Improved DV-Hop Localization Algorithm Based on Dynamic Anchor Node Set for Wireless Sensor Networks

Yuxiao CAO¹, Zhen WANG²*
¹School of Computer Engineering, Nanjing Institute of Technology, Nanjing, Jiangsu, 211167, China; ²School of Information Science and Technology, Huizhou University, Huizhou, Guangdong, 516007, China

Corresponding author: Zhen WANG (E-mail: wangzhen@hzu.edu.cn).

The research was supported by Research Fund of Nanjing Institute of Technology (CKJC201506), the Doctoral Research Fund of Huizhou University (2018JB007) and the Blue Fire Plan Foundation of China (No.2018#4).

ABSTRACT Node localization is a key issue in wireless sensor networks (WSN) area, and distance vector hop (DV-Hop) algorithm is widely adopted in WSN localization. Existing DV-Hop based algorithms employ all anchor nodes to localize the unknown node. However there is a large error of the estimated distance from the unknown node to some anchor nodes, which also results in the final unknown node localization error large. To improve the localization accuracy, we propose an improved DV-Hop algorithm based on dynamic anchor node set (DANS IDV-Hop). Differently to the existing DV-Hop based algorithms which apply total anchor nodes, DANS IDV-Hop utilizes part of anchor nodes to participate in localization. Firstly, the selection of anchor nodes is abstracted into a combinatorial optimization problem. For selecting appropriate anchor nodes, a novel binary particle coding scheme and fitness function are designed. Subsequently, the binary particle swarm optimization (BPSO) algorithm is applied to construct the dynamic anchor node set (DANS), and the localization is carried out on the DANS. Finally, the continuous particle swarm optimization (PSO) algorithm is utilized to further optimize the unknown node coordinates. Simulation results show that DANS IDV-Hop has excellent localization accuracy than that of the original DV-Hop and other DV-Hop based improved algorithms.

INDEX TERMS Internet of things, Wireless sensor networks, Localization, DV-Hop, Dynamic anchor node set, Particle swarm optimization

I. INTRODUCTION

The Internet of things (IOT) is the fastest growing and most promising branch of information technology in recent years. Wireless sensor networks (WSN) is a core technology of IOT [1, 2]. Usually WSN consists of a large number of sensor nodes deployed in the monitoring area. These nodes can measure surrounding physical parameters, perform simple calculation and storage, and transmit data to base station through wireless channel. WSN is widely used in many fields such as industrial control, target tracking, fire detection, smart home, intelligent transportation, battlefield reconnaissance [3-9], etc. In many WSN applications, sensed data must be combined with location information. Data without location content is often useless [10], hence node localization becomes one key issue for WSN.

The easiest way to localize the unknown node is to equip global positioning system (GPS) on each node. However, installation of GPS increases the cost of network. Moreover, the usability of GPS is not satisfactory enough in indoor and other complex environments. Therefore it is not universal to localize a node with GPS. Another way to localize sensor nodes is to deploy them manually and record the node coordinates. However, this solution is not general as there are a large number of nodes in the network and some monitoring areas are not feasible for human to enter for deployment. The commonly used node localization scheme is to obtain the coordinates of some nodes (called anchor nodes) by means of GPS or manual deployment, and then utilize localization algorithms to calculate the coordinates of other nodes (called unknown nodes).

Localization algorithms can be classified into two categories: range-based [11] and range-free [12], based on whether it is required to measure the actual distance or angle information between nodes. The range-based algorithms include: received signal strength indicator (RSSI)
[13], time of arrival (TOA) [14], time difference of arrival (TDOA) [15], angle of arrival (AOA) [16], and so on. The range-free algorithms include: centroid [17], distance vector hop (DV-Hop) [18], approximate point in triangle (APIT) [19], convex position estimation (CPE) [20], etc. The range-based algorithms have high accuracy but require extra hardware. Although range-free algorithms provide less accuracy compared to range-based schemes, they are still adopted in many applications, such as coal mine monitoring [21], patient tracking and positioning [22], port container positioning [23], because such algorithms do not require extra hardware and are simple to be implemented.

As the original DV-Hop algorithm has relatively large error [24, 25, 33] compared to range-based localization algorithms, how to improve the DV-Hop localization accuracy has both theoretical and practical significance. Theoretically, such attempt can provide some inspiration for future research. In practice, to improve positioning accuracy can make the existing DV-Hop based applications owning superior performance. In recent years, researchers have proposed a large number of improved schemes for DV-Hop. The improved DV-Hop algorithms can be classified into two categories. One kind of them focus on the second step of DV-Hop and correct the unknown node hop distance to a relatively accurate degree based on the network topology. The unknown node localization accuracy may be enhanced by improving the accuracy of its hop distance to anchor. The second kind of improved DV-Hop algorithms take the sum of the squares of the distance error from the unknown node to all anchor nodes as the optimization objective, and apply various optimization algorithms to optimize the unknown node coordinates which are calculated by DV-Hop.

Both improved algorithms mentioned above have increased the localization accuracy to a certain extent, however in terms of the improving mechanism, all of them ignored an important fact that in the third step of DV-Hop, the localization error will be lower if part of anchor nodes rather than all are employed. The reason is that if the error of estimated distance from the unknown node to some anchor nodes is large, the error of localization coordinates also becomes large [24, 25, 30, 33]. Therefore, if a number of anchor nodes which have smaller error of estimated distance to unknown nodes are selected to participate in localization, the accuracy may be improved.

In this paper, a novel improved DV-Hop algorithm based on dynamic anchor node set (DANS IDV-Hop) is proposed on the basis of experiments. The first and second steps of DANS IDV-Hop are same as that of the original DV-Hop. In the third step of DANS IDV-Hop, the binary particle swarm optimization (BPSO) algorithm is utilized to construct a dynamic anchor node set (DANS) to localize the unknown nodes. The particle swarm optimization (PSO) is further applied to correct the localization result in the fourth step of DANS IDV-Hop.

In this paper, we have three main contributions as follows.

1. We discover a phenomenon in the field of WSN that the localization error with part of anchor nodes will be smaller than that of applying all anchor nodes in the third step of DV-Hop.
2. We abstract the selection of anchor nodes into a combinatorial optimization problem. The binary particle coding scheme and fitness function are designed, and BPSO is utilized to construct the set of dynamic anchor nodes.
3. To further improve the localization accuracy, the continuous PSO is employed to optimize the coordinates of the unknown node.

