknown node and anchor nodes. The GA is used to search for the optimal solution of DV-Hop algorithm. In [25], a hybrid algorithm of GA and the simplex method is given to optimize the location of the unknown node. The fitness function combines the cost function with a penalty function and the simplex method to improve the ability of algorithm on local search.

3. The BDV-Hop Algorithm

The BDV-Hop algorithm can be described as a hybrid localization algorithm that combines the DV-Hop algorithm and BA. First, the DV-Hop algorithm is used to acquire the location information of nodes using a range-free approach. Then, the location information is optimized and transformed to the coordinates of the unknown nodes by the BA. In this paper, we assume that the sensor nodes work as follows.

- (1) All nodes are randomly deployed in the network and use a plane routing protocol.
- (2) In the process of hop counting, nodes communicate with their neighbor nodes in a flooding communication mode.
- (3) Only the anchor nodes transmit data gathered from the sensor nodes to the base station; they also issue orders from the base station to the sensor nodes.
- (4) The improvements of IBDV-Hop algorithm are implemented in the base station.

3.1 Location Information Acquisition by the DV-Hop Algorithm

In the original DV-Hop algorithm, the distance between the anchor nodes and the unknown node is estimated by the product of the average hop distance and hop count value. The process of location information acquisition includes three phases [11, 18].

(1) Hop counting. Anchor nodes broadcast data packets that contain their location information and the number of hops to their neighbor nodes. When a node receives a packet, it adds 1 to the hop count and then records the hop count and the location of the sending anchor node. Subsequently, the nodes transmit these packets to their neighbor nodes until all nodes have received packets.

(2) Distance estimation. In the first phase, the location information and the number of hops to anchor nodes are recorded. Therefore, the average hop distance to an anchor node is given by

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

where (x_i, y_i) is the location of anchor node i and h_{ij} is the number of hops between anchor node i and anchor node j . The estimated distance is calculated by

$$d_{ik} = HopSize_i \times h_{ik} \quad (2)$$

where d_{ik} is the estimated distance between anchor node i and node k .

(3) Coordinate calculating. In (1) and (2), the location of the unknown node is expressed as

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = d_1 \\ (x_2 - x)^2 + (y_2 - y)^2 = d_2 \\ \vdots \\ (x_n - x)^2 + (y_n - y)^2 = d_n \end{cases} \quad (3)$$

where (x, y) is the coordinate of an unknown node. Next, (3) is transformed into matrix form

$$AX = b \quad (4)$$

$$\text{where } A = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ 2(x_2 - x_n) & 2(y_2 - y_n) \\ \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix}, X = \begin{bmatrix} x \\ y \end{bmatrix} \text{ and } b = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 - d_1^2 + d_n^2 \\ x_2^2 - x_n^2 + y_2^2 - y_n^2 - d_2^2 + d_n^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-2}^2 - y_n^2 - d_{n-1}^2 + d_n^2 \end{bmatrix}.$$

Then, X is represented by

$$X = (A^T A)^{-1} A^T b \quad (5)$$

3.2 Location Information Optimization and Transformation by the BA

The optimal solution for (5) is obtained by solving a multi-objective optimization problem. The distance error is defined as

$$|d_i^* - d_i| \leq \varepsilon_i \quad (6)$$

where d_i denotes the estimated distance between the anchor node i and the unknown node and d_i^* denotes the actual Euclidean distance between the anchor node i and the unknown node.

The overall distance error ε between anchor nodes (x_i, y_i) and the unknown node (x, y) can be expressed as

$$\begin{cases} d_1^2 - \varepsilon_1^2 \leq (x - x_1)^2 + (y - y_1)^2 \leq d_1^2 + \varepsilon_1^2 \\ d_2^2 - \varepsilon_2^2 \leq (x - x_2)^2 + (y - y_2)^2 \leq d_2^2 + \varepsilon_2^2 \\ \vdots \\ d_n^2 - \varepsilon_n^2 \leq (x - x_n)^2 + (y - y_n)^2 \leq d_n^2 + \varepsilon_n^2 \end{cases} \quad (7)$$

Obviously, the accuracy of the estimated locations in (6) and (7) is negatively correlated with the distance error. To evaluate the localization accuracy, a fitness function is modeled as

$$F(x, y) = \sum_{i=1}^n (\sqrt{(x - x_i)^2 + (y - y_i)^2} - d_i)^2 \quad (8)$$

In the BA, the location of a bat represents one potential solutions for (8). Therefore, solving the problem of multi-objective optimization is regarded as searching for the optimal individual locations of the bat group. The process of optimizing and transforming the location information of an unknown node by the BA is divided into four steps [26, 28].

(1) Initialize the bat group. The bat group is generated randomly, and each bat is initialized with a location and velocity. Then, the bat group is evaluated by the fitness function in (8). The bat with the smallest fitness value is chosen to be the initial optimal individual, and its location is the initial optimal solution x^* .

(2) Update the status of bat group. The status of each bat is updated by adjusting the pulse frequency as

$$\begin{cases} f_i = f_{\min} + (f_{\max} - f_{\min})\beta \\ v_i^t = v_i^{t-1} + (x_i^t - x^*)f_i \\ x_i^t = x_i^{t-1} + v_i^t \end{cases} \quad (9)$$

where f_i is the pulse frequency of bat i , f_{\max} is the maximum pulse frequency, f_{\min} is the minimum pulse frequency, β ($\beta \in [0, 1]$) is a random factor with uniform distribution, v_i^t is the velocity of bat i at time t , and x_i^t is the location of bat i .

(3) Update the solutions. A new solution with a random perturbation is generated with a given probability by

$$x_i^t = x_i + \alpha A^t \quad (10)$$

where α ($\alpha \in [-1, 1]$) is a random variable, and A^t is the volume parameter of bat i at time t .

Then, compare the fitness values of the new solution x_i^t with the previous optimal solution. If x_i^t is smaller, x_i^t replaces the previous optimal solution. Finally, update the solutions of the bat group and continue to search for better optimal solutions.

(4) Adaptively adjust the parameters. According to the biological principles of the bat searching behavior, its pulse frequency is low but its volume is high during the initial search process, which is conducive to enlarging the search scope. Later, the bat increases the pulse frequency and gradually decreases the volume to improve the search accuracy. In the BA, the pulse frequency and volume are updated by

$$\begin{cases} A_i^{t+1} = \delta A_i^t \\ r_i^{t+1} = r_i^0 \times [1 - \exp(-\gamma t)] \end{cases} \quad (11)$$

where $\delta(0 < \delta < 1)$ and $\gamma(\gamma > 0)$ are constants, r_i^{t+1} is the pulse emission rate at time $t+1$, and r_i^0 is the maximum pulse emission rate.

Repeat the last three steps until the iterative termination condition has been reached. The last optimal solution is the location of the unknown node.

3.3 Deficiencies of the BDV-Hop Algorithm

Although the BDV-Hop algorithm provides a novel method to solve the multi-objective optimization problem, there are some deficiencies in both the DV-Hop algorithm and in the BA.

