Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 1194752, 12 pages
https://doi.org/10.1155/2022/1194752

Research Article

Improved DV-Hop Algorithm Based on Swarm Intelligence for AI and IoT-Federated Applications in Industry 4.0

Lizhi Zhang

College of Engineering and Technology, Xi'an Siyuan University, Xi'an 710038, China

Correspondence should be addressed to Lizhi Zhang; zhanglizhi@mjwhedu.cn

Received 11 April 2022; Revised 14 June 2022; Accepted 22 June 2022; Published 11 August 2022

Academic Editor: Mukesh Soni

In the Internet of Things (IoT) ecosystem, localization is critical for tracking and monitoring targets via nodes. The distance vector-hop (DV-Hop) technique is a good choice for localizing neighborhood in IoT networks. The conventional DV-Hop algorithm is a distributed localization approach that does not consider the distribution of the nodes into full deliberation when calculating the hop count from the source to destined nodes. The transfer distance and node positions thus do not attain higher efficiency while ascertaining the distance between sources and destined nodes. The study aims to resolve the pitfalls in the traditional algorithm by making enhancements in controlling the original DV-Hop algorithm's hop count and transfer distance method by utilizing the particle swarm to estimate the node positions. Error rate in the distance between beacon nodes and unseen nodes is effectively reduced with the proposed technique that calculate error factors with corrections in a reversed fashion to revise hop counts. An escape factor is introduced to take control of updating particles' velocity in the system, and the inertia weight is defined by a piecewise function to enlarge search space. This mechanism increases the diversity of the particle populations and mitigates the tendency of estimations on node positions to be trapped into local optima under stationary state. Also, the improved DV-Hop algorithm described in the paper has a better convergence speed due to the presence of random inertia weight logarithmic method. Finally, the problem of premature convergence is also tackled as a variation factor is adopted in collaboration with a fitness function that affects the particles' movement range and assists in global convergence. The overall performance of improved DV-Hop is evaluated by statistical metrics and also compared with the traditional DV-Hop algorithm under simulated environment with the data collected from real-world scenarios. Industry 4.0 is fully dependent upon IoT and the count of hops is very important for deciding the routing from the source to destination for speedy transmission of data. The improved DV-hop algorithm can achieve better results and has reduced error rate by more than 30%. The DV-Hop algorithm plays an important role in IoT-enabled environment especially in Industry 4.0.

1. Introduction

Wireless sensing networks (WSNs) have a large number of small sensing nodes, which means they have less compute, storage, and transmission capability. Wireless sensor networks are less expensive, have lower power consumption, and are self-configurable sensor nodes [1]. By considering these parameters, WSNs are used in areas like healthcare, for smart transportation, home monitoring, military tracking, environmental monitoring, national security purpose, and indoor navigation [2]. The sensing nodes sense the location and accordingly disseminate the data. For this, the position of the destination node needs to be detected. The detection of position becomes difficult due to the fluctuations of signal and noise in the environment. Many difficulties have been faced for location analysis [3]. Localization methods used in WSNs are independent of the previous localization position. They rely on the position data of a few specific sensor nodes as well as some inter-network measurements. Using Global Positioning System (GPS), the accurate and precise location can be found. However, this technique is impractical due to its high cost, increased power consumption, unavailability of signal, and inefficient performance [4]. Hence, there are many methods proposed in the literature to locate the position of node by exchanging data between the nodes.

Many localization approaches have been presented as a result of the location estimation problem. It is primarily classified as a range-assisted and range-free method [4]. To compute location among neighboring sensing, range-based localization technique uses count of hops, distance, or angle information which requires a higher cost to calculate distance. The range-based technique is based on pattern matching or connectivity measurements. It is categorized as (1) time of arriving (TOA), (2) time difference of arriving (TDOA), (3) received signaling strength indicator (RSSI), and (4) angle of arriving (AOA) [2]. These approaches provide good location accuracy, but they necessitate the use of hardware for location calculation. In contrast, the range-free technique does not require any hardware. The cost is reduced here because it does not use any hardware and also the power consumption is less. To calculate distance between nodes, it uses hop counts and distance approximation algorithm [5].

The range-free approach is further classified as follows: centroid algorithm, amorphous, DV-Hop method, multi-dimensional scaling (MDS) method, and approximate point in triangulation (APIT) method. The DV-Hop technique is simple and easy to implement in a range-free localization algorithm. As a result, the DV-Hop algorithm is the most often employed [3]. The DV-Hop algorithm reduces localization errors. This algorithm has its localization function. Using this function, it requests an anchor node, which provides information of the node position. Sensor node arranges anchor node and calculates position. This technique gives better scalability and distribution. If sensor node distribution is not uniform, it affects the accuracy. Because of this, the algorithm gives poor localization accuracy [4, 6]. So, the improvement in the existing DV-Hop algorithm has been done. The main contributions of the paper are as follows:

(i) Improved DV-Hop algorithm is proposed which is hybrid of the DV-Hop algorithm and RSSI measurements to enhance the accuracy of localizing nodes.

(ii) An RSSI-based ranging mechanism is used that predicts one-hop distance.

(iii) Levenberg–Marquardt approach is used to compute node position.

(iv) Analysis of DV-Hop algorithm is performed on parameters like mean hop distance, hop count, and node coordinates. Also, error analysis has been done to achieve accuracy.

(v) To minimize the calculation error, the algorithm is improved.

(vi) The formula for mean hop distance, the hop count, and particle swarm optimization has improved. The improved DV-hop algorithm has achieved better results and has reduced the error rate by more than 30%.

Rest of the paper is organized as follows: In Section 2, state of the art of the existing work is discussed. In Section 3, existing working approach of DV-Hop algorithm is elaborated. In Section 4, detailed implementation details of the proposed improved DVH algorithm with the parameters introduced and steps of implementation are elaborated. In Section 5, detailed simulation and experimental result analysis are discussed. In Section 6, the conclusion of the research work is elaborated.

2. Related Works

Many authors have contributed on the same problem statement by offering solutions based on fuzzy-based approaches, evolutionary algorithms, swarm intelligence-based algorithms, and machine leaning-based schemes.

In [7], the authors have addressed the localization technique used to compute the place of nodes using a collection of nodes known as anchors. The density collections of these anchors would be increased or decreased due to various reasons such as maintenance, breakdown, and lifetime. The DV-Hop (DVH) technique is appropriate for the positioning of nodes that consists of a few neighbor anchors. However, the existing DVH-based technique has not taken into account the issue of anchor failure, which can occur during a localization operation. To solve this issue, the authors have proposed an online sequential DVH (OS-DVH) algorithm which is used to calculate the localization of nodes sequentially and to enhance the position accuracy of the nodes for multi-hop WSN. DVH method is used to process node localization using an optimized approach for estimating the average distance of hops between nodes. In [8], the authors have proposed an advanced DVH technique. The authors have tried to lessen the error rate for the DVH technique in two ways. At first, it is equivalent to communication radius when the gap between the two hops is less. These hops are known as sequential hops. The distance between the hops is computed by the shadowing structure. The unknown nodes indicate that the hop size started from beacon nodes; however, the distribution of nodes in WSN is not equal.