The rest of this paper is organized as follows. In section II, related works are discussed. Section III describes the original DV-Hop algorithm. In section IV, PSO and BPSO are briefly explained respectively, and then proposed algorithm, DANS IDV-Hop, is presented. The simulation results are shown to demonstrate the effectiveness of the proposed algorithm in Section V. Finally, conclusions are drawn in Section VI.

II. Related works

Among WSN localization algorithms, DV-Hop is adopted widely due to its simplicity, feasibility and less hardware requirement. However, the accuracy of DV-Hop is not enough to satisfy stricter requirements, so researchers proposed a number of improved schemes based on DV-Hop. In this section, we briefly describe some DV-Hop based improved algorithms.

Gui et al. proposed a method of reference nodes selection and two improved algorithms for DV-Hop in [26]. Gui pointed out that the low accuracy of DV-Hop is due to its multi-hop nature and defective position calculation procedure. The authors investigated the third step of DV-Hop and found that the localization results are various if applying different anchor node as the last reference node of the least square method. Then two new algorithms based on DV-Hop, i.e., RAS DV-Hop and GOS DV-Hop, are presented. RAS DV-Hop takes each anchor node as the last reference node of the least square method to participate in the unknown node localization. In the network with $N$ anchor nodes, $N$ candidate localization results can be obtained, and the candidate coordinates which has the nearest distance to all anchors is selected as the final localization result. GOS DV-Hop applies a global optimization algorithm in the third step of DV-Hop, and it can obtain higher localization accuracy than RAS DV-Hop. However GOS DV-Hop also increases the complexity. That all anchor nodes are utilized as reference nodes, which will lead to some localization error expansion, can affect the final localization accuracy.

In [27], Kaur et al. considered that the high localization error of DV-Hop arises from the application of linear method such as the least square method to solve the nonlinear equations during the localization. Then, the authors proposed an enhanced DV-Hop algorithm which employs a non-linear technique named Gauss-Newton.
method to reduce the error of DV-Hop. The first two steps of proposed method are similar to that of the original DV-Hop. In the third step of localization, the mean value of coordinates of the three anchor nodes nearest to the unknown node are utilized as the estimated coordinates of the unknown node. Subsequently, Gauss-Newton method is applied to refine the coordinates of the unknown node. This approach improves the unknown nodes localization accuracy, however due to the estimated distance error of the unknown node, sometimes the selected three anchor nodes are not the nearest ones.

In [28], Tao et al. considered that different anchor nodes have various influence on the unknown node localization results after analyzing how the DV-Hop calculates the distance between nodes. The authors introduced the concept of weighted coefficient. In DV-Hop, weighted coefficient reflects the influence of the anchor node on the calculation of the unknown node coordinates. Weighted coefficient is related to the number of hop counts from the unknown to anchor nodes, and it decreases with the increasing of hop counts. When calculating the hop distance from the unknown to anchor nodes, the influence of weighted coefficient is considered, which promotes the estimated distance more accurate and improves the localization accuracy. Due to the diversity of network topology, the weighted coefficient is inaccurate in heterogeneous networks, which leads to the inaccuracy of estimated distance.

In [29], Peyvandi et al. proposed an improved DV-Hop algorithm based on hop distance correction and localization optimization. Firstly, the hop distance is calculated for whole network based on the difference between the estimated and actual distance of the anchor nodes. Secondly, the hop counts from the unknown to the anchor node is corrected by the received signal strength indicator. Finally, the unknown node location is optimized by the Levenberg-Marquardt algorithm. Although the accuracy is improved, this method requires extra hardware.

Cui et al. proposed a novel accurate localization algorithm based on DV-Hop and differential evolution called DECHDV-Hop for WSN in [30]. In DECHDV-Hop algorithm, the number of hop counts between nodes is changed from a discrete value to a continuous value, so that the estimated distance from the unknown node to different anchor nodes is not identical when the hop counts between them is the same. Due to the improved accuracy of the estimated distance, the accuracy of the localization results is also improved. Furthermore, DECHDV-Hop abstracts the localization process of the unknown node as an optimization problem that minimizes the weighted mean square error of the estimated distance from the unknown to the anchor node, and applies the differential evolution algorithm to solve the problem. It is an innovation that changes hop counts between nodes from discrete positive integers to continuous decimals in WSN localization.

In [31], in order to solve the problem of DV-Hop low accuracy, Song et al. presented two improved algorithms, that is, Hyperbolic-DV-Hop and improved weighted centroid DV-Hop localization algorithm (IWC-DV-Hop). In the second step of localization, Hyperbolic-DV-Hop calculates the average hop distance of the unknown node with the average hop distance of total anchors instead of the hop distance of the nearest anchor node. And Hyperbolic-DV-Hop utilizes hyperbolic localization algorithm instead of maximum likelihood estimation algorithm in the third step of the localization. Moreover, the IWC-DV-Hop algorithm reduces the calculation of the average hop distance of the unknown node. It directly employs the average hop distance of the anchor node to calculate the distance from the unknown to the anchor nodes. In the third step, the maximum likelihood estimation algorithm is replaced by the centroid location algorithm. Simulation results show that Hyperbolic-DV-hop and IWC-DV-Hop can improve the localization accuracy compared with the original DV-Hop.

In [32], Mehrabi et al. put forward an improved DV-Hop localization algorithm based on evolutionary algorithms. In the second phase of DV-Hop, the authors applied Shuffled Frog Leaping Algorithm (SFLA) to correct the average hop distance of the anchor node to reduce the localization error. When correcting the average hop distance, various anchor nodes were given different weights. The anchor nodes with larger influences on localization were assigned more weight. In the third stage, the hybrid genetic-particle swarm algorithm is applied to solve the problem of excessive localization error of the least square method. The localization accuracy has been improved, however control parameters of evolutionary algorithms need to be adjusted. And various parameters have a significant impact on localization accuracy. Though the algorithm shows good performance, the application of SFLA and hybrid Genetic-PSO make the algorithm spend more computational time.

In [33], an improved scheme named IDV-Hop using TLBO is proposed by Sharma et al. In the improved algorithm, the average hop distance of anchor node is corrected by using the correction factor, and the collinearity concept is introduced to reduce the localization error caused by the collinearity between nodes. To localize more unknown nodes, the unknown node successfully located in the first round is upgraded to the auxiliary anchor node. Finally, to further improve the localization accuracy, the coordinates calculation of the unknown node is abstracted as an optimization problem, and the parameter-independent optimization algorithm teaching-learning based optimization (TLBO) is applied to solve the problem. The introduction of collinearity can improve the localization accuracy, yet the average hop distance has a relatively large error while it is calculated with all anchor nodes.