3.3.1 Error of Estimated Distance

For the BDV-Hop algorithm and most of the improved DV-Hop algorithms, the distance between nodes is estimated by the average hop distance and the number of hops in (2). Therefore, the localization accuracy of the BDV-Hop algorithm is closely related to the accuracy of the average hop distance.

Definition 1: n nodes are randomly distributed in networks. The communication range of all nodes is r . Therefore, the actual distance between node i and node j is $l_{ij} (l_{ij} \leq r)$. According to the rules of the BDV-Hop algorithm, the estimated distance between nodes is defined as

$$l_{ij}^* = h_{ij}r \tag{12}$$

where h_{ij} is the number of hops between node i and node j . Therefore, the error of estimated distance between node i and node j can be defined as

$$\begin{aligned} e_{ij} &= |l_{ij}^* - l_{ij}| \\ &= h_{ij}r - l_{ij} \end{aligned} \tag{13}$$

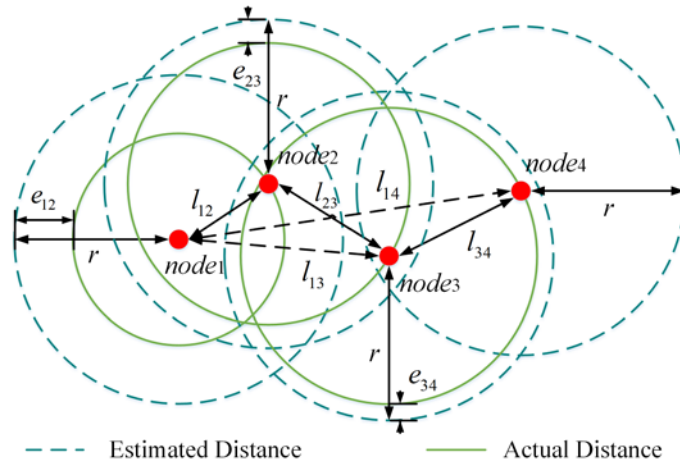


Fig. 1. Network with usual distribution

Fig. 1 shows the network with usual distribution. The error of the estimated distance between $node_1$ and $node_2$ is $e_{12} = |l_{12}^* - l_{12}| = r - l_{12}^*$, the error of the estimated distance between $node_1$ and $node_3$ is $e_{13} = |l_{12}^* - l_{12}| + |l_{23}^* - l_{23}| = 2r - l_{12} - l_{23}$, and the error of the estimated distance between $node_1$ and $node_4$ is $e_{14} = |l_{12}^* - l_{12}| + |l_{23}^* - l_{23}| + |l_{34}^* - l_{34}| = 3r - l_{12} - l_{23} - l_{34}$. According to **Fig. 1**, the error of the estimated distance between $node_m$ and $node_n$ is defined as

$$\begin{aligned}
e_{mn} &= \left| l_{m(m+1)}^* - l_{m(m+1)} \right| + \left| l_{(m+1)(m+2)}^* - l_{(m+1)(m+2)} \right| + \dots + \left| l_{(n-1)n}^* - l_{(n-1)n} \right| \\
&= (n-m)r - \sum_{i=m}^{n-1} l_{i(i+1)}
\end{aligned} \tag{14}$$

where n and m ($1 < m < n$) are positive integers.

Lemma 1: The greater the number of hops among nodes is, the larger the error of estimated distance will be.

Proof: In (14), we have

$$\begin{aligned}
e_{mn} - e_{m(n-1)} &= (n-m)r - \sum_{i=m}^{n-1} l_{i(i+1)} - ((n-1-m)r - \sum_{i=m}^{n-2} l_{i(i+1)}) \\
&= r - l_{(n-1)n}
\end{aligned} \tag{15}$$

From **Definition 1**, the actual distance between $node_n$ and $node_{n-1}$ is $l_{(n-1)n} \leq r$. Therefore, (15) is expressed as

$$e_{mn} - e_{m(n-1)} = r - l_{(n-1)n} \geq 0 \tag{16}$$

We can observe that the distance error accumulates as the number of hop among nodes increases. That is, the estimated distance between nodes with small hop counts is more accurate than among those nodes with large hop counts. Therefore, decrease the influence of anchor nodes that have small hop counts to the unknown node will contribute to improving the accuracy of estimated distance.

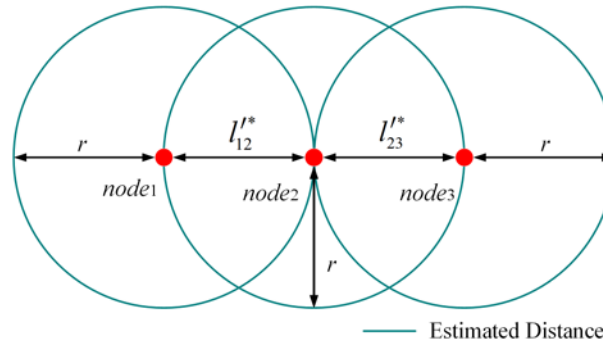


Fig. 2. Network with ideal distribution

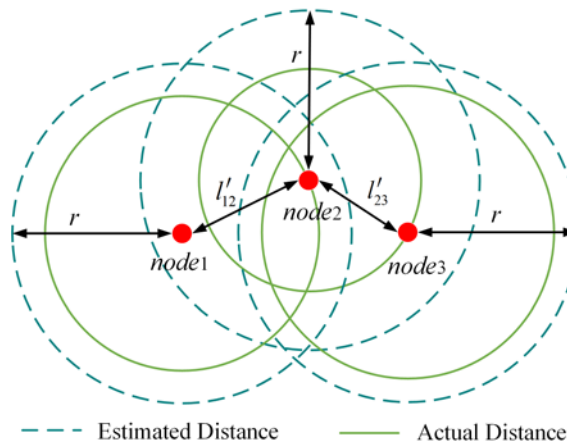


Fig. 3. Network with random distribution

Fig. 2 and **Fig. 3** show the ideal distribution and random distribution of network. In **Fig. 2**, the distance between two neighbor nodes in the ideal distribution equals the communication range r . In **Fig. 3**, the distance between the two neighbor nodes in the random distribution is expressed as $l_{12} < r$ and $l_{23} < r$. In most cases, the random network distributions as shown in **Fig. 3** are more in accord with practical circumstances.

Definition 2: In the network with ideal distribution, the distance between node i and node j in one-hop is defined as l_{ij}^* ($l_{ij}^* = r$). In the network with random distribution, the distance between node i and node j in one-hop is defined as l'_{ij} ($l'_{ij} \leq r$).

Lemma 2: The network with random distribution generates more error compared to the network with ideal distribution.