In [9], the authors have presented an improved DVH algorithm used to boost the accuracy of the DVH algorithm without enhancement in the computational complexity. The authors presented two different algorithms. Firstly, the presented algorithms use the K-mean approach followed by the repositioning of nodes. The second algorithm also uses the K-mean approach but is followed by cluster division localization. The first algorithm was not used frequently due to certain conditions in the applications during the repositioning of the nodes. Hence, this algorithm cannot achieve accuracy optimally. That is why the advanced algorithm was used to evaluate the distance among hops and have shown better enhancement in accuracy. In [10], the authors have examined the issue of hop count info among nodes having a large influence on the localization accuracy of the standard DVH method. As a result, an advanced method based on RSSI was proposed to overcome the problem. In [11], the authors have proposed to enhance the version of DVH algorithms such as quadratic DVH (QDVH) and unconstrained DVH (UDVH) algorithm for the greater localization without the requirement of additional hardware for the measurement of range among nodes. The QDVH algorithm is utilized to reduce the error rate to generate higher localization, and the UDVH algorithm attains localization accuracy equivalent to QDVH. The proposed algorithms outperform than the existing DVH algorithm.

When applying the DVH algorithm to node position in WSN, the authors presented an iDVH algorithm in [12], which takes into account deprived precision localization. In this approach, the mean hop size of anchor nodes is enhanced by minimum mean square error (MSE) and is modified by the error factor. Then, the mean hop size among the unknown hops and anchor hop is improved through the dynamic load coefficient. In [13], the authors have discussed the two kinds of localization algorithms (LA) such as range-based LA and range-free LA. Range-based LA, on the other hand, has stringent hardware requirements, making it difficult to implement. In the case of range-free, however, it lowers the hardware cost. This practice is good for only known hops to evaluate accuracy. Hence, to evaluate better accuracy for unknown hops, a proposed RSSI-based DVH is proposed in this paper.

In [14], the authors have presented a DVH positioning algorithm to evaluate better positioning accuracy. The node position is determined based on the radio range of anchor nodes. In this paper, they have used the discovery probability technique for the evaluation of localization error. In [15], the authors have proposed the RMADVH (regular moving anchor DVH) algorithm to enhance the DVH algorithm based on the regular moving anchor node and RSSI range approach. The proposed localization algorithm utilizes a few anchor nodes, achieves distribution of hops equally, and reduces the hardware cost.

In [16], the authors have discussed the localization algorithm, which is of utmost proficiency because of its simplicity, low cost, and less complexity. But having the limitation is poor localization accuracy when the anchor nodes are reduced. Hence, this paper proposed a weighted DVH based on RSSI. In [17], the authors have proposed an improved DVH based on a dynamic anchor node-set (DANS DVH) to increase the accuracy of location. In this proposed algorithm, part of anchor nodes participates in localization, whereas in existing DVH algorithms complete nodes are applied. Binary particle swarm optimization (BPSO) algorithm is utilized to design DANS. In [18], the authors have a detailed analysis of the DVH algorithm which is not efficient in the evaluation of localization accuracy due to various constraints.

In order to overcome the drawbacks of existing approaches, this paper is introducing two new mechanisms such as the weighted averaged approach which is utilized to calculate each hop distance and estimation of unknown node distance and beacon nodes to estimate average of hop distance.

3. DV-Hop Algorithm

The hop counts operate as multipliers in most DV-Hop extensions' algorithms, applying to a normalized distance unit that represents the discrete form of real distance between nodes after quantification. When initializing the nodes, the minimum hop count of a beacon node gets initialized together with that of unseen nodes by simply checking connectivity according to the transmission range in the channel [19]. The current hop count is 1 if the two nodes can communicate or else is infinity (also, in implementations, mark the current hop count as 0 to indicate the nodes are identical). These connectivity-related meta-data are broadcast to the entire network, forming a topology model for future usage [20]. Then, iterate through the connection matrix using the minimum path method, updating the local hop count and sending it to the network. Finally, the number of hops to the destination node is calculated and logged as h_{ij}.

3.1. Mean Hop Distance. Firstly, after calculating the minimum hop count, the value is assigned to the nodes of the whole network through broadcasting, denoted as h_{ij}, and the value of correction nodes is calculated as given in

$$\begin{gathered} d_i = \frac{\sum_{j\neq i}\sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}}{\sum_{j\neq i} h_{ij}}, \\ k_{ij} = \left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2, \\ d_i = \frac{\sum_{j\neq i}\sqrt{k_{ij}}}{\sum_{j\neq i} h_{ij}}. \end{gathered} \tag{1}$$

After the unseen node receives the transfer distance of the nearest beacon node, it takes the product of the transfer distance obtained from that node to estimate its value to each beacon node as shown in equation (2) [21]. Then, the known nodes and target nodes are calculated by using this algorithm.

$$D_i = d_i \times h_{ij}. \tag{2}$$

Unseen node coordinates: The distance between two points is as given in

$$\begin{cases} \sqrt{\mathrm{k}_{11}} = d_1, \\ \sqrt{\mathrm{k}_{22}} = d_2, \\ \vdots \\ \sqrt{\mathrm{k}_{33}} = d_n, \end{cases} \tag{3}$$

where (x, y) represents the distance between the unknown points.

Equation (3) uses the first $n-1$ terms to subtract the n-th term convertible least squares difference to estimate the unseen node coordinates as shown in

$$\mathbf{X} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{B}. \tag{4}$$

Among them,

$$\mathbf{A} = \begin{bmatrix} (x_1 - x_n) & M & (x_{n-1} - x_n) \\ (y_1 - y_n) & M & (y_{n-1} - y_n) \end{bmatrix},$$
$$\mathbf{B} = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_n^2 - d_1^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + d_n^2 - d_{n-1}^2 \end{bmatrix}, \quad (5)$$
$$\mathbf{X} = [x, y]^T.$$

3.2. DV-Hop Algorithm Error Analysis. Many methods use a correction factor to change the jump distance in order to increase the robustness [21], but the errors of the transfer distance are not weighted. Hence, the improvement is relatively large and has unstable positioning error rate. In other works, an error factor is used to improve the weight; however, since it did not arrange the problem into the early, middle, and later stage based on the particle swarm's properties, in the process of calculation, thus the global optimal position cannot be found. By assistance from some normalization and correction techniques on hop count and hop distance, the coordinates of nodes can be calculated more accurately [22].

3.2.1. Hop Count Error. The hop count is a number that represents the value of how many aliquots a number can be divided into. Previous methods assumed that the distribution of nodes was even enough to ignore disparities in distance per hop between node pairs. Thus, it simply counts every direct communication between two nodes as 1 hop, regardless of the actual distance. Therefore, the errors in hop count get accumulated in the routing process, which makes the final result inaccurate [23].

3.2.2. Error in Mean Hop Distance. The entire traditional algorithm resolves the routing paths as straight paths connecting node pairs, as opposed to polygonal lines in most cases when the node positioning strategy is generally considered random. Therefore, the current distance estimate model representing the airline distances rather than the actual routing paths would certainly produce large errors, especially when its propagation routes cannot be reduced to straight lines.

Maximum limit method for the error of point coordinates: From the formula of calculating the coordinates of points by the maximum likelihood method [24], the coordinates (x, y) of the unseen nodes are the intersections of the circles with the radius of $d_1, d_2, d_3, \ldots, d_m$ and $(x_1, y_1), \ldots, (x_m, y_m)$ as the center of the circle.

However, in practical applications, the circles would not intersect at one point in most cases, but intersect in a small area. Therefore, there certain error is obtained in the node coordinates that cannot be eliminated. Moreover, when the n^{th} equation is sequentially subtracted by the $(n-1)^{th}$ equation, a large iteration error is generated each time [25].