In [34], Kaur et al. pointed out that the low accuracy of DV-Hop is due to the average hop distance error of anchor nodes, and then proposed two nature inspired algorithm-based improved variants of DV-Hop for randomly deployed 2D and 3D WSN. The first proposed algorithm applies Grey-Wolf optimization to optimize the average hop distance of anchor nodes.
distance of anchor nodes and improves the localization accuracy of the unknown node. Based on the first algorithm, the second one takes the reciprocal of hop counts which is from the unknown node to each anchor node as weight and optimizes the hop distance of the anchor nodes. Finally, the second improved algorithm applies the least square method to localize the unknown node. Although this algorithm considers that the low accuracy comes from the calculation of the average hop distance, it still utilizes all anchor nodes to calculate the anchor node average hop distance.

Singh et al. presented a PSO-based improved localization algorithm for WSN called PSODV-Hop [35]. The first two steps of PSODV-Hop are the same as that of the DV-Hop. In step 3, PSODV-Hop employs 2D hyperbolic algorithm instead of the least square method to estimate the location of the unknown nodes. In addition, PSODV-Hop adds a fourth step compared with the DV-Hop. In the fourth step, PSODV-Hop utilizes the PSO algorithm to correct the unknown nodes coordinates. Although PSO and 2D hyperbolic algorithm are applied in this method, the error introduced by anchor nodes with large estimated distance error cannot be completely eliminated.

It can be seen from the above literatures that all improved algorithms employ total anchor nodes to participate in the localization. If the estimated distance from the anchor to the unknown node is more accurate, the localization accuracy of the unknown node can be improved. Localization can be implemented when there are at least three non-collinear anchor nodes in plane. So, if some anchor nodes which have large estimated distance error are discarded and only part of them are applied for positioning, the localization accuracy can be improved.

III. DV-Hop algorithm
DV-Hop algorithm was firstly proposed by Niculescu et al. in 2003 [18]. Although the accuracy of DV-Hop is not entirely satisfactory, it has been adopted widely in many applications due to its simplicity, stabilization, feasibility and less requirements of hardware. For a better understanding of our proposed algorithm, the brief overview of original DV-Hop is presented as below.

Original DV-Hop consists of three steps.

A. Step 1: Calculation of hop counts between nodes
Every anchor node broadcasts a packet containing its location and hop counts initialized to 0 to the whole network. Each node \( N(\text{anchor or unknown}) \) in the network maintains a table in which the coordinates of all anchor nodes and hop counts from the node itself to all anchor nodes are stored. When a packet received from the same anchor node and the latest received hop counts is less than the value recorded in the table, the hop counts in the table is modified to the latest received one. Then, the packet is forwarded in the network with increased hop counts by 1. If the number of hop counts in the packet is greater than or equal to the hop counts in the table, the latest received packet is discarded. In this way, all nodes in the network know the minimum hop counts to any other anchor node.

B. Step 2: Estimation of the distance from the unknown node to the anchor nodes
After step 1, each anchor node received the information (coordinates and hop counts) of all other anchor nodes in the network. Then the \( i \)-th anchor node applies Eq. (1) to calculate the average distance per hop,

\[
\text{HopSize}_i = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \sum_i h_{ij}
\]

where \((x_i, y_i)\) and \((x_j, y_j)\) are the coordinates of anchor node \(i\) and \(j\) respectively, and \(h_{ij}\) indicates the hop counts from anchor node \(i\) to \(j\). After the calculation of average hop distance \(\text{HopSize}_i\), each anchor node broadcasts its \(\text{HopSize}\) to the network. The unknown node takes the first received \(\text{HopSize}\) as its average hop distance and transmits the \(\text{HopSize}\) to its neighbor nodes. The unknown node estimates its distance to an anchor node with Eq. (2),

\[
d_{ua} = \text{HopSize}_u \times h_{ua}
\]

where \(\text{HopSize}_u\) is the hop distance of the unknown node, \(h_{ua}\) represents the hop counts and \(d_{ua}\) indicates the estimated distance from the unknown node to the anchor node.

C. Step 3: Calculation of the unknown node coordinates
Let the coordinates of the unknown node \(u\) are \((x, y)\), the coordinates of the \(i\)-th anchor node are \((x_i, y_i)(1 \leq i \leq n)\), and the distance from the unknown node \(u\) to the \(i\)-th anchor node is \(d_i(1 \leq i \leq n)\). If there are at least three distance that have been calculated in step 2, the coordinates of the unknown node can be calculated with Eq. (3),

\[
\begin{align*}
(x - x_1)^2 + (y - y_1)^2 &= d_1^2 \\
(x - x_2)^2 + (y - y_2)^2 &= d_2^2 \\
&\vdots \\
(x - x_n)^2 + (y - y_n)^2 &= d_n^2
\end{align*}
\]

Subtract the last equation from the first \((n-1)\) equations, the Eq. (3) is transformed as follows,

\[
\begin{align*}
(x_1^2 - x_2^2) + (y_1^2 - y_2^2) - 2(x_1 - x_2)x - 2(y_1 - y_2)y &= d_1^2 - d_2^2 \\
(x_1^2 - x_3^2) + (y_1^2 - y_3^2) - 2(x_1 - x_3)x - 2(y_1 - y_3)y &= d_1^2 - d_3^2 \\
&\vdots \\
(x_n^2 - x_{n-1}^2) + (y_n^2 - y_{n-1}^2) - 2(x_n - x_{n-1})x - 2(y_n - y_{n-1})y &= d_n^2 - d_{n-1}^2
\end{align*}
\]

We rewrite Eq. (4) in the form \(AX = B\), where \(A, B\) and \(X\) are denoted by Eqs. (5) - (7).

\[
A = \begin{bmatrix}
    x_1 - x_n & y_1 - y_n \\
    x_2 - x_n & y_2 - y_n \\
    \vdots \\
    x_{n-1} - x_n & y_{n-1} - y_n
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
    x_1^2 + y_1^2 - x_n^2 - y_n^2 + d_1^2 - d_2^2 \\
    x_2^2 + y_2^2 - x_n^2 - y_n^2 + d_2^2 - d_3^2 \\
    \vdots \\
    x_{n-1}^2 + y_{n-1}^2 - x_n^2 - y_n^2 + d_{n-2}^2 - d_{n-1}^2
\end{bmatrix}
\]

\[
X = \begin{bmatrix}
x \\
y
\end{bmatrix}
\]

At last, the equation \(AX = B\) is solved with the least square method, and the location of the unknown node is obtained with Eq. (8).
\[ X = (A'A)^{-1}A'B \]  

where \( A' \) is the transpose of matrix \( A \), and \( A^{-1} \) is the inverse of matrix \( A \).