Proof: By **Definition 2**, the difference of the distance between node i and node j in the ideal distribution and in the random distribution is expressed as

$$\begin{aligned} l_{ij}^* - l'_{ij} &= r - l'_{ij} \\ &= e_{ij} \geq 0, (i, j = 1, \dots, k, i \neq j) \end{aligned} \quad (17)$$

The average hop distances of network in the ideal distribution and the random distribution are defined as

$$\left\{ \begin{aligned} HopSize_i^* &= \frac{\sum_{i \neq j} l_{ij}^*}{\sum_{i \neq j} h_{ij}} = \frac{\sum_{i \neq j} r}{\sum_{i \neq j} h_{ij}} \\ HopSize_i &= \frac{\sum_{i \neq j} l_{ij}}{\sum_{i \neq j} h_{ij}} = \frac{\sum_{i \neq j} (r - e_{ij})}{\sum_{i \neq j} h_{ij}} \end{aligned} \right. \quad (18)$$

and their difference is given by

$$HopSize_i^* - HopSize_i = \frac{\sum_{i \neq j} e_{ij}}{\sum_{i \neq j} h_{ij}} \geq 0 \quad (19)$$

Hence, we can draw the conclusion that the error of estimated distance in the network with random distribution is larger than in the network with ideal distribution. However, both the BDV-Hop algorithm and most of the improved DV-Hop algorithms assume that the network is one with the ideal distribution.

Above all, the error caused by estimated distance has a major impact on the localization accuracy of the BDV-Hop algorithm and its improved versions. It is crucial to design improvements to estimate the distance.

3.3.2 Error of Fitness Function

A fitness function is used to evaluate the localization accuracy of the BDV-Hop algorithm. In (6), (7) and (8), we can infer that the fitness value is mainly determined by the distance error ε . Therefore, the fitness function in (8) can be expressed as

$$F(x, y) = \sum_{i=1}^k \varepsilon_i^2 \quad (20)$$

The solution in (8) is a potential optimal solution, which means that the sum of $\varepsilon_1^2, \varepsilon_2^2, \dots, \varepsilon_k^2$ is the minimum value. Therefore, the fitness value of the optimal solution can be described as

$$F^*(x, y) = \text{Min}\left(\sum_{i=1}^k \varepsilon_i^2\right) \quad (21)$$

$$= \text{Min}(\varepsilon_1^2 + \varepsilon_2^2 + \dots + \varepsilon_k^2)$$

However, if the value of $\varepsilon_i (1 \leq i \leq k)$ is quite large or quite small, the evaluation of the optimal solution will be seriously affected. This condition results in a non-optimal solution replacing the actual optimal solution.

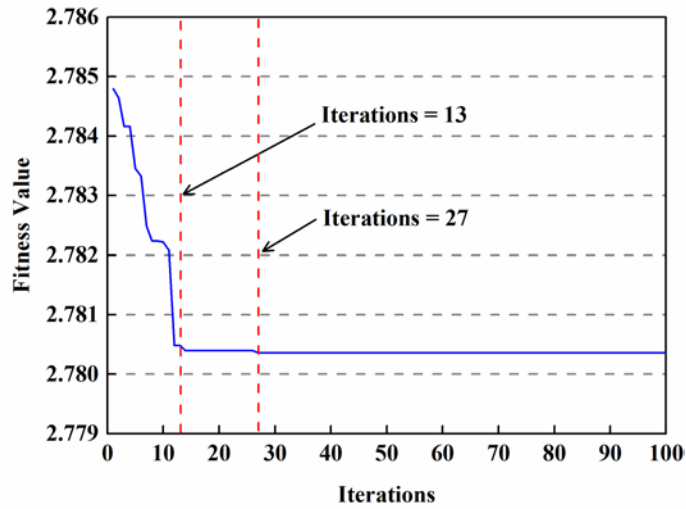


Fig. 4. The changes of fitness value with the number of iterations

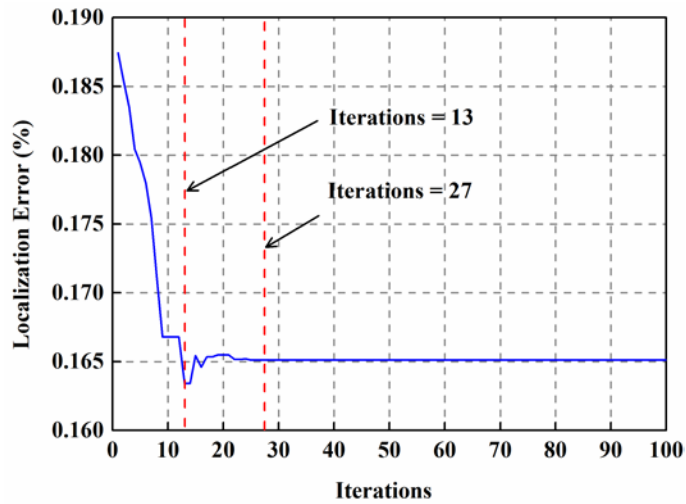


Fig. 5. The changes of localization error with the number of iterations

Fig. 4 and **Fig. 5** show an example of the error caused by the fitness function. The process of searching for the optimal solution is depicted in **Fig. 4** and the changes of actual localization error are illustrated accordingly in **Fig. 5**. In **Fig. 4**, the fitness values decrease gradually as the number of iterations increases. After 27 iterations, the fitness value converges to a fixed value; consequently, the algorithm is converged. According to the fitness function and value for the BDV-Hop algorithm shown in (20) and (21), the solution found after these 27 iterations is the optimal solution. However, **Fig. 5** shows that at iterations 13, the localization error has already

reached the minimum value. Although the iterations continue, the localization error is already minimized by iterations 13. In other words, the localization performance cannot be precisely evaluated by the fitness function in the BDV-Hop algorithm. Accordingly, the error caused by the original fitness function seriously affects the evaluation of the optimal solution.

3.3.3 Premature Convergence

Algorithm convergence is closely related to the updated methods of the solutions. Because it lacks an effective variation mechanism and updated solutions strategy, the BDV-Hop algorithm results in poor convergence.

The new solution of the BDV-Hop algorithm in (10) is generated with a random perturbation that is determined by the average volume of the bat group. However, if most of the bats tend to the local optimal solution in one iteration, the perturbations are unlikely to contribute to producing new solutions. In this situation, the updated method of the BDV-Hop algorithm has little impact on the diversity of solutions and the premature convergence is still quite likely to occur. Thus, the lack of an effective variation mechanism in the BDV-Hop algorithm makes it impossible to consistently guarantee a high localization accuracy. In addition, the local solutions of past iterations are abandoned in the BDV-Hop algorithm. For each iteration, the local solution will contain available and useful localization information that is conducive to searching for the optimal solution and avoiding premature convergence.

Consequently, a more effective variation mechanism and better updated method should be designed to avoid premature convergence and increase the diversity of the solutions of the BDV-Hop algorithm.

4. Proposed IBDV-Hop Algorithm

In this section, four optimization methods are designed for the IBDV-Hop algorithm. These optimization methods aim at improving the deficiencies of the BDV-Hop algorithm as discussed above. The time complexity and energy consumption of the IBDV-Hop localization algorithm are also analyzed.

4.1 Improvement of Average Hop Distance

Because the locations of the anchor nodes are known in advance, we can calculate the actual distance among the anchor nodes to improve the average hop distance.