4. The Proposed Improved DVH Algorithm

According to the actual distribution of sensor nodes, the principle of shortening the gap between actual data and ideal conditions is adopted to minimize the calculation error as much as possible. When resolving the transition distance and number of transitions, consider the distribution of the sensor nodes in actual situations and then improve it according to the different distribution conditions by putting weight to the discrete raw data with the help of correction factor or error factor.

4.1. Improvement of the Mean Hop Distance. Calculating the average transfer distance by multiplying the transition count will directly affect the accuracy of the algorithm. In this paper, two variables are introduced as error factors to control the weight and average transfer distance of the beacon node.

First, this error factor calculates the transfer distance error of beacon nodes. The algorithm for most DV-Hop extensions adopts the principle of proximity which only accepts the first message sent by beacon nodes. The individualized data selection cannot match the random distribution situation of the actual nodes. It reduces the accuracy rate of the algorithm.

Then, this algorithm takes the measuring average error caused by the average node and the average numbers of transfer count of any single beacon node by using the iterative error generated by the traditional algorithm when calculating the distance between the nodes.

Equation (6) describes how to calculate measuring mean error of transfer distance of beacon nodes E.

$$E = \frac{1}{2}(e_1 + e_2). \quad (6)$$

In equation (6), e_1 and e_2 represent error rates.

$$e_1 = \left(d_i - D_{\text{avg}}\right)^2, \quad (7)$$

e_1 represents the error, which is generated by comparing the average transfer distance of the beacon node with the actual hop distance as shown in

$$D_{\text{avg}} = \frac{1}{n}\sum_{i=1}^{n} d_i, \quad (8)$$

D_{avg} represents the mean transfer distance of all beacon nodes. n represents the number of beacon nodes. d_i represents the mean transfer distance of beacon nodes.

The error rate of e_2 is represented by

$$e_2 = \frac{1}{n}\sum_{i \neq j}\left(\frac{d_{ij} - D_{ij}}{hop_{ij}}\right)^2, \quad (9)$$

e_2 represents mean transfer distance error of every hop. d_{ij} denotes the actual distance of the beacon nodes. hop_{ij} denotes the minimum transition count.

The distance of the beacon node is represented by

$$D_{ij} = d_i \times hop_{ij}, \tag{10}$$

where D_{ij} represents the estimated distance between beacon nodes.

A weighted average method is proposed to reduce the error by making the weighted value inversely proportional to the average error. The weights are calculated as shown in

$$w_i = \frac{(1/|E|) + (1/hop_i)}{\sum_{j=1}^{M}\left((1/|E|) + \left(1/hop_j\right)\right)}. \tag{11}$$

In equation (11), hop is the transition count between unseen nodes and the beacon node. M is the number of nodes.

Finally, the modified weighting method is used to calculate by using

$$d = \sum_{i=1}^{M} w_i d_i, \tag{12}$$

where d_i is the original transfer distance and d represents the improved transition distance.

4.2. Improvement of the Hop Count. The hop count is defined as a value that describes the amount of normalized distance unit. The conventional DV-Hop algorithm records the hop count as 1 hop abstractly regardless of actual hop distance. Obviously, this hypothesis is improper. In response to this problem, error factor and correction factor are introduced in this paper.

First, get the correction factor by broadcast

$$\xi_{\mathrm{ij}} = 1 - L_{ij}^2, \tag{13}$$

where L_{ij} represents the error factor. Larger L_{ij} represents greater error as given in

$$L_{ij} = \frac{h_{ij} - H_{ij}}{h_{ij}}, \tag{14}$$

where h_{ij} represents the estimated transition count. H_{ij} represents the ideal hop count. The ideal hop count is calculated as actual distance d_{ij} as given in equation (15) which divide the communication radius R.

$$H_{ij} = \frac{d_i}{R}. \tag{15}$$

Then, calculate the improved hop count using

$$hop_{ij} = \left(1 - \frac{\left(h_{ij} - H_{ij}\right)^2}{h_{ij}^2}\right). \tag{16}$$

Finally, through the calculation of unseen nodes, an accurate location is found in

$$D_s = d \times hop_{ij}. \tag{17}$$

4.3. Improvement of the Particle Swarm Optimization. Particles in particle swarm algorithm not only act independently and move randomly, but also share information and cooperate among particles. The algorithm uses iteration search to find optimal solution. Then, we calculate the optimal position of the target node in the entire particle population.

The speed and position are updated as follows:

$$v_{id}^{k+1} = w * v_{id}^{k} + c_1 * r_1\left(p_{id}^{k} - x_{id}^{k}\right) + c_2 * r_2\left(p_{gd}^{k} - x_{id}^{k}\right), \tag{18}$$

$$x_{id}^{k+1} = \mathrm{x}_{id}^{k} + v_{id}^{k+1}. \tag{19}$$

In the common particle swarm algorithm, the particles will fall into a relatively stable stage which causes premature convergence of the particles. During this period, the convergence of particles will slow down so that the particles are difficult to achieve global optimal, thereby affecting the local minimum and increasing the difficulty of particle escape.

In this paper, the improvement has been made. Firstly, the particle velocity update equation is changed. This escape element is introduced to velocity update equation to disturb the particle learning strategy, thus escaping the local optimum. Then, the weight of the particle swarm algorithm is changed into a classification function classified according to the number of iterations, and different weights are calculated for different iteration times. Finally, the variation factor s is added to enhance the population diversity and reduce the probability of premature.

4.4. Improvement of the Particle Velocity Update Formula. To prevent particles from falling in local optimum in stable stage, this paper puts a premature flag to determine current position of particles by examining whether it is in the standard threshold. If the algorithm is in a normal state, it is optimized by the standard particle swarm algorithm. When the flag reaches the set threshold, it is judged that the particle enters the premature convergence at this time. At this time, a central learning strategy is applied to the particles, and an escape factor is defined to avoid premature aging of the particles. Let the particles escape the local optimum and continue to find the global optimal as in

$$\begin{aligned} v_{id}^{k+1} &= w * v_{id}^{k} + c_1 * r_1\left(p_{id}^{k} - x_{id}^{k}\right) + c_2 \\ &\quad * r_2\left(p_{gd}^{k} - x_{id}^{k}\right) + c_3 * r_3\left(c_{\mathrm{end}}^{k} - x_{id}^{k}\right). \end{aligned} \tag{20}$$

In equation (20), k is number of iterations, w is inertia weight, c_1 and c_2 are acceleration constants, p_{id} is the optimal position at which the particles are present, and p_{gd} is the optimal position of the global particles. r_1, r_2, r_3 are constants in the interval (0, 1), and x_{id} is the position vector of the particle.

Define $c_{\text{end}} = r_1 * p\text{best} + r_2 * pgest/2$ as the escape factor, the *pbest* is the optimal position in the transition process i, and *pgest* is the global counterpart.

4.5. Improvement of Weights. The inertia weight is usually used to control search ability of particle as shown in equation (21). Based on this theory, this paper proposes to set the weight as a piecewise function.

$$w = \begin{cases} w_{\max} - 0.1 \cdot (w_{\max} - w_{\min}) \cdot e\dfrac{k}{100}, \\ \\ w_{\max} - (w_{\max} - w_{\min}) \cdot \log\left(1.15 \cdot \dfrac{k}{100}\right) + 0.1A, \end{cases} \tag{21}$$

where $w_{\min}$ and $w_{\max}$ represent the minimum and maximum inertia weights. $A = 0.5 \cdot rand\ (\,)\ 0$.