**IV. Proposed algorithm: improved DV-Hop based on dynamic anchor node set**

The proposed algorithm, DANS IDV-Hop, is presented in this section. Firstly, the principle of DANS IDV-Hop is described. Secondly, the BPSO algorithm which is applied for dynamic selection of anchor nodes is explained briefly. Thirdly, the PSO algorithm which is adopted for optimizing the final localization coordinates is described. Finally, the implementation of DANS IDV-Hop is described in detail.

**A. Principle of DANS IDV-Hop**

From Eq. (3), it can be known that as \( d_i \) is an estimated value, there must be an error on the right side of Eq. (3). The larger the estimation distance error from the unknown to the anchor node exists, the greater the final localization error occurs [24, 33]. According to the geometry, it is enough to locate an unknown node with the coordinates of three or more anchor nodes as well as the distance from the unknown node to them in the plane space. As mentioned above, we infer that the final positioning accuracy can be improved by eliminating anchor nodes which have large error of estimated distance to the unknown node. Fortunately, the experiments show that the above inference is feasible. The experiments demonstrate that if there are enough anchor nodes, higher accuracy can be obtained by only using part rather than all of them for localization. This can be explained by the following data.

Suppose that there is an unknown node \( U \) and 5 anchor nodes in a WSN, real coordinates of \( U \) are (12.20, 4.45). The nodes distribution is shown in Fig. 1. The coordinates of 5 anchor nodes, the estimated and actual distance from \( U \) to anchors, and the error of the estimated distance are listed in Table 1.

From Table 1, it can be seen that the error of estimated distance from \( A1, A2 \) to \( U \) is greater than that of \( A3, A4 \) and \( A5 \) to \( U \). If only the anchor nodes \( A3, A4 \) and \( A5 \) are utilized to localize \( U \), (11.72, 5.03) are the estimated coordinates. Otherwise, if all anchor nodes are applied, the estimated coordinates are (11.69, 3.14). The localization errors of above two cases are 7.5% and 14%, respectively. It can be seen that the proposed scheme has an excellent performance.

The existing improved DV-Hop algorithms either focus on the refinement of the estimated distance from anchor to unknown nodes [26-31], or apply various optimization algorithms to optimize the final localized coordinates in step 3 of DV-Hop [32-35]. However, these existing algorithms fail to recognize that the localization accuracy will be higher if part of the anchor nodes rather than all are used to localize. Although only one case is shown, the above experiments are universal proved by a large number of experiments.

The proposed algorithm bases on the dynamic selection of anchor nodes to participate in localization. Although which anchor node has lower error of the estimated distance is not known in advance, selecting part of anchor nodes can be abstracted as a combinatorial optimization problem [36]. Combinatorial optimization is an optimization approach to find the optimal arrangement, grouping, order or screening of discrete events through the study of mathematical methods. In fact, the best state is selected from a finite number of discrete states. In this paper, anchor node selection is abstracted as a combinatorial optimization problem as follows. \( M \) candidates from \( n \) anchor nodes are selected to minimize the localization error of the unknown node. BPSO algorithm is adopted to solve this combinatorial optimization problem. Our method with binary PSO is differently to the approaches with continuous PSO. We utilize binary PSO to select \( m \) from \( n \) anchor nodes to participate in localization, while other algorithms employ continuous PSO to optimize the final positioning results of DV-Hop. In the following, PSO and BPSO will be described briefly.

**B. Briefly introduction for PSO and BPSO**

Particle swarm optimization [37] (PSO) algorithm was firstly proposed by Eberhart and Kennedy in 1995. PSO is a branch of heuristic techniques, and it is motivated by societal behavior of bird flocks. For PSO, each particle represents a potential solution to an optimization problem

![FIGURE 1. The unknown and anchor nodes distribution.](image)

**Table 1 The anchor nodes information**

<table>
<thead>
<tr>
<th>Anchor node</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordinates</td>
<td>15.38, 5.64</td>
<td>1.41</td>
<td>10.66</td>
<td>6.26</td>
<td>9.43</td>
</tr>
<tr>
<td>Estimated distance to U</td>
<td>6.48</td>
<td>6.48</td>
<td>12.96</td>
<td>6.48</td>
<td>12.96</td>
</tr>
<tr>
<td>Actual distance to U</td>
<td>4.40</td>
<td>9.03</td>
<td>11.21</td>
<td>7.24</td>
<td>10.98</td>
</tr>
<tr>
<td>Error of estimated distance</td>
<td>47%</td>
<td>28%</td>
<td>16%</td>
<td>10%</td>
<td>18%</td>
</tr>
</tbody>
</table>
in a $D$ dimensional space. PSO algorithm defines a fitness function and makes the function value minimum (or maximum) as the optimization goal. Each particle has a position vector $X^k_i = (X^k_{1i}, X^k_{2i}, \ldots, X^k_{Ni})$ and a velocity vector $V^k_i = (V^k_{1i}, V^k_{2i}, \ldots, V^k_{Ni})$. Particle flies to the best personal and global extreme point of fitness function in the $D$ dimensional space. The velocity and position vector of particle in each iteration can be expressed as follows,

$$v^k_{id} = wv^k_{id-1} + c_1r_1(p_{id} - x^k_{id-1}) + c_2r_2(g_{id} - x^k_{id-1})$$ \hspace{1cm} (9)

$$x^k_{id} = x^k_{id-1} + v^k_{id}$$ \hspace{1cm} (10)

where $w$ is inertia factor which can keep the balance between the best personal and global function value; $c_1$ and $c_2$ are learning factors which represent the summary of the particle’s experience and the population’s experience; $r_1$ and $r_2$ denote random numbers between 0 and 1; $PB$ and $GB$ represent the best personal and global fitness function value respectively.

Standard PSO is suitable for continuous function optimization, however many practical applications are finally abstracted to combinatorial optimization problems. Hence, Kennedy and Eberhart proposed binary particle swarm optimization [38] (BPSO) in 1997. For BPSO, particle’s position is no longer continuous vector but discrete vector of 0 or 1 in binary space. In each iteration of BPSO, the position vector of the particle determines the probability of taking 0 or 1 in each dimension. Furthermore, the particle’s velocity vector updating formula is consistent with that of standard PSO.