The estimated distance is given in (2). The entire distance error E , among the anchor nodes is expressed as

$$E_{ij} = \sum_{i \neq j} (d_{ij}^* - d_{ij}) \quad (22)$$

where $d_{ij}^*(i, j=1, 2, \dots, k)$ is the actual distance between anchor node i and anchor node j , and d_{ij} is the estimated distance. Then, the error of the average hop distance is expressed as

$$\bar{E}_{ij} = \frac{E_{ij}}{h_{ij}} = \frac{\sum_{i \neq j} d_{ij}^* - d_{ij}}{h_{ij}} \quad (23)$$

To reduce the error of estimated distance, a normalized weight factor is introduced to revise the average hop distance. When an anchor node records the number of hops between its location and that of an unknown node, the weight factor of the average hop distance can be normalized as

$$W_i = \frac{1/H_i}{\sum_{j=1}^k 1/H_j} \quad (24)$$

where H_i is the number of hops between the anchor node i and the unknown node, and H_j represents the hops between all anchor nodes and the unknown node.

In (24), we learn that the larger H_i is, the smaller the W_i is. According to **Lemma 2**, the estimated distance from large hops among anchor nodes produces greater error. Hence, the large hop needs a small weight factor and small hops needs a large weight factor. Accordingly, the average hop distance in (1) can be improved as

$$HopSize'_i = \sum_{i=1}^k W_i (HopSize_i - \bar{E}_{ij}) \quad (25)$$

From (25), the anchor nodes that are near the unknown node have large weight factors and greatly affect the average hop distance while those anchor nodes far from the unknown node only weakly influence the average hop distance. Consequently, the error caused by the estimated distance will be reduced.

4.2 Improvement of Fitness Function

The error caused by the fitness function is expressed in (20) and (21). To reduce this error, it is necessary to reduce the influence of ε_i^2 values that are quite large or quite small in (21).

Definition 3: The value p denotes the reliability of each anchor node. Then, the distance error can be improved as

$$\varepsilon_i'^2 = \varepsilon_i^2 p_i \quad (26)$$

where $p_i (0 \leq p_i \leq 1, i = 1, 2, \dots, k)$ is the reliability of the anchor node i .

We introduce the root mean square (RMS) value to evaluate the distance error of anchor nodes, which is given by

$$\varepsilon_{rms} = \sqrt{\frac{\sum_{i=1}^k \varepsilon_i'^2}{k}} \quad (27)$$

In addition, the reliability p_i is modeled as

$$p_i = \begin{cases} p_1, & |\varepsilon_i - \varepsilon_{rms}| \leq \varepsilon_0 \\ p_2, & \varepsilon_0 < |\varepsilon_i - \varepsilon_{rms}| \leq 2\varepsilon_0 \\ p_3, & 2\varepsilon_0 < |\varepsilon_i - \varepsilon_{rms}| \leq 3\varepsilon_0 \\ p_4, & |\varepsilon_i - \varepsilon_{rms}| > 3\varepsilon_0 \end{cases} \quad (28)$$

where ε_0 is a threshold. The improved fitness function then becomes

$$F'(x, y) = \sum_{i=1}^k |\varepsilon_i'^2| p_i \quad (29)$$

In this case, an anchor node that produces a quite large or quite small distance error has low reliability. Therefore, the evaluation of solutions by the improved fitness function will be more precise than the original fitness function in (8).

4.3 Correction of Velocity

Because of its lack of an effective variation mechanism and updated solutions strategy, the BDV-Hop algorithm is likely to become trapped in local optimal solutions. To balance the global search capability and the local search accuracy, a nonlinear dynamic inertial weight

strategy is given by

$$v_i' = w_i v_i^{t-1} + (x_i^t - x^*) f_i \quad (30)$$

$$w_i = \begin{cases} w_{\min} - \frac{(w_{\max} - w_{\min})(F_i' - F_{\min}')}{(F_{\text{avg}}' - F_{\min}')}, & F_i' \leq F_{\text{avg}}' \\ w_{\max}, & F_i' > F_{\text{avg}}' \end{cases} \quad (31)$$

where w_i is the nonlinear dynamic inertial weight of bat i , F_i' is the fitness value in the current iteration and F_{avg}' is the average fitness value. We can deduce from (31) that the nonlinear dynamic inertial weight w is affected by the fitness value F_i' . Therefore, a bat with a large fitness value has a small inertial weight and vice versa.

When the bats converge to a local optimal solution, their fitness values decrease and their inertial weights increase. Bats with large inertial weights help to extend the search scope and increase the diversity of the solutions. In contrast, bats with small inertial weights weaken the influence from large fitness values during the evaluation of the optimal solution, which serves to further optimize the local optimal solution. Hence, the nonlinear dynamic inertial weight strategy contributes to improving the search accuracy and increasing the diversity of solutions.

4.4 Updated Solutions Strategy

Because premature convergence severely affects the localization performance of the BDV-Hop algorithm, a new updated solutions strategy is established in IBDV-Hop algorithm. In each iteration, the fitness value of the current local optimal solution is recorded. If the differences among the fitness values in the last c iterations are quite small, update the bat group once as

$$\begin{cases} v_i'' = w_i v_i' + c_1(\hat{x} - x_i') + c_2(x^* - x_i') \\ x_i'' = x_i' + v_i'' \end{cases} \quad (32)$$

where $c_1(0 < c_1 < 1)$ and $c_2(0 < c_2 < 1)$ are random variables, x_i^t is the location of bat i at time t , x^* is the current global optimal solution and \hat{x} is the last local optimal solution. In (32), the bat group is revised considering both the local optimal solutions of previous iterations and the current global optimal solution. When the bat group is about to achieve local convergence, the updated strategy will introduce variations in some solutions, thereby avoiding premature convergence and improving the probability of obtaining better solutions. Meanwhile, the updated solutions strategy is also beneficial for promoting the diversity of the bat group and improving the localization accuracy.

4.5 Algorithm Flow

The flow of the IBDV-Hop algorithm is as follows.

Step 1: Initialize the network in accordance with the schemes described in the location information acquisition for the BDV-Hop algorithm. Record the hops among nodes.

Step 2: Calculate the estimated distance and actual distance among the anchor nodes in (1) and (2). Calculate and improve the average hop distance using (25).

Step 3: Generate the bat group randomly. Determine the optimal fitness value and the local optimal solution \hat{x} for the bat group using (8) and (21).

Step 4: Update the bat group. Adjust the pulse frequency f and the location x in (9) and update the velocity v in (30) for each bat.

Step 5: Generate a random number $rand(0 < rand < 1)$. Compare the pulse emission rate r_i

with $rand$. If $rand > r_i$, go to **Step 6**, otherwise, generate a new solution x'_i using (10) to replace the former solution x_i .

Step 6: If $rand > A_i$ & $f(x_i) < f(\hat{x})$, let the x_i replace the local optimal solution \hat{x} , and update the pulse emission rate r_i and the volume A_i using (11).

Step 7: After each bat has been updated, go to **Step 8**, otherwise go to **Step 4**.

Step 8: Compare the local optimal solution \hat{x} with the global optimal solution x^* . If the former is better, the local optimal solution is the new global optimal solution.

Step 9: Evaluate the local optimal solutions for the previous c iterations. If the differences between the local optimal solutions and the current global optimal solution are continuously lower than a given threshold ε_0 , update the bat group once using (32).