By gradually decreasing the inertial weight of the entire particle population with a random algorithm, the accuracy of the particle swarm algorithm is improved.

4.6. Addition of a Variation Factor. Variation factor is introduced to clearly show the fitness of particle between variation factors s and s_i. It is simpler to choose particles with relatively low fitness to continue iteration. This not only enlarges search space and increases the population diversity, but also clearly shows the distribution situation of new nodes, as represented in

$$s = s_i - a(s_{i1} - s_{i2}) + (1 - a)(s_i - pg), \tag{22}$$

where s_i represents the position of the current particle, s_{i1} and s_{i2} are any two particle positions except the whole population, and $s_{i1} \neq s_{i2}$.

4.7. Selection of Fitness Function. The distance error is expressed by

$$\begin{cases} \sqrt{\mathrm{k}_{11}} = d_1 + \varepsilon_1, \\ \sqrt{\mathrm{k}_{22}} = d_2 + \varepsilon_2, \\ \vdots \\ \sqrt{\mathrm{k}_{nn}} = d_n + \varepsilon_n. \end{cases} \tag{23}$$

When ε equals to its minimum, position of the optimal, the reciprocal of hop counts is introduced as weight to control error caused by estimated distance of numerical high hop in variation of fitness function. Fitness function is all shown in

$$fitness(x, y) = \sum_{i=1}^{n} \left(\frac{1}{h_i} \left| \sqrt{\mathrm{k}_{ij}} - d_i \right| \right). \tag{24}$$

Minimize the value of $fitness(x, y)$ by multiple iterations to improve positioning accuracy.

4.8. Steps of Proposed Improved DVH Algorithm

(1) Calculate hop count h.

(2) Calculate the error of hop count by

$$L_{ij} = \left(h_{ij} - H_{ij}\right)/h_{ij}. \tag{25}$$

(3) Calculate the correction factor by the reciprocal of error. Use correction factor to correct the hop count to get new hop count with less error as shown in

$$hop_{ij} = \left(1 - \frac{\left(h_{ij} - H_{ij}\right)^2}{h_{ij}^{\ 2}}\right) h_{ij}. \tag{26}$$

(5) Calculate mean transfer distance.

(6) Calculate weight according to error of hop count by using

$$hop_{ij} = \left(1 - \frac{\left(h_{ij} - H_{ij}\right)^2}{h_{ij}^{\ 2}}\right) h_{ij}. \tag{27}$$

(7) Use weight to correct transfer distance to get new transfer distance with less error by using

$$d = \sum_{i=1}^{M} w_i d_i. \tag{28}$$

(8) Distance between beacon node and unseen node is shown in

$$D = hd. \tag{29}$$

(9) Initialize k, N, c_1, c_2, $w_{\max}$, and $w_{\min}$.

(10) Update speed formula and calculate the optimum position of global particle.

(11) Get the final coordinate of node (x, y).

The optimized and improved location algorithm is as follows:

(1) The minimum hop count is calculated by the shortest path method.

(2) For each beacon node, first repeat steps (13)–(16) to correct the number of bars, and the result of the loop execution is the average hop distance.

(3) Use equations (6)–(9) to correct the mean transfer distance and combine the hop count to estimate the distance from each beacon node.

(4) Set relevant parameters.

(5) Initialize a certain number of particles in a given area.

(6) The variation factor s is generated by (22) and compared with s_i to preserve the less adaptable.

(7) Let $t = t + 1$, and update the velocity and position.

(8) The fitness value of each particle at its current position is calculated using the fitness function $\text{fitness}(x, y)$.

(9) First, compare the individual optimal values of individuals into *pbest*.

(10) Node positioning is based on the optimized particle swarm algorithm.

(11) Determine whether the iteration stop condition is reached, and if not return to step 8. Otherwise, the optimal coordinate solution is output.

(12) The optimal coordinate solution of the current output is used as the positioning coordinate of the unseen node.

5. Experimental Results and Analysis

5.1. Simulation and Result Analysis. The simulation was carried out on the MATLAB 2019b platform. The datasets used in this paper are the longitude and latitude of weather station and mobile terminals of the main campus of BUPT. The data selection range of weather station information in partial areas is from 99.228211E, 41.186073N to 117.404741E, 25.930456N, while the selection range of mobile terminal information in the main campus of BUPT is from 116.36188E, 39.969686N to 116.36806E, 39.969686N as shown in Figures 1 and 2. After normalization, the latitude and longitude are, respectively, mapped to the interval of (0, 1) in proportion to meet the input data requirements of the algorithm.

The advantages of this algorithm from the above results can be observed. The two pictures simulate sensor clusters with datasets of two real environments that shows sensor nodes are distributed in different places at random states. Red stars and black dots are used to represent beacon nodes and unseen nodes, respectively. After random shuffling of the dataset, the first few ones are selected as the beacon nodes. In each experiment, the beacon nodes obtained were fixed at the same scale.

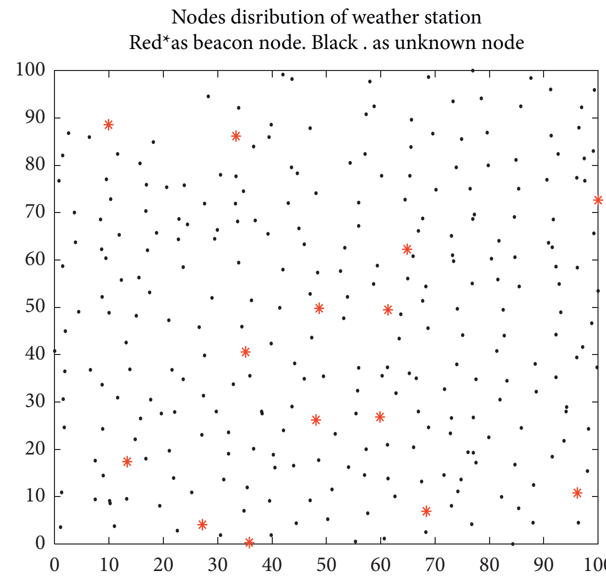

Figure 1: Weather stations in partial areas of China.

5.2. Analysis of Experimental Results. Hop count and hop distance are shown in Figures 3 and 4.

Hops count and hop distance have an impact on maximum likelihood estimation. The suggested technique improves a variety of input parameters, including the number and distance of leaps transferred, as well as the positional fitting model that processes them. In terms of hop count, the hop count resolved by the original algorithm using simple linear division method cannot fit into certain scenarios very well, where the nodes are gathered partially while they are discretized globally. For instance, although the original DV-Hop works well with evenly distributed nodes, as the weather station dataset shown in Figure 1, it cannot handle datasets like mobile terminals inside buildings as shown in Figure 2 and can be sensitive to these changes. So, the improved DV-Hop algorithm can cover more cases with better precision by weighting the original maximum hop count, which can better reflect the actual routing path.

The actual node distribution is complicated, and the hops between nodes are completely different. To solve this problem further, the article solves the problem of dropping mean transfer distance information through broadcasting and introduces two error factors averaged algebraically. From Figures 3 and 4, when solving the hop count and hop distance, the average accuracy is higher; thus, it has higher robustness under different concentrations of node dataset. By comparison, it is found that the derivative of the traditional calculation is unstable, so serious deviation will occur in the operation. In other words, the distribution of nodes has high randomness in the process of selecting beacon nodes. This is because the two datasets have different properties on the distribution of nodes. It shows that the traditional original algorithm cannot effectively use the meta-information captured in the broadcast communication process. The improved DV-Hop algorithm aims to solve the error loss and utilizes more information of the broadcast mechanism, so that the accuracy of the estimated distance is steadily improved, and the impact is smoother as the nodes' distribution feature changes. Use particle swarm to improve the unseen node coordinate estimation. Figure 5 shows average error against transmission radius with 30 particles and 50 iterations, and Figure 6 shows average error with 30 particles and 100 iterations. Figure 7 shows average error with 100 particles and 100 iterations.