The BPSO velocity is updated by Eq. (11),

$$v^k_{id} = wv^k_{id-1} + c_1r_1(p_{id} - x^k_{id-1}) + c_2r_2(g_{id} - x^k_{id-1})$$ \hspace{1cm} (11)

where every parameter has the same meaning as that in Eq. (9). The position is updated with Eq. (12),

$$x^k_{id} = \begin{cases} 1 & \text{if } S(V^k_{id}) < r(0,1) \\ 0 & \text{otherwise} \end{cases}$$ \hspace{1cm} (12)

where $S$ is a fuzzy function commonly used in neural networks, which is defined by Eq. (13),

$$S(V^k_{id}) = \frac{1}{1 + e^{-V^k_{id}}}$$ \hspace{1cm} (13)

For BPSO, the probability that a particle set to 0 or 1 is determined by Eq. (14),

$$P[X^k_{id} = 1] = S[V^k_{id}]$$ \hspace{1cm} (14)

C. Procedure of DANS IDV-Hop

In this section, the DANS IDV-Hop is presented in detail. DANS IDV-Hop can be divided into four steps: (1) calculation of hop counts between nodes; (2) estimation of distance from an unknown node to an anchor node; (3) localization of the unknown node based on DANS with BPSO; (4) further optimization of the unknown node estimated coordinates with PSO.

Our main contribution is that we construct a DANS to localize the unknown node. In DANS IDV-Hop, the DANS substitutes the set of all anchor nodes of original DV-Hop. The procedure of DANS IDV-Hop is shown in Fig. 2. As the first and second step of DANS IDV-Hop are the same as that of traditional DV-Hop, we only describe the third and fourth step of DANS IDV-Hop as Fig. 2.

D. Localization based on DANS

In this section, the key idea and innovation of our proposed algorithm, localization on DANS, is described as follows. Suppose WSN has $N$ anchor nodes and $K$ unknown nodes. When an unknown node $i$ is localized with DANS IDV-Hop, its localization error should be minimized by constructing a DANS. Based on section IV A, it must be existed such a set. However, it is impossible to determine which anchor node should be a member of the DANS only considering the average hop distance of each anchor node and hop counts from the anchor to the unknown node. But the localization accuracy can be evaluated by Eq. (15),

$$E = \sum_{i=1}^{N} \left( \sqrt{(x - x_i)^2 + (y - y_i)^2} - d_i \right)^2$$ \hspace{1cm} (15)

where $(x, y)$ and $(x_i, y_i)$ indicate the estimated and actual coordinates of the unknown node respectively, $d_i$ represents the estimated distance from the unknown node to the $i$-th anchor node.

Construction of DANS can be formulated into an optimization problem which can be solved with BPSO. It can be observed clearly that the smaller is the value of $E$, the more accurate are the estimated coordinates.

1) Binary particle coding

For PSO, position and velocity are two properties of particle, and position vector represents the solution of optimization problem. To solve the problem with PSO, the particle coding should be defined first, i.e., how to use the particle position vector to represent the solution of the optimization problem. Therefore, a novel binary particle...
coding scheme for DANS IDV-Hop is proposed firstly. For an anchor node, there are only two possibilities whether participating in the localization. For BPSO, the component of the particle position vector is either 1 or 0, hence the particle position vector can be used to represent the anchor node set. Here the following particle coding scheme is designed. Let the particles number of the population as $M$, that is, each unknown node has potential $M$ possible locations. If there are $N$ anchor nodes in the network, the dimension of the particle is set to $N$, and the position vector of the $i$-th particle can be denoted as $X_i = (X_{i1}, X_{i2}, \ldots, X_{iN})$. For the application scenario in this paper, the component $X_{ij} = 1$ denotes that the $j$-th anchor node is selected into the DANS, on the contrary $X_{ij} = 0$ denotes that the $j$-th anchor node is not selected, i.e., it does not participate in the positioning operation. To better illustrate the above methods, an example is presented as follows. For a network with 7 anchor nodes, if a particle's position vector is $(0,1,0,1,1,0,0)$, it means that the 2nd, 4th, 5th and 6th anchor node are selected to participate in the localization, while the rest of anchor nodes, i.e. the 1st, 3rd and 7th anchor are not taken part in the localization.

2) Fitness function design

The second key issue to apply BPSO is to design a fitness function which is utilized to evaluate the quality of solution in the iteration process. For all localization algorithms, it is impossible to know the exact unknown node coordinates while the location of the anchor node can be recognized in advance. For DANS IDV-Hop, the fitness function is illustrated as Eq. (16),

$$f(X) = \frac{\sum_{i=1}^{N} \left( \sqrt{(x - x_i)^2 + (y - y_i)^2} - d_i \right)^2}{K \times R}$$  \hspace{1cm} (16)$$

where $(x, y)$ is the estimated coordinates of the unknown node, $(x_i, y_i)$ denotes the $i$-th anchor node coordinates, $d_i$ represents the estimated distance from the unknown node to the $i$-th anchor node, $K$ stands for the number of anchor nodes, and $R$ means the communication radius of nodes. In the localization process, the $(x, y)$ that makes the fitness function get the minimum value is the optimal solution of the localization problem. In Eq. (16), applying all anchor nodes to calculate the fitness function is to use the evolutionary mechanism of BPSO to construct the DANS for the unknown node localization.

3) Localization with DANS

For DANS IDV-Hop, the DANS is applied to localize the unknown node, and for different unknown node the DANS is perhaps different. This is not same as original DV-Hop. The process of step 3 of DANS IDV-Hop is shown in Fig. 3.

To construct DANS with BPSO, all particles in the population should be initialized, that is, the velocity and position vector components of all particles should be defined. The dimension of particle is set to the number of anchor nodes in the network. The velocity component of each dimension is decided by a random number on $[0,1]$, and the position component of each dimension is determined by Eq. (14). According to the component of each dimension in the initial position vector, an anchor node corresponding to the component with a value of 1 is selected into DANS. After constructing the DANS, Eqs. (5) - (8) are utilized to calculate the estimated coordinates of the unknown node, and Eq. (16) is applied to calculate the fitness function value and determine the particle's individual and global optimum.