Step 10: If the iterative termination condition is fulfilled, go to **Step 11**. Otherwise, go to **Step 4**.

Step 11: Export the global optimal solution x^* and terminate the IBDV-Hop algorithm.

At this point, x^* is the estimated location of the unknown node.

4.6 Analysis of Algorithm Complexity

Assuming that n nodes and k anchor nodes are randomly deployed in a network, then the dimension of the search space is 2, the size of the bat group is m , the maximum iterations is T and the interval of the updated solutions strategy is c .

In the network initialization stage of the DV-Hop algorithm, the time complexity required for hop counting is $O(n*n)$. The time complexity generated by the calculation and improvements of the average hop distance is $O(k*(n-k))$, and the time complexity for calculating the fitness value is $O(m*k)$. The processes to update the pulse frequency f , velocity v , solution x and the nonlinear dynamic inertial weight w in IBDV-Hop algorithm require a time complexity of $O(T*m)$. Meanwhile, the maximum time complexity for the processes of updated the pulse emission rate r , volume A and the optimum solution are less than $O(T*m)$. In addition, the time complexity required for updated solutions strategy is $O(T*(m/c))$ and for the other statements it is $O(T*m)$. In total, the time complexity of IBDV-Hop algorithm is $O(n*n) + O(k*(n-k)) + O(m*k) + O(T*(m/c)) + O(T*m)$. Because the maximum number of iterations is much larger than the size of bat group, the time complexity of the IBDV-Hop algorithm can be reduced to $Max\{O(T*m), O(n*n)\}$. **Table 1** lists the time complexity of the DV-HOP, BDV-Hop and IBDV-Hop algorithms. By adjusting the maximum number of iterations and the size of the bat group, the time complexity of the IBDV-Hop algorithm and the DV-Hop algorithm have the same order of magnitude.

Table 1. Algorithm complexity

Localization algorithm	Time complexity	Space complexity
DV-Hop	$O(n*n)$	$O(1)$
BDV-Hop	$O(T*m)$	$O(m)$
IBDV-Hop	$O(T*m)$	$O(m)$

Table 1 also lists the space complexity of the DV-HOP, BDV-Hop and IBDV-Hop algorithms. The temporary storage space occupied by the operational process of the IBDV-Hop algorithm is positively correlated with the bat group. Hence, the space complexity of IBDV-Hop algorithm is $O(m)$, and it is determined by the size of the bat group.

4.7 Analysis of Energy Consumption

The energy consumed in IBDV-Hop algorithm is mainly concentrated in the stages of network initialization and hop counting between nodes, anchor nodes and the unknown node. The same is true of the DV-Hop and BDV-Hop algorithms. Moreover, the proposed improved methods in the IBDV-Hop algorithm are carried out in the base station, which has powerful computing capability and no strict requirements for saving energy. Therefore, the IBDV-Hop algorithm requires little additional node energy consumption compared to the DV-Hop and BDV-Hop algorithms, and the improvements in IBDV-Hop algorithm are both reasonable and feasible.

5. Simulation Results and Discussion

To demonstrate the localization performance of our proposed IBDV-Hop algorithm, we performed a variety of simulation experiments to compare it with the DV-Hop, PSODV-Hop [21], IPSODV-Hop [22] and BDV-Hop algorithms. In the simulation experiments, nodes with the same characteristics are distributed randomly in a $100\text{ m} \times 100\text{ m}$ square area. The default settings of the parameters are listed in Table 2. Each simulation experiment is randomly simulated 50 times on the MATLAB 2012b platform. The localization error is defined as

$$\text{error} = \frac{\sum_{i=1}^k \sqrt{(x_i - \bar{x}_i)^2 + (y_i - \bar{y}_i)^2}}{kr} \times 100\% \quad (33)$$

where (\bar{x}_i, \bar{y}_i) denotes the actual location of the unknown node i and (x_i, y_i) denotes the estimated location.

Table 2. Default settings of parameters

Symbol	Description	Value
k	Number of anchor nodes	5-50
r	Communication range	20-60 m
N	Maximum iterations	0-300
m	Size of bat group	20-160
A_0	Initial volume	0.25
r_0	Initial pulse rate	0.5
λ_a	Attenuation coefficient of volume	0.95
λ_f	Enhancement coefficient of search frequency	0.05
f_{\max}, f_{\min}	Maximum and minimum search frequency	2,0
w_{\max}, w_{\min}	Maximum and minimum velocity weight	0.9,0.4
\mathcal{E}_0	Threshold of fitness value	18
c_1, c_2	Random variables	0.6,0.4
p_1, p_2, p_3, p_4	Reliabilities	1,0.9,0.8,0.5
c	Update interval of solution	5

5.1 Algorithm Convergence

The convergence graphs for the PSODV-Hop, IPSODV-Hop, BDV-Hop and IBDV-Hop algorithms are shown in Fig. 6, where it is evident that the localization error of the four algorithms decreases as the number of iterations increases. As the iterations vary from 0 to 60, the localization error declines sharply. Subsequently, the decline in localization error levels off until the algorithms reach fixed convergence values.

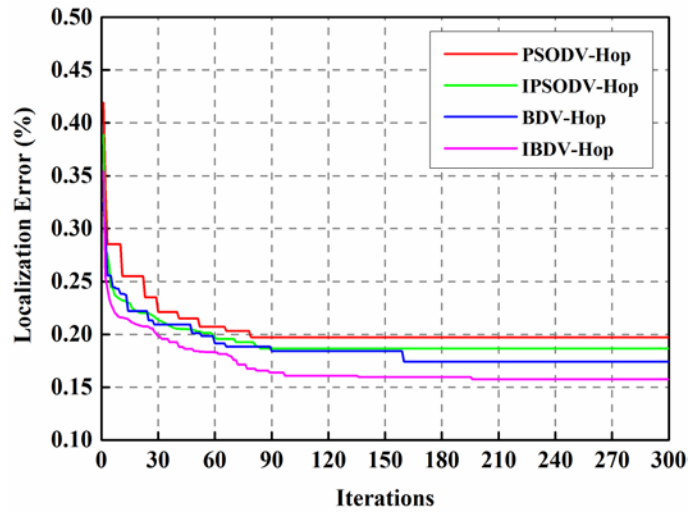


Fig. 6. Optimization processes of algorithms

Due to the fast local convergence of PSO, the PSODV-Hop algorithm converges rapidly and exhibits the largest localization error compared to other algorithms. Although the IPSODV-Hop algorithm has better search capability and convergence than the PSODV-Hop algorithm, its localization accuracy is still unsatisfactory. The localization error of the BDV-Hop algorithm is smaller than those of the PSODV-Hop and IPSODV-Hop algorithms. However, the curve of the localization error in the BDV-Hop algorithm is steep in Fig. 6. That is, the BDV-Hop algorithm has poor stability, and large variations in solutions occur as the number of iterations increases. In contrast, localization error curve for the IBDV-Hop algorithm is more stable than that of the BDV-Hop algorithm, and the IBDV-Hop algorithm achieves the smallest localization error among all algorithms.