Figure 8 shows average error against beacon nodes with 30 particles and 50 iterations, and Figure 9 shows average error against beacon nodes with 30 particles and 100 iterations. Figure 10 shows average error against beacon nodes with 100 particles and 100 iterations.

The enhanced technique can estimate the unseen node with accuracy. The original technique uses the estimated distance as the radius and uses the coordinates of m beacon nodes as the center of the circle. The m circles are difficult to intersect at one point, leading to a certain error in the obtained node coordinates. After iteration, the error becomes larger, so the position accuracy is greatly affected. Unlike previous methods that employ the maximum likelihood estimate approach to determine node coordinates, the modified DV-Hop algorithm chooses to improve on the

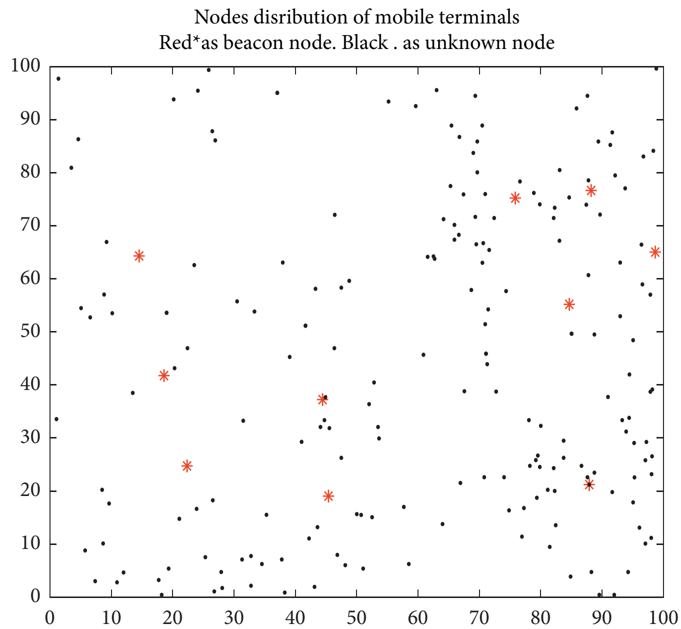

Figure 2: Mobile terminals of the main campus of BUPT.

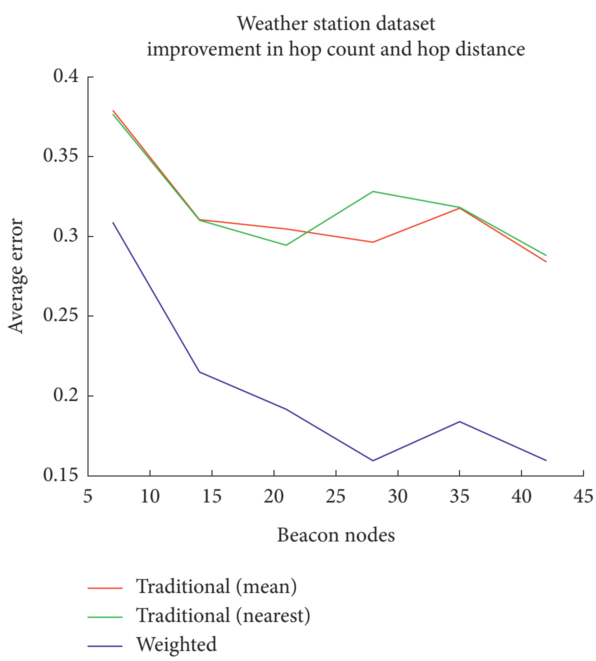

Figure 3: Hop count and hop distance on workstations.

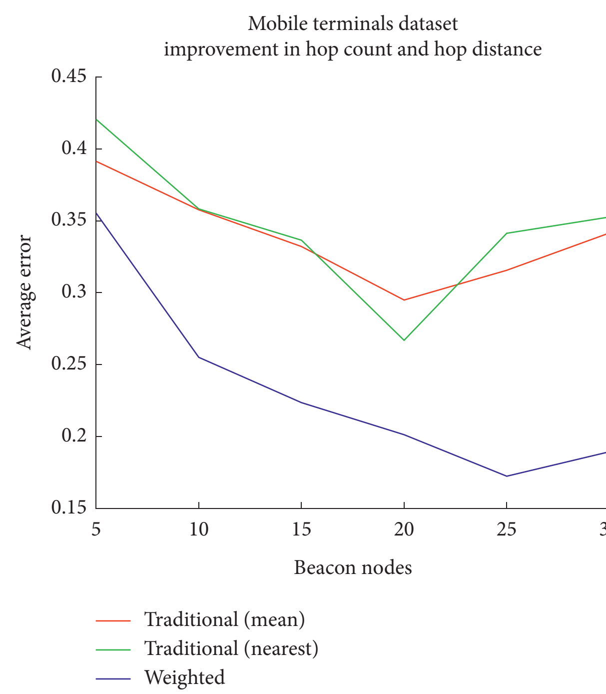

Figure 4: Hop count and hop distance on mobile terminals.

classic particle swarm algorithm. It incorporates escape factors to increase and manage particle update speed, preventing particles from escaping from the local optimum owing to premature status change, as well as finding the global optimum rapidly. Then, a piecewise function is introduced to calculate inertia weight in the iterative process, and weights are calculated according to the number of iterations.

The comparison between Figures 3 and 9 shows that this algorithm can effectively calculate the location of unseen nodes. By comparing Figures 5, 6, 8, and 9, it is concluded that the proposed algorithm optimizes the hop counts and hop distances. For example, within the communication range of 10, the average error of the experiment with 30 particles is 0.38, and the average error of the experiment with 100 particles is 0.24, so the improvement is more than 36%. The rest of Figures 6 to 10 shows the difference in average error when using different iteration times. After reducing the number of iterations from 100 to 50, the results of the algorithm did not change significantly; that is, analyzing its advantages, the number of iterations increases without causing an overreaction.

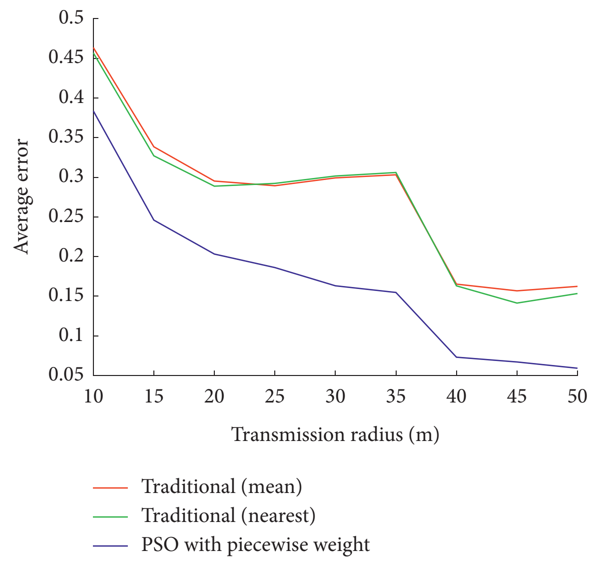

Figure 5: 30 particles and 50 iterations.