Iterative optimization is performed after initialization of the population. In the iteration process, the velocity vector of each particle can be calculated with Eq. (11), and the position vector can be calculated with Eqs. (12) - (13). A new DANS is constructed according to the new position vector of particle, and the unknown node is localized with Eqs. (5) - (8) on the new DANS. Here, just like the original DV-Hop, the least square method [39] is employed to calculate the unknown node coordinates. After calculating the unknown node coordinates in the iteration process, Eq. (16) is used to calculate the fitness function value and update the individual and global optimality of the particle. At the end of iteration, the estimated coordinates corresponding to the global optimum value are taken as the final estimated coordinates of the unknown node.

E. Optimization of the localized coordinates with continuous PSO

Although the accuracy of the unknown node coordinates calculated in step 3 with DANS IDV-Hop has been improved significantly compared with that of the original DV-Hop, step 4 has been implemented to further reduce the localization error with continuous PSO. When using continuous PSO to optimize the estimated coordinates of the unknown node, two problems should be solved firstly like BPSO: particle coding and fitness function design.
1) Particle coding of continuous PSO
When continuous PSO is employed to optimize the coordinates of an unknown node, as the position and velocity of particles are both continuous, the position vector of unknown node can be represented by a two-dimensional coordinates, so the particles position vector can be directly used to represent the unknown node coordinates, and the dimension of position and velocity vector is both set to 2.

2) Fitness function of continuous PSO
The evaluation rule of continuous PSO used for node localization is the same as that of BPSO. So Eq. (16) is also taken as fitness function of continuous PSO.

3) Optimization of the unknown node coordinates
The process of the unknown node coordinates optimization with continuous PSO is shown in Fig. 4. When continuous PSO is applied to optimize the unknown node coordinates, the initialization of the particles should be completed firstly, that is, the initial position vector of each particle are the coordinates calculated in step 3 of DANS IDV-Hop, and the velocity vectors are generated randomly in the velocity range. After the initial velocity and position vector are calculated, the best value of personal and global fitness function of each particle can be calculated with iteration.

After initialization, the velocity and position of the particles are generated according to Eqs. (9) - (10) respectively in each iteration. When one iteration is completed, new fitness function value is calculated with Eq. (16), and the particle's personal and global best value are upgraded. After all iteration process is completed, the location of the particle corresponding to the global best value is the estimated coordinates of the unknown node.

V. Simulation and performance evaluation
In order to verify the performance of DANS IDV-Hop, MATLAB R2016a is utilized to simulate it. Firstly, the simulation parameters and performance metrics are given in V A, and then the application of DANS is demonstrated in V B with the positioning process of an unknown node. Finally, the comprehensive performance of DANS IDV-Hop is evaluated in the rest of section V. In addition, the DV-Hop [18], the algorithm in [28] and [35] are chosen as comparison algorithms. As a representative of range-free localization algorithm, DV-Hop has always been applied by researchers as a benchmark for comparison. Algorithm in [28] and [35] are representatives of two types of algorithms mentioned in section I.

A. Simulation parameters and performance metrics
For all simulated localization algorithms, the nodes of WSN are located in a square area of 100 m * 100 m. The sensor nodes (anchor and unknown) are randomly deployed and evenly distributed. The simulation parameters are listed in table 2, which are the same ones as the parameters in [35]. The WSN area in the simulation remains unchanged, and three parameters vary within a certain range, i.e., the total number of nodes in the network, the proportion of the anchor nodes, and the communication radius. In the real environment, node residual energy, spatial location, signal direction, reflection, attenuation and other factors are different, so the communication radius of each node is not absolutely equal. In view of the fact mentioned above, the communication radius error of node in the simulation process is set to 0 - 10%, 0 - 20% and 0 - 30% respectively.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Border length</td>
<td>100 m</td>
</tr>
<tr>
<td>Number of sensor nodes</td>
<td>100 - 300</td>
</tr>
<tr>
<td>Ratio of anchor to all nodes</td>
<td>10% - 60%</td>
</tr>
<tr>
<td>Communication range</td>
<td>15 m - 60 m</td>
</tr>
<tr>
<td>Communication range error</td>
<td>0 - 30%</td>
</tr>
<tr>
<td>Initial population</td>
<td>20</td>
</tr>
<tr>
<td>Maximum iterations</td>
<td>50</td>
</tr>
<tr>
<td>w</td>
<td>0.8</td>
</tr>
<tr>
<td>c1</td>
<td>2</td>
</tr>
<tr>
<td>c2</td>
<td>2</td>
</tr>
</tbody>
</table>

The main performance metrics of localization algorithms is average localization error (ALE) which is illustrated in Eq. (17),

\[
ALE = \frac{\sum_{i=1}^{N-M} \sqrt{(x_{\text{est}} - x_{\text{act}})^2 + (y_{\text{est}} - y_{\text{act}})^2}}{R * (N - M)}
\]

where \((x_{\text{est}}, y_{\text{est}})\) and \((x_{\text{act}}, y_{\text{act}})\) indicate the estimated and actual coordinates of the \(i\)-th unknown node respectively, \(R\) means the communication radius of sensor nodes, \(N\) stands for the total number of sensor nodes, \(M\) denotes the number of anchor nodes. From Eq. (17), it can be seen that localization algorithms have higher accuracy with smaller value of \(ALE\). Moreover, from Eq. (17), it is obvious that the value of \(ALE\) is affected by the total number of nodes, the number of anchor nodes and the communication radius of nodes. Hence, we evaluate the performance of algorithms based on parameters as described as follows,
1. Total number of nodes;
2. Ratio of the number of anchor nodes to the number of all nodes;
3. Communication radius of sensor nodes.

B. The usage and advantage of DANS in localization

To illustrate the usage and advantage of DANS, the localization accuracy comparison of four algorithms, i.e., the original DV-Hop, the algorithm in [28], the algorithm in [35] and DANS IDV-Hop are described below. In the simulation, the total number of nodes is set to 100, the number of anchor nodes to 20, the node communication radius to 30 m without error. Nodes distribution is shown in Fig. 5 in which red star represents anchor node and black circle denotes the unknown node. The DANS, estimated coordinates, and the localization error of a randomly selected unknown node are listed in Table 3. All algorithms localization error of a randomly selected unknown node is listed in Table 4. All unknown node localization error of four algorithms are illustrated in Figs. 6-8.

