

## A Novel Localization Algorithm Based on Steepest Descent Method for the Wireless Sensor Network

Yan Xu<sup>1</sup>, Yi Zhuang<sup>2</sup>, Jingjing Gu<sup>3</sup> and Jinhui Zhao<sup>4</sup>

<sup>1, 2, 3, 4</sup>College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

<sup>1</sup>xugn1@163.com, <sup>2</sup>zy16@nuaa.edu.cn,

<sup>3</sup>gujungjing@nuaa.edu.cn, <sup>4</sup>jinhui90@hotmail.com

### Abstract

*To improve the positioning accuracy, this paper analyses the error source of the DV-Hop algorithm, and proposes an improved DV-Hop algorithm (IDV-Hop). The improvements of the IDV-Hop algorithm are threefold. First of all, we correct the average hop distance calculated by the beacon nodes. Secondly, the localization uses weighted average hop distances of multiple beacon nodes. Finally, we use the steepest descent method to optimize the localization result. The simulation results show that the proposed algorithm improves significantly in both positioning accuracy and stability, comparing with DV-Hop and CDV-Hop. Therefore, it is a feasible localization scheme in wireless sensor network.*

**Keywords:** *Wireless Sensor Network; the average hop distance; the steepest descent method; positioning accuracy*

### 1. Introduction

Wireless Sensor Networks (WSNs) are composed of a large number of sensor nodes which can collect, compute, and communicate. The sensor nodes form a multi-hop and self-configured network by means of wireless communication [1]. In the applications of WSNs, location information of sensor nodes is critical to the monitoring activities. The detected data is meaningless without knowing the location [2].

So the node localization in WSNs has attracted great interest. GPS (Global Positioning System) is one of the most accurate positioning technologies, with great features such as strong anti-interference, real-time location, *etc.* But it is quite expensive and energy consuming to equip all the nodes in WSNs with GPS. As an alternative, we usually equip some nodes with GPS to locate other nodes. In recent years, researches on localization in WSNs have achieved fruitful results. The localization algorithm of WSNs can be divided into two categories, range-based localization and range-free localization, according to whether or not the actual distance or angle between nodes is measured in the positioning process [3]. The application of range-based localization algorithms is limited, because they have high requirements for hardware of the WSN nodes, and are greatly influenced by the surroundings. The range-free localization algorithms are based on the connectivity or hop information of the WSNs. Compared with range-based localization, the range-free localization has lower cost, smaller energy consumption, simpler hardware and stronger robustness to the environment. As a result, range-free localization is widely studied in academic and applied in industry in recent years. At present, the most studied algorithms in range-free localization category include Amorphous algorithm [4,5], Centroid algorithm [6] and DV-Hop algorithm [7], *etc.*

To improve the accuracy of localization algorithm in irregular network, an improved DV-Hop algorithm is proposed in this paper from three aspects. It considers both the local and whole topology of the network and uses steepest descent method to optimize the

localization result. Experiment results show that the proposed scheme outperforms the DV-Hop algorithm and CDC-Hop algorithm.

Following is a summary of the key contributions:

This paper improves the accuracy of DV-Hop algorithm from three aspects:

1) Correct the average distance per hop: replace the average distance per hop of each beacon node with the average distance per hop of all the beacon nodes in the network. The corrected average distance per hop can reflect both the status of the closest beacon nodes and the entire network.

2) After the unknown node receives the average distance per hop from all the beacon nodes, weight them and get the new average distance per hop.

3) Use the steepest descent method to optimize the localization result.

The rest of the paper is organized as follows. The second section introduces traditional DV-Hop algorithm and analyzes the error source of the DV-Hop algorithm. In Section 3, the proposed IDV-HOP algorithm is described in details. Section 4 is a report of the simulation results and Section 5 is the conclusion and a prospect for further work.

## 2. Related Work

DV-Hop algorithm was first proposed by Niculescu in Rutgers University [7]. DV-Hop algorithm uses the product of the network average hop-distance and the number of hops to replace the distance between the beacon node and the unknown node. Then, the unknown nodes calculate their localization using trilateration or maximum likelihood estimation. DV-Hop algorithm is one of the typical range-free localization algorithms, with the advantage of low cost and small energy consumption. It has become one of the most widely used algorithms.

However, DV-Hop algorithm only works well in regular network. The accuracy of the DV-Hop still needs to be improved, especially in irregular network. So the researchers have improved accuracy from different aspects of DV-Hop. Jeffrey HS Tay *et al.* proposed a novel Selective Iterative Multilateration (SIM) algorithm [8] to improve the accuracy of location estimation in hop count-based localization schemes. New anchor nodes are selected so that the density of beacon nodes is improved. MAO