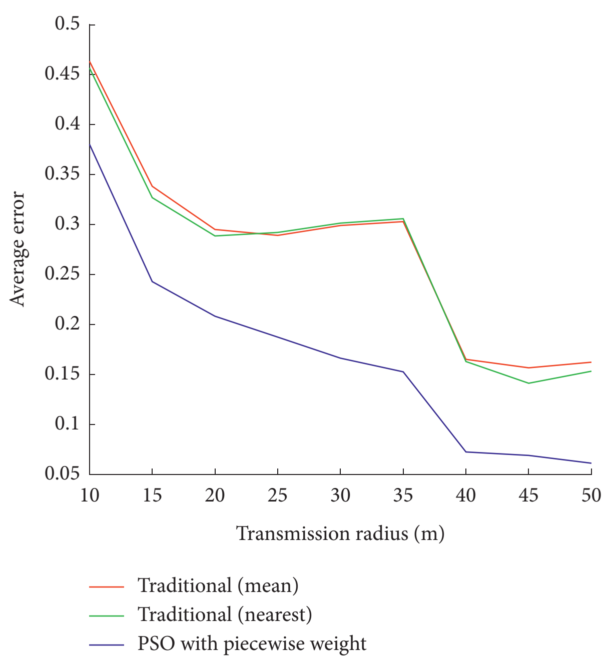

Figure 6: Average error for 30 particles and 100 iterations.

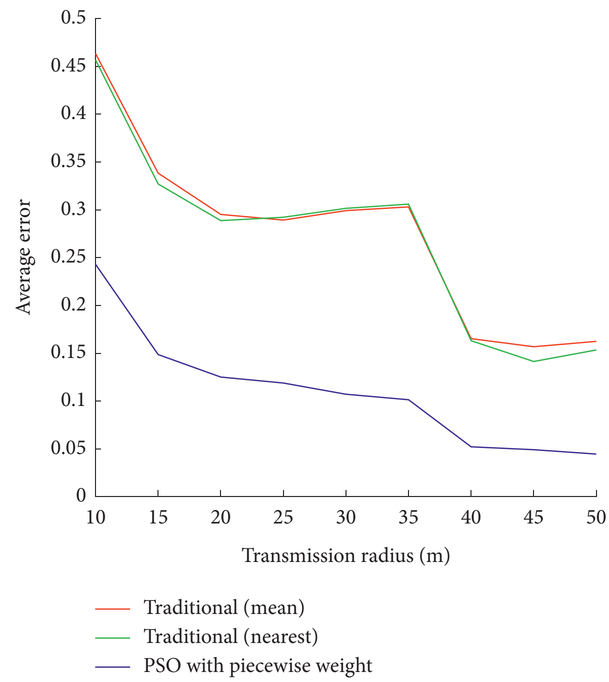

Figure 7: Average error for 100 particles and 100 iterations.

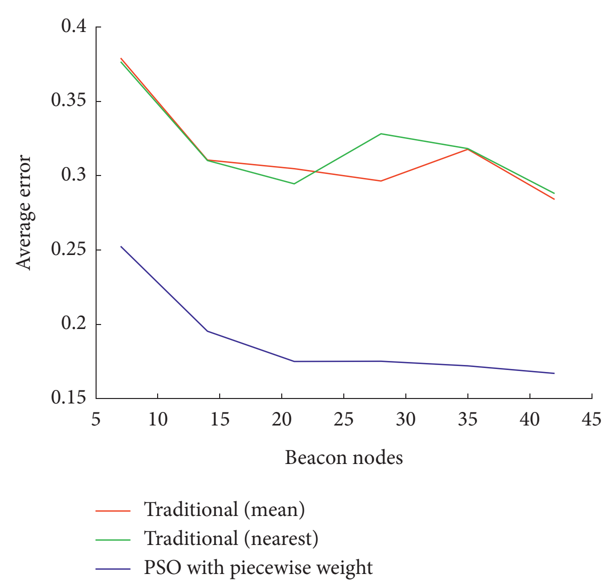

Figure 8: Average error for 30 particles and 50 iterations.

Figures 11 and 12 show the comparison of the two algorithms. The average error of this algorithm decreases more obviously with the change of communication range. Considering that the routing protocol used in this experiment is pre-static routing, the shortest path method is adopted. The communication range has little effect on the final routing path, so the main reason for the rapid improvement of efficiency in this part is the removal of invalid communication nodes, and the coordinate estimation accuracy of the node of this part is relatively close to a two-dimensional uniform distribution which can be found in experiments, especially if the mobile terminal dataset is used. After the communication range keeps increasing, the number of isolated nodes can be ignored. It can be observed that the two types of algorithms are in a stable state, but due to the improved DV-Hop algorithm using its reasonable correction of number of conversion and distance, the error rate of the results in any dataset all shows a monotonously decreasing nature. The traditional algorithms have a situation where the error rate increases as the communication radius increases. Although the performance of the communication range around 40 m that appeared on the weather station dataset has improved due to the local distribution

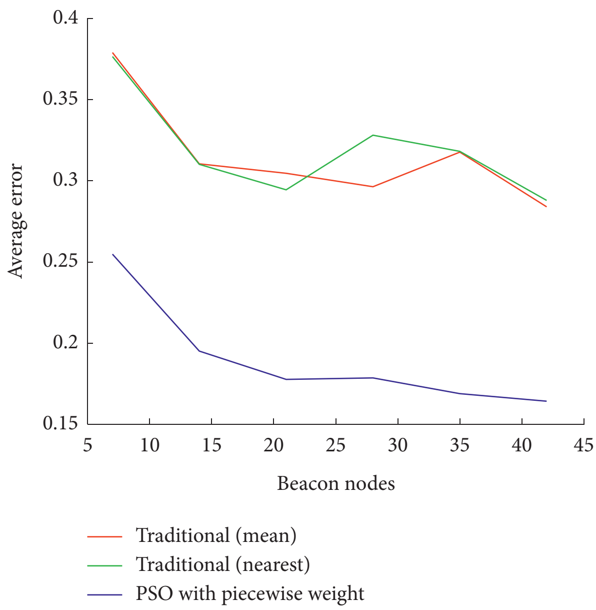

Figure 9: Average error for 30 particles and 100 iterations.

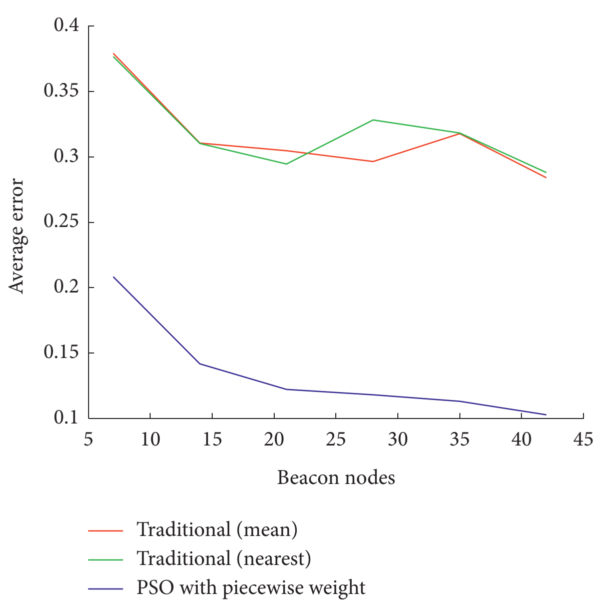

Figure 10: Average error for 100 particles and 100 iterations.