Table 3. Convergence values of algorithms

Localization algorithm	Convergence value
DV-Hop	0.2467
PSODV-Hop	0.1971
IPSODV-Hop	0.1865
BDV-Hop	0.1742
IBDV-Hop	0.1575

Table 4. Statistical indicators for algorithms

Algorithms	Mean value	Variance	Standard deviation	Standard error
DV-Hop	0.2467	0.0425	0.2062	0.0292
PSODV-Hop	0.1986	0.0552	0.2349	0.0332
IPSODV-Hop	0.1857	0.0341	0.1847	0.0261
BDV-Hop	0.1762	0.0859	0.2931	0.0414
IBDV-Hop	0.1571	0.0249	0.1578	0.0223

The convergence values of the IBDV-Hop algorithm and the other algorithms are listed in Table 3. The smaller the convergence value is, the higher the localization accuracy is. We can conclude from Table 3 that the localization accuracy of IBDV-Hop algorithm is 36.2% higher than that of DV-Hop algorithm and is 9.6% higher than that of BDV-Hop algorithm. Clearly, the IBDV-Hop algorithm achieves the highest localization accuracy among all the tested

algorithms.

Some important statistical indicators of the simulation results for algorithms are listed in [Table 4](#), which shows that the dispersion of the results in IBDV-Hop algorithm is lower than that of the other algorithms. In other words, the IBDV-Hop algorithm is considerably more stable.

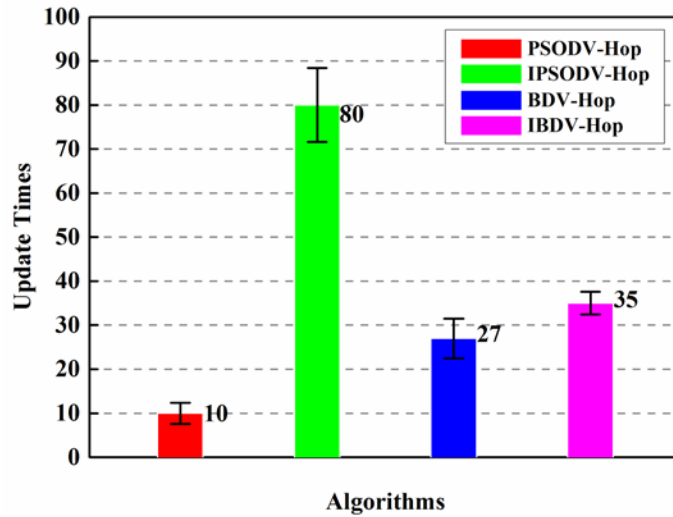


Fig. 7. Update times of optimal solution

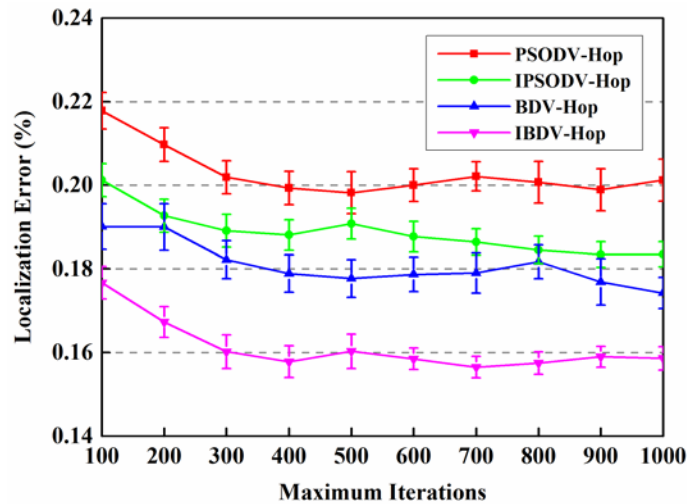


Fig. 8. Impact of the maximum number of iterations on the localization error

Fig. 7 provides the number of updates to the optimal solution for the tested algorithms. The PSODV-Hop algorithm updates the optimal solution only 10 times, while the IPSODV-Hop algorithm updates it 80 times, the BDV-Hop algorithm 27 times and the IBDV-Hop algorithm 35 times. Hence, the PSODV-Hop algorithm achieves the fastest optimization speed and requires the fewest updates to the optimal solution, but its localization error is also the highest. In contrast, the IPSODV-Hop algorithm has the slowest optimization speed and greatest number of updates to the optimal solution. The BDV-Hop algorithm and IBDV-Hop algorithm achieve moderate optimization speeds and updates to the optimal solution. From [Table 3](#), [Table 4](#), [Fig. 6](#) and [Fig. 7](#), we can conclude that the proposed IBDV-Hop algorithm has both

high localization accuracy and good stability at the cost of only a few updates to the optimal solution compared to other algorithms.

5.2 Impact of the Number of Iterations on Localization Accuracy

Fig. 8 describes the impact of the number of iterations on the localization error in four algorithms as the maximum number of iterations varied from 100 to 1000. **Fig. 8** shows that the localization error of the PSODV-Hop, IPSODV-Hop and IBDV-Hop algorithms decreases as the maximum iterations changes from 100 to 300; however, when the maximum number of iterations is greater than 300, there is almost no variation of the localization error in those three algorithms. The BDV-Hop algorithm is unaffected by an increase in the maximum number of algorithm. Finally, the curve of IBDV-Hop algorithm is smoother than that of BDV-Hop algorithm (see **Fig. 8**), showing that the IBDV-Hop algorithm is more stable than the BDV-Hop algorithm. In addition, the IBDV-Hop algorithm reveals higher efficiency than BDV-Hop algorithm when the number of iterations is below 300.

5.3 Impact of the Size of the Bat Group on the Localization Accuracy

We can assume that the function of the particle swarm in the PSODV-Hop and IPSODV-Hop algorithms is similar to that of the bat group in the BDV-Hop and IBDV-Hop algorithms. The impact of the size of the bat group on localization error is shown in **Fig. 9**, where we can see that increasing the size of the bat group reduces the localization error in all the algorithms. As the bat group size varies from 20 to 100, the localization error steadily decreases. As the size of the bat group changes from 100 to 160, the localization error in PSODV-Hop, IPSODV-Hop and IBDV-Hop algorithms remains almost unchanged. Therefore, we can infer that the size of bat group is conducive to reducing the localization error within a certain range, and that the bat group size should be set appropriately for each specific experiment to gain the maximum advantage from the IBDV-Hop algorithm.