![Fig. 5. Node distribution](image)

![Fig. 6. Localization error of each unknown node of DANS IDV-Hop and DV-Hop](image)

![Fig. 7. Localization error of each unknown node of DANS IDV-Hop and algorithm in [28]](image)

![Fig. 8. Localization error of each unknown node of DANS IDV-Hop and algorithm in [35]](image)

<table>
<thead>
<tr>
<th>Iteration number</th>
<th>Anchor node of DANS</th>
<th>Estimated coordinates</th>
<th>Actual coordinates</th>
<th>Localization error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2,4,7,12,13,14,15,16,18,20</td>
<td>(72.35,28.37)</td>
<td>(68.04,79.05)</td>
<td>56.50</td>
</tr>
<tr>
<td>2</td>
<td>4,7,12,13,14,15,16,18,19</td>
<td>(67.42,80.97)</td>
<td>(68.04,79.05)</td>
<td>3.32</td>
</tr>
<tr>
<td>3</td>
<td>4,7,12,13,14,15,16,18,19</td>
<td>(77.90,60.78)</td>
<td>(68.04,79.05)</td>
<td>7.40</td>
</tr>
<tr>
<td>4</td>
<td>7,11,12,13,14,15,16,17,18,19,20</td>
<td>(67.97,79.90)</td>
<td>(68.04,79.05)</td>
<td>0.31</td>
</tr>
<tr>
<td>5</td>
<td>7,11,12,13,14,15,16,17,18,19,20</td>
<td>(61.48,68.18)</td>
<td>(68.04,79.05)</td>
<td>13.43</td>
</tr>
<tr>
<td>6</td>
<td>7,11,12,13,14,15,16,17,18,19,20</td>
<td>(69.22,53.12)</td>
<td>(68.04,79.05)</td>
<td>20.43</td>
</tr>
<tr>
<td>7</td>
<td>7,11,12,13,14,15,16,17,18,19,20</td>
<td>(77.05,30.01)</td>
<td>(68.04,79.05)</td>
<td>59.83</td>
</tr>
<tr>
<td>8</td>
<td>7,11,12,13,14,15,16,17,18,19,20</td>
<td>(45.86,66.21)</td>
<td>(68.04,79.05)</td>
<td>11.69</td>
</tr>
<tr>
<td>9</td>
<td>1,6,7,11,14,15,16,17,18,19</td>
<td>(69.24,80.08)</td>
<td>(68.04,79.05)</td>
<td>20.43</td>
</tr>
<tr>
<td>10</td>
<td>1,6,7,11,14,15,16,17,18,19</td>
<td>(61.21,67.62)</td>
<td>(68.04,79.05)</td>
<td>13.99</td>
</tr>
<tr>
<td>11</td>
<td>1,6,7,11,14,15,16,17,18,19</td>
<td>(69.21,52.17)</td>
<td>(68.04,79.05)</td>
<td>21.77</td>
</tr>
<tr>
<td>12</td>
<td>1,6,7,11,14,15,16,17,18,19</td>
<td>(27.17,75.12)</td>
<td>(68.04,79.05)</td>
<td>33.82</td>
</tr>
</tbody>
</table>
Table 4 Localization error of single unknown node of four algorithms

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Localization error (%)</td>
<td>14.16</td>
<td>4.01</td>
<td>3.73</td>
<td>3.32</td>
</tr>
</tbody>
</table>

FIGURE 9. Average localization error with variation of total number of nodes (communication range is 30 m, range error is 0-10%, anchor ratio is 0.2)

FIGURE 10. Average localization error with variation of total number of nodes (communication range is 30 m, range error is 0-20%, anchor ratio is 0.2)

FIGURE 11. Average localization error with variation of total number of nodes (communication range is 30 m, range error is 0-30%, anchor ratio is 0.2)

It can be observed from table 3 that DANS IDV-Hop employs a DANS to localize the unknown node in each
iteration, while comparison algorithms utilize all anchor nodes. Meanwhile it can be observed from table 4 that DANS IDV-Hop has the smallest error compared to other three algorithms. It can be seen from Figs. 6 - 8 that although there are a few nodes whose localization error of DANS IDV-Hop is larger than that of other algorithms, DANS IDV-Hop has the most excellent performance in terms of mean localization error of all unknown nodes.

C. The effects of total number of nodes on localization accuracy

In order to verify the impact of total number of nodes on ALE, the ratio of anchor nodes is set to 20%, and node communication radius is set to a constant of 30 m. The number of nodes increases from 100 to 300. Simulation results in different scenarios with total number nodes varies from 100 to 300 are shown in Figs. 9-11, respectively. It can be seen that when the ratio of anchor nodes and the radius of node communication keep as constant, the localization error of algorithm in [35] and DANS IDV-Hop gradually decrease with the increase of total number of nodes. Moreover, it can be seen that the localization error of DV-Hop and algorithm in [28] decrease gradually before the total number of nodes increases to 200, and the error of both algorithms increase when total number of nodes is more than 200. This is due to the fact that the connectivity of network becomes better with the growth of the number of nodes. When hop counts from the unknown node to anchor nodes becomes smaller, which leads to the estimated distance more accurate, the localization error becomes smaller. However, when the total number of nodes exceeds a certain limit, the actual distance from the unknown node to anchor nodes may be already within one hop distance. In this situation, if the hop count is still set to 1, the estimated distance will be greater than the actual value, so the localization error will become larger. Obviously, it can be concluded that DANS IDV-Hop has the least localization error in each network environment.

Furthermore, it can be observed that in three scenarios the localization error of DV-Hop is decreased by -1%, -6% and -2% respectively when total nodes number increases from 100 to 300. At the same time, the localization error is decreased by -1%, 0% and -8% of algorithm in [28] respectively, and the localization error is decreased by 7%, 10% and 5% of algorithm in [35] respectively. However, the localization error of DANS IDV-Hop is decreased by 7%, 9% and 3% respectively. When total number of nodes increases, the accuracy of algorithm in [35] and DANS IDV-Hop both improves continually, but the accuracy of DANS IDV-Hop improves more smoothly. In each considered scenario, the best localization accuracy is provided by DANS IDV-Hop. It can be seen from Figs. 9 - 11 that when there are 300 nodes in the network, the localization error of DV-Hop is 29%, 37% and 35%, respectively; 20%, 32% and 29% localization error of algorithm in [28], 12%, 17% and 19% localization error of algorithm in [35], respectively; while it is about 11%, 16% and 18% localization error of DANS IDV-Hop, respectively. The simulation results show that the larger the error of node communication radius, the greater the error of all localization algorithms in the same scenario. From Figs. 9 - 11, it can be concluded that DANS IDV-Hop has the smallest localization error and the best performance in every scenario.