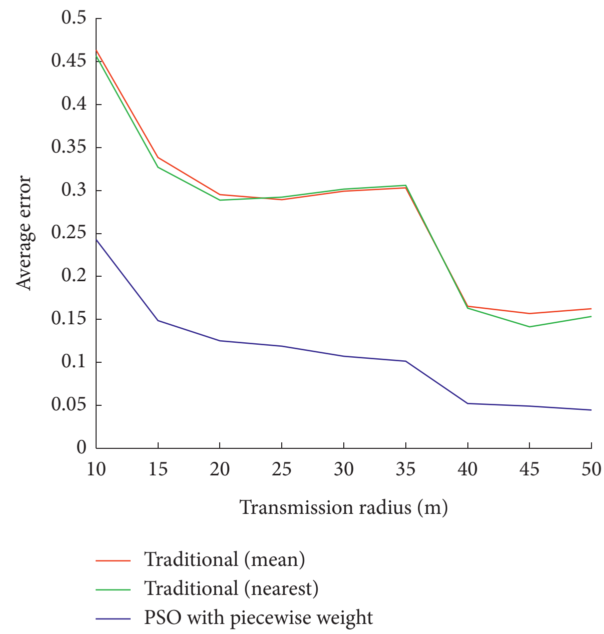

Figure 11: Influence of beacon node density on positioning error.

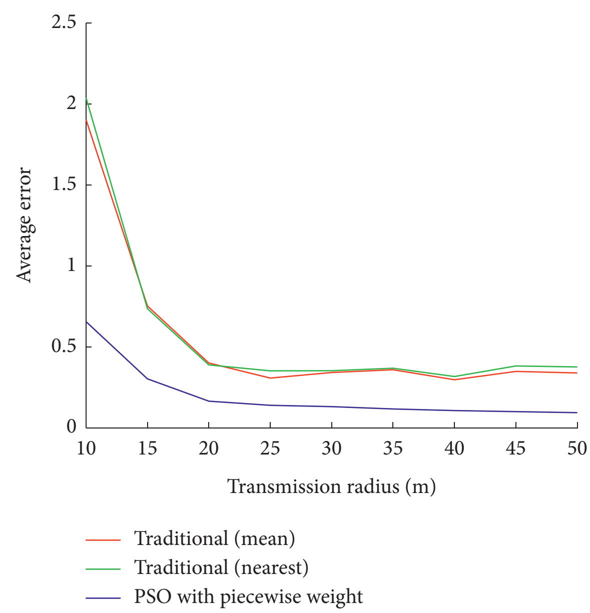

Figure 12: Coordinate estimation influenced by communication range.

characteristics of nodes, the error rate of the traditional algorithm remains stable at 0.5, which is close to unavailable on the mobile terminal dataset. In this situation, the improved DV-Hop algorithm's predicted coordinate error rate is 1/2 to 1/5 of the traditional algorithm's error rate. Influence of transmission radius on positioning error is shown in Figures 13 and 14.

Figures 13 and 14 show the variation trend of the average error calculated according to the different beacon nodes in different datasets. Moreover, for the traditional DV-Hop that uses maximum likelihood estimation, more beacon nodes mean longer estimation depth, which has a certain effect on the improvement of accuracy. Under large quantity of beacon nodes, traditional DV-Hop is restricted by the transfer distance acquisition under the proximity principle. As new beacon nodes are added to the global transfer distance value, the global transfer distance value changes greatly, resulting in the accuracy rate, and fluctuates up and down. The derived traditional algorithm basically effectively alleviates the instability problem by averaging the broadcasts of all beacon nodes, but it does not contribute to the overall accuracy rate, and there is still a serious error rate rebound. The improved DV-Hop algorithm can achieve better results in both datasets and has reduced error rate by more than

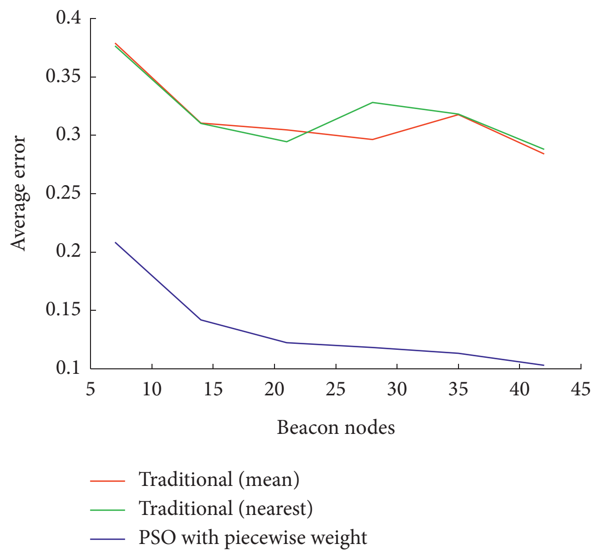

FIGURE 13: Coordinate estimation affected by beacon nodes.

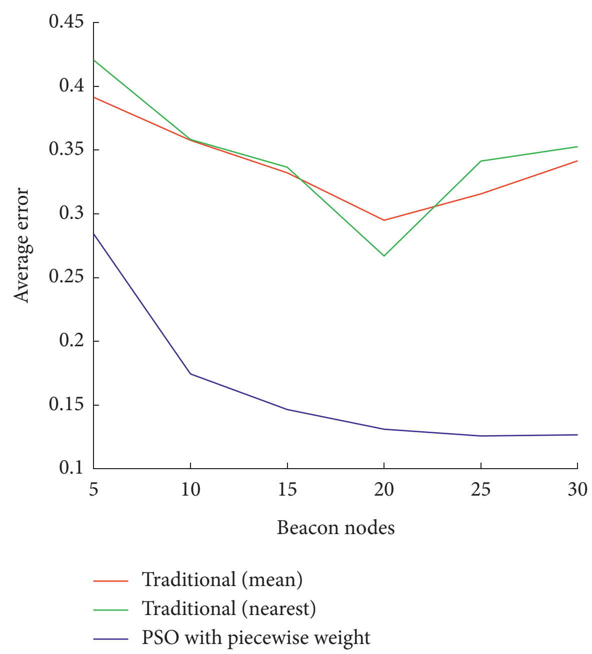

FIGURE 14: Average error influenced by beacon nodes.

30%. Until the beacon node accounts for 10% of all nodes, it has not entered the platform period. And as the proportion of beacon nodes increases, it shows a strict monotonic decrease, showing good robustness.

5.3. Discussion. In this paper, the improvement has been made in traditional DV-Hop algorithm. Firstly, the particle velocity update equation is changed. The escape element is introduced in velocity update equation to disturb the particle learning strategy, thus escaping the local optimum. Then, the weight of the particle swarm algorithm is changed into a classification function and classified according to the number of iterations. Different weights are calculated for different iteration times. Finally, the variation factor is added to enhance the population diversity and reduce the probability of premature convergence. To prevent particles from falling in local optimum in stable stage, this paper puts a premature flag to determine current position of particles by examining whether it is in the standard threshold. If the algorithm is in a normal state, it is optimized by the standard particle swarm algorithm. When the flag reaches the set threshold, it is judged that the particle enters the premature convergence at this time. The original DV-Hop works well with evenly distributed nodes, as the weather station dataset shown in Figure 1; it cannot handle datasets like mobile terminals inside buildings as shown in Figure 2 and can be sensitive to these changes. So, the improved DV-Hop algorithm can cover more cases with better precision by weighting the original maximum hop count, which can better reflect the actual routing path.