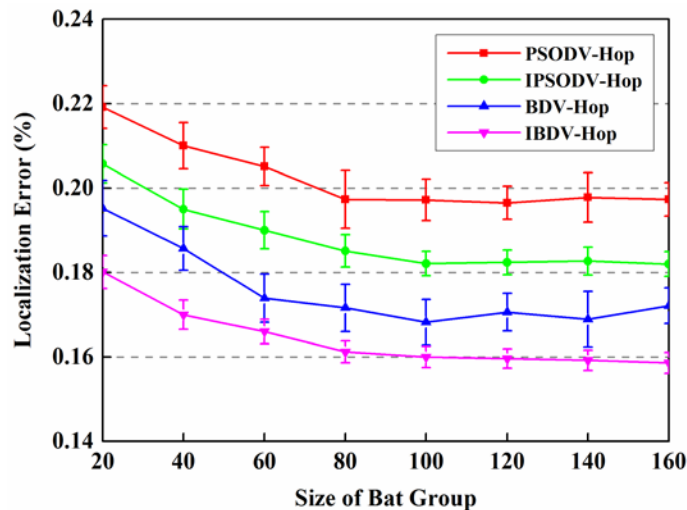


Fig. 9. Impact of the size of the bat group on the localization error

5.4 Impact of the Density of Anchor Nodes on Localization Accuracy

Fig. 10 depicts how the density of anchor nodes influences the localization error of the tested algorithms as the number of anchor nodes varies from 5 to 50. In **Fig. 10**, the density of anchor nodes has a significant effect on the localization error of the DV-Hop, PSODV-Hop and

IPSODV-Hop algorithms, and the localization error is markedly reduced as the number of anchor nodes increases from 5 to 25. In contrast, the localization error of the BDV-Hop and IBDV-Hop algorithms barely changes. Hence, the number of anchor nodes is almost insignificantly correlated with the localization accuracy for the IBDV-Hop algorithm. Compared to the DV-Hop, PSODV-Hop and IPSODV-Hop algorithms, the IPSODV-Hop algorithm is less dependent on the anchor node density.

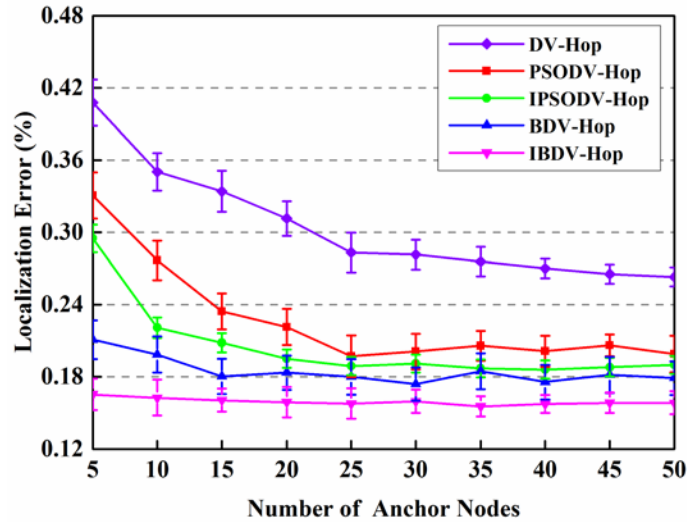


Fig. 10. Impact of the density of anchor nodes on the localization error

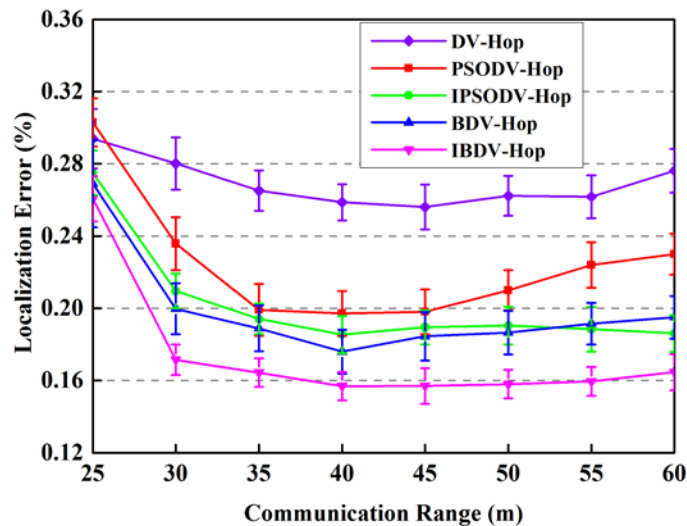


Fig. 11. Impact of communication range on localization error

5.5 Impact of Communication Range on Localization Accuracy

Generally, the actual communication range of RF devices is susceptible to interferences from different environmental situations. Therefore, we explored the impact of changing communication ranges on the localization accuracy of the tested algorithms. The simulation results are presented in Fig. 11 as the communication range varies from 25 to 60 m. Clearly, the localization error of all algorithms decreases as the communication range varies between

25 and 40 m. However, at a communication range of 40 m, all algorithms obtain the minimum localization error values, and as the communication range continues to rise, the localization error of all the algorithms also gradually rises. However, the localization error curve of the IBDV-HOP algorithm is more robust than that of the other algorithms as the communication range increases from 40 to 60 m. Thus, variations in the communication range have less effect on the localization accuracy in the IBDV-Hop algorithm. This implies that the IBDV-Hop algorithm is both stable and applicable to varied environmental conditions.

6. Conclusion

In this paper, we proposed the IBDV-Hop algorithm, based on the BA, to improve the localization accuracy of the DV-Hop algorithm. In the IBDV-Hop algorithm, the optimization methods for the average hop distance and fitness function effectively reduce the localization error and avoid error accumulation. The nonlinear dynamic inertial weight strategy to correct the velocity not only extends the global search scope and increases the accuracy of the local search but also contributes to searching for better solutions and further improves the localization accuracy. The updated solutions strategy provides a valid variation mechanism that avoids premature convergence and promotes the solution diversity. The theoretical analysis and simulation results demonstrate that the IBDV-Hop algorithm presents the highest localization accuracy compared to the other algorithms tested in this study. Its stability, search capability and convergence is better than that of the BDV-Hop algorithm. At the same time, the IBDV-Hop algorithm imposes little additional time complexity and energy consumption compared to the DV-Hop and BDV-Hop algorithms.

We conclude that the IBDV-Hop algorithm exhibits improved localization accuracy as the number of iterations and the size of bat group (within a certain range) increase. In contrast, the algorithm is only weakly affected by changes in the density of anchor nodes or node communication ranges. Therefore, the IBDV-Hop algorithm shows good stability and adaptability for localizing WSNs under different environmental conditions.

However, because the IBDV-Hop algorithm is sensitive to parameter values, future work should include more exploration of the relevance between the algorithm's performance and the values of crucial parameters under different circumstances.

References

- [1] V. Potdar, A. Sharif, and E. Chang, "Wireless sensor networks: a survey," in *Proc. of 2009 Int. Conf. on Advanced Information Networking and Applications Workshops*, pp. 636-641, May 26-29, 2009. [Article \(CrossRef Link\)](#).
- [2] A. Thakkar, K. Kotecha, "A new Bollinger Band based energy efficient routing for clustered wireless sensor network," *Applied Soft Computing*, vol. 32, pp. 144-153, July 2015. [Article \(CrossRef Link\)](#).
- [3] V. Gunes, S. Peter. T. Givargis et al., "A survey on concepts, applications, and challenges in Cyber-Physical Systems," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 12, pp. 4242-4268, December 2014. [Article \(CrossRef Link\)](#).
- [4] N. Iliiev, and I. Paprotny, "Review and comparison of spatial localization methods for low power wireless sensor networks," *IEEE Sensor Journal*, vol. 15, no. 10, pp. 5971-5987, October 2015. [Article \(CrossRef Link\)](#).
- [5] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28-34, October 2000. [Article \(CrossRef Link\)](#).