D. The effects of ratio of anchor nodes on localization accuracy

To evaluate the impact of ratio of anchor to total nodes on the localization accuracy, total number of nodes in the network is kept constant at 200 and the radius of node communication is set to 30 m. The ratio of anchor nodes increases from 10% to 60% with 5% as step size. Simulation results can be seen from Figs. 12 - 14. In this environment, it can be observed that the localization error of algorithm in [35] and DANS IDV-Hop decreases with the increasing of ratio of anchor nodes. It also can be observed that the localization error of DV-Hop and algorithm in [28] decreases in the first half and increases in the second half. This is due to the fact that when the total number of nodes of WSN remains unchanged, the network will own more anchor nodes with the growth of anchor nodes ratio. The increasing of anchor nodes can improve the connectivity and decrease the number of hop
counts from unknown to anchor nodes. As a result, the estimated distance from the unknown to anchor nodes becomes closer to the actual value. From mentioned in section II, it is known that the unknown node localization accuracy can be improved as the estimated distance becomes more accurate. However, when the number of anchor nodes increases to a certain extent, the error of estimated distance and localization becomes larger again because the actual distance from unknown to some anchor nodes is less than one hop distance. In Figs. 12 - 14, with the ratio of anchor nodes increasing from 10% to 60%, the error of DV-Hop is decreased by about 5%, 0% and 10% respectively.

At the same time, the localization error is decreased by about 1%, -2% and 3% of algorithm in [28] respectively, and the localization error of algorithm in [35] as well as DANS IDV-Hop is decreased by about 8%, 6% and 8% respectively.

Furthermore, when the ratio of anchor nodes reaches 60%, the localization error of DV-Hop is 30%, 36% and 35% respectively, 28%, 33% and 33% the localization error of algorithm in [28], 12%, 15% and 18% the localization error of algorithm in [35] respectively. However it is 11%, 14% and 16% of DANS IDV-Hop localization error respectively. Obviously, in each scenario, DANS IDV-Hop has the smallest localization error and the best performance.

E. The effects of nodes communication radius on localization accuracy

In order to evaluate the impact of nodes communication radius on the localization accuracy, total number of nodes in the network is kept constant at 200 and the ratio of anchor nodes is set to 30%. The radius of nodes communication increases from 10 m to 60 m. Simulation results can be seen from Figs. 15 - 17. It can be observed that the localization error of DV-Hop, algorithm in [28], algorithm in [35] and DANS IDV-Hop decreases with the increasing of nodes communication radius. When nodes communication range becomes wider, which leads to the number of hop counts from the unknown to anchor nodes reduction, the error of estimated distance from the unknown to anchor nodes becomes smaller and the localization accuracy is improved.

Furthermore, it can be observed that in three scenarios when node communication radius increases from 15 m to 60 m, the localization error of DV-Hop is decreased by about 14%, 24% and 29% respectively, about 12%, 18% and 15% the localization error of algorithm in [28], about 11%, 18% and 15% the localization error of algorithm in [35] respectively, while the localization error of DANS IDV-Hop is decreased by 12%, 17% and 13% respectively. At last, in three scenarios when node communication radius is 60 m, the localization error of DV-Hop is 29%, 26% and 26% respectively, 22%, 24%
and 23% the localization error of algorithm in [28], 18%, 16% and 20% the localization error of algorithm in [35] respectively, while the localization error of DANS IDV-Hop is 16%, 15% and 20% respectively. The simulation results show that DANS IDV-Hop has the best performance in all scenarios.

![Graph 1](image1.png)

**FIGURE 15.** Average localization error with variation of communication radius of sensor nodes (the number of sensors is 200, ratio of anchor nodes is 20%, range error is 0-10%)

![Graph 2](image2.png)

**FIGURE 16.** Average localization error with variation of communication radius of sensor nodes (the number of sensors is 200, ratio of anchor nodes is 20%, range error is 0-20%)

![Graph 3](image3.png)

**FIGURE 17.** Average localization error with variation of communication radius of sensor nodes (the number of sensors is 200, ratio of anchor nodes is 20%, range error is 0-30%)

### VI. Conclusion

Node localization is a key issue for WSN, which is needed by many applications. Among localization algorithms, DV-Hop is widely adopted because it does not require extra hardware. However the accuracy of DV-
Hop is not enough to satisfy some strict applications, hence a number of improved algorithms based on DV-Hop are presented. From the localization process of DV-Hop, we discovered that if only a part of anchor nodes are selected to participate in positioning, the accuracy becomes higher than that with total anchor nodes. So a novel improved DV-Hop algorithm based on dynamic anchor node set, i.e., DANS IDV-Hop, is proposed.

How to select part of anchor nodes and minimize the localization error can be formulated to a combinatorial optimization problem. In this paper, the binary particle coding and fitness function for DANS IDV-Hop are designed, and BPSO is applied to construct the DANS. The selection of anchor nodes is represented by binary value of particle position vector, which is the innovation of this paper. Afterwards, the DANS is applied to localize the unknown nodes. After the third stage of DANS IDV-Hop, the localization accuracy has been greatly improved with DANS. Finally, the localization results calculated in step 3 is further optimized with continuous PSO.

The simulation results show that the localization accuracy of DANS IDV-Hop is significantly improved compared with that of other comparison algorithms in various scenarios. It has proved that DANS IDV-Hop can provide a potential solution for accurate localization in WSN area.

Considering the complex situation of node localization of WSN, the DANS IDV-Hop has some shortcomings to overcome. For example, the execution time of DANS IDV-Hop is a little more than that of other comparison algorithms due to its multiple iterations. Future research should make full use of the previous results of intermediate iterations to reduce the computation load of DANS IDV-Hop.

REFERENCES


This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2938558, IEEE Access


**YUXIAO CAO** received the master degree in software engineering from University of Science and Technology of China (USTC), Hefei, China in 2007. Currently he is a teacher of school of computer engineering of Nanjing Institute of Technology, Nanjing, China. His research interests include wireless sensor networks, swarm intelligence and embedded system.

**ZHEN WANG** received the Ph.D. degree in computer technology and application from the Faculty of Information Technology, Macau University of Science and Technology (MUST), in June 2017. From 2006 to 2017, he was a teacher of Beijing Institute of Technology, Zhuhai. He joined the Huizhou University, Huizhou, China in October 2017. Currently, he is a teacher of School of Information Science and Technology of Huizhou University. His research interests focus on vehicular networks and intelligent transportation system.