6. Conclusion

The study aims to propose an improved DV-Hop algorithm for IoT-enabled Industry 4.0 applications which make use of wireless communications, and hop count plays an important role. The improved DV-Hop improves the transfer distance method by using the advantages of particle swarm for the assessment of the node positions. Error rate in the distance between known and unseen nodes is optimized with the proposed technique that calculate error factors with corrections in a reversed fashion to revise hop counts. A new escape factor is devised to take control of updating particles' velocity in the system, and the inertia weight is defined by a piecewise function to enhance the search space. This mechanism increases the diversity of the particle populations and mitigates the tendency of particles' estimations on node positions to be trapped into local optima under stationary state. The improved DV-Hop algorithm described in the paper has a fast global convergence speed due to the presence of random inertia weight logarithmic method. The overall performance of improved DV-Hop is evaluated as shown in result section and is also compared with the traditional DV-Hop algorithm under simulated environment with the data collected from real-world scenarios. The DV-Hop algorithm plays an important role in IoT-enabled environment especially in Industry 4.0. In the future, we will propose a case study by using the improved DV-Hop algorithm.

Data Availability

The data are available for users on request.

Conflicts of Interest

The author declares no conflicts of interest with respect to this article.

Acknowledgments

We are thankful to all who have supported us to accomplish this research work.

References

[1] D. Xue, “Research of localization algorithm for wireless sensor network based on DV-Hop,” *J Wireless Com Network*, vol. 218, 2019.

[2] S. Messous, H. Liouane, O. Hamam, and H. Hamam, “Improved recursive DV-hop localization algorithm with RSSI measurement for wireless sensor networks,” *Sensors*, vol. 21, no. 12, p. 4152, 2021.

[3] S. Messous and H. Liouane, “Online sequential DV-Hop localization algorithm for wireless sensor networks,” *Mobile Information Systems*, 2020.

[4] V. Kanwar and A. Kumar, “DV-Hop-based range-free localization algorithm for wireless sensor network using runner-root optimization,” *The Journal of Supercomputing*, vol. 77, no. 3, pp. 3044–3061, 2021.

[5] M. Peyvandi and A. A. Pouyan, “An improved DV-Hop localization algorithm in wireless sensor networks,” in *Proceedings of the 2015 Sig. Proc. & Int. Sys. Conf. (SPIS)*, pp. 153–158, IEEE, Wuhan, China, June 2018.

[6] Y. Hu, S. Liu, and Z. Zhang, “An advanced DV-hop algorithm,” in *Proceedings of the 2017 1st Int. Conf. on Elec. Inst. & Info. Sys. (EIIS)*, pp. 1–5, IEEE, Harbin, China, June 2017.

[7] A. R. Kulaib, R. M. Shubair, and M. A. Al-Qutayri, “Improved DV-hop localization using node repositioning and clustering,” in *Proceedings of the Int. Conf. on Comm. Signal Proc. and App*, pp. 1–6, IEEE, Sharjah, United Arab Emirates, February 2015.

[8] B. Mareschal, M. Kaur, V. Kharat, and S. S. Sakhare, “Convergence of smart technologies for digital transformation,” *Tehnički glasnik*, vol. 15, pp. 1–IV, 2021.

[9] Y. Chen, X. Li, Y. Ding, J. Xu, and Z. Liu, “An improved DV-hop localization algorithm for wireless sensor networks,” in *Proceedings of the 2018 13th IEEE Conf. on Ind. Elec. and App. (ICIEA)*, pp. 1831-1836, *IEEE, Wuhan, China*, June 2018.

[10] H. Xiao, H. Zhang, Z. Wang, and T. A. Gulliver, “An RSSI based DV-hop algorithm for wireless sensor networks,” in *Proceedings of the 2017 IEEE Pacific Rim Conf. on Comm., Comp. and Sig. Proc.(PACRIM)*, pp. 1–6, Shanghai, China, September 2007.

[11] Y. Wang and Z. Wang, “Improved DV-hop localization algorithm based on dynamic anchor node set for wireless sensor networks,” *IEEE Access*, vol. 7, pp. 124876–124890, 2019.

[12] L. Jing and L. Zhang, “Improved DV-Hop localization algorithm based on hop distance optimization,” *Computer Times*, vol. 36, no. 4, pp. 945–951, 2016.

[13] M. Mehrabi, H. Taghdiri, and P. Taghdiri, “An improved DV-Hop localization algorithm based on evolutionary algorithms,” *Telecommunication Systems*, vol. 64, no. 4, pp. 639–647, 2017.

[14] H. Dang and Z. Li, “An improved DV-Hop location algorithm for wireless sensor network,” *Inst. Tech. and Sensor*, vol. 1, pp. 159–163, 2017.

[15] S. Xia, J. Zou, and X. Zhu, “An improvement on DV-Hop localization algorithm in wireless sensor networks,” *Journal of Computer Applications*, vol. 35, no. 2, pp. 340–344, 2015.

[16] F. Zeng, W. Li, and X. Guo, “An improved DV-hop localization algorithm based on average hop and node distance optimization,” in *Proceedings of the 2018 2nd IEEE Adv. Info. Mgmt. Comm., Elec. and Auto. Control Conf. (IMCEC)*, pp. 1336–1339, IEEE, Xi’an, China, May 2018.

[17] X. Li and X. Guo, “DV-Hop localization algorithm based on best mean hop distances and improved particle swarm,” *Application Research of Computers*, vol. 1, no. 34, p. 12, 2017.

[18] A. Jadhav, M. Kaur, and F. Akter, “Evolution of software development effort and cost estimation techniques: five decades study using automated text mining approach,” *Mathematical Problems in Engineering*, vol. 2022, Article ID 5782587, 17 pages, 2022.

[19] W. Kaur and M. Kaur, “A novel QACS automatic extraction algorithm for extracting information in blockchain-based systems,” *IETE Journal of Research*, vol. 1, p. 13, 2022.

[20] Y. Sun, M. Cao, Y. Gao et al., “Uncertain data stream algorithm based on clustering RBF neural network,” *Microprocessors and Microsystems*, vol. 81, no. 5, p. 103731, 2021.

[21] Y. Sun, Z. Song, and J. Zhao, “Research and improvement of particle swarm optimization in financial risk model,” *Journal of Jilin University (Engineering and Technology Edition)*, vol. 38, no. 02, pp. 85–91, 2020.

[22] D. Xue, “Research of location algorithm of wireless sensor networks based on DV-HOP,” *EURASIP Journal on Wireless Communications and Networking*, vol. 1, p. 2019, 2019.

[23] M. Kaur, A. Jadhav, and F. Akter, “Resource selection from edge-cloud for IIoT and blockchain-based applications in industry 4.0/5.0", security and communication networks,” *Hindawi*, vol. 2022, Article ID 9314052, 2022.

[24] M. Kaur, S. Kadam, and N. Hannoon, “Multi-level parallel scheduling of dependent-tasks using graph-partitioning and hybrid approaches over edge-cloud,” *Soft Computing*, vol. 26, no. 11, pp. 5347–5362, 2022.

[25] S. Qu, L. Xiong, and Z. Xiong, “Cross-layer congestion control of wireless sensor networks based on fuzzy sliding mode control,” *Neural Computing & Applications*, vol. 32, no. 17, pp. 13505–13520, 2020.