- [6] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proc. of -IEEE INFOCOM*, vol. 3, pp. 1655 - 1663, April 22-26, 2001. [Article \(CrossRef Link\)](#).
- [7] T. Vinh, N. Thu, and J. Minh, "A Lateration-localizing Algorithm for Energy-efficient Target Tracking in Wireless Sensor Networks," *Ad Hoc & Sensor Wireless Networks*, accepted for publication. [Article \(CrossRef Link\)](#).
- [8] Y. Shang, W. Ruml, Y. Zhang et al., "Localization from connectivity in sensor networks," *IEEE Transactions on Parallel & Distributed Systems*, vol. 15, no. 11, pp. 961-974, November 2004. [Article \(CrossRef Link\)](#).
- [9] M. Shona, M. Job, and H. Choo, "An interactive cluster-based MDS localization scheme for multimedia information in wireless sensor networks," *Computer Communications*, vol. 35, no. 15, pp. 1921-1929, September 2012. [Article \(CrossRef Link\)](#).
- [10] T. He, C. Huang, B. M. Blum et al., "Range-free localization and its impact on large scale sensor networks," *Acm Transactions on Embedded Computing Systems*, vol. 4, no. 4, pp. 877-906, November 2005. [Article \(CrossRef Link\)](#).
- [11] D. Niculescu, and B. Nath, "DV based positioning in Ad Hoc networks," *Telecommunication Systems*, vol. 22, no. 1, pp. 267-280, January 2013. [Article \(CrossRef Link\)](#).
- [12] Y. Wang, X. Wang, D. Wang et al., "Range-free localization using expected hop progress in wireless sensor networks," *IEEE Transactions on Parallel & Distributed Systems*, vol. 20, no. 10, pp. 1540-1552, October 2009. [Article \(CrossRef Link\)](#).
- [13] Y. Hu, and X. Li, "An improvement of DV-Hop localization algorithm for wireless sensor networks," *Telecommunication Systems*, vol. 53, no. 1, pp. 13-18, June 2013. [Article \(CrossRef Link\)](#).
- [14] S. Kumar, and D. K. Lobiyal, "An advanced DV-Hop localization algorithm for wireless sensor networks," *Wireless Personal Communications*, vol. 57, no. 2, pp. 1365-1385, July 2013. [Article \(CrossRef Link\)](#).
- [15] Y. Zhang, S. Xiang, W. Fu et al., "Improved normalized collinearity DV-Hop algorithm for node localization in wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 2014, no. 2, pp. 1-14, January 2014. [Article \(CrossRef Link\)](#).
- [16] N. A. M. Maung, and M. Kawai, "Experimental evaluations of RSS threshold-based optimized DV-HOP localization for wireless ad-hoc networks," *Electronics Letters*, vol. 50, no. 17, pp. 1246-1248, August 2014. [Article \(CrossRef Link\)](#).
- [17] P. Kristalina, Wirawan, and G. Hendranto, "Weighted hybrid localization scheme for improved node positioning in wireless sensor networks," *International Journal on Smart Sensing & Intelligent Systems*, vol. 6, no. 5, pp. 1986-2010, December 2013. [Article \(CrossRef Link\)](#).
- [18] X. Li, L. Yan, W. Pan et al., "Optimization of DV-hop localization algorithm in hybrid optical wireless sensor networks," *Journal of Heuristics*, vol. 21, no. 2, pp. 177-195, April 2014. [Article \(CrossRef Link\)](#).
- [19] L. Gui, T. Val, A. Wei et al., "Improvement of range-free localization technology by a novel DV-hop protocol in wireless sensor networks," *Ad Hoc Networks*, vol. 24, pp. 55-73, January 2015. [Article \(CrossRef Link\)](#).
- [20] S. Tomic, and I. Mezei, "Improvements of DV-Hop localization algorithm for wireless sensor networks," *Telecommunication Systems*, vol. 61, no. 1, pp. 1-14, March 2015. [Article \(CrossRef Link\)](#).
- [21] F. Zhang, "Positioning research for wireless sensor networks based on PSO algorithm," *Elektronika Ir Elektrotehnika*, vol. 19, no. 9, pp. 7-10, September 2013. [Article \(CrossRef Link\)](#).
- [22] Musong Gu, Yusong Yan, Lei You et al., "An improvement of localization algorithm based on particle swarm optimization and simulated annealing in wireless sensor networks," *J Inf Comput Sci*, vol. 10, no. 5, pp. 1497-1505, March 2013. [Article \(CrossRef Link\)](#).
- [23] Xiao Chen and Benliang Zhang, "Improved DV-Hop node localization algorithm in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 8 no. 8, pp. 1-7, August 2012. [Article \(CrossRef Link\)](#).

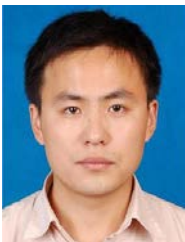
- [24] P. Bo, and L. Lei, “An improved localization algorithm based on genetic algorithm in wireless sensor networks,” *Cognitive Neurodynamics*, vol. 9, no. 2, pp. 249-256, January 2015.
- [25] [Article \(CrossRef Link\)](#).
- [26] F. Wang, C. Wang, Z. Z. Wang et al., “A hybrid algorithm of GA + Simplex method in the WSN localization,” *International Journal of Distributed Sensor Networks*, vol. 2015, no. 1, pp. 1-9, January 2015. [Article \(CrossRef Link\)](#).
- [27] X. S. Yang, “Bat algorithm: a novel approach for global engineering optimization,” *Engineering Computations*, vol. 29, no. 5, pp.464-483, November 2012. [Article \(CrossRef Link\)](#).
- [28] T. Niknam, R. Azizipناه-Abarghooee, M. Zare et al., “Reserve constrained dynamic environmental/economic dispatch: a new multiobjective self-adaptive learning bat algorithm,” *IEEE Systems Journal*, vol. 7, no. 4, pp. 763-776, May 2013. [Article \(CrossRef Link\)](#).
- [29] S. Mirjalili, S. M. Mirjalili, and X. S. Yang, “Binary bat algorithm,” *Neural Computing & Applications*, vol. 25, no. 3, pp. 663–681, September 2014. [Article \(CrossRef Link\)](#).



Yuan Liu is currently pursuing the Doctoral degree with School of Instrument Science and Engineering, Southeast University, China. His research interests include wireless sensor networks localization and intelligent information processing.



Junjie Chen is currently a Professor and Ph.D. Supervisor with Southeast University. His research interests include theory and applications of wireless sensor networks, technology and applications for system integrations of Internet of Things, theory and method of ubiquitous computing, intelligent sensing and control technology, control technology and applications of protected agriculture and intelligent agricultural machinery equipment.



Zhenfeng Xu is currently pursuing the Doctoral degree with School of Instrument Science and Engineering, Southeast University, China. His research interests include intelligent control and theory of wireless sensor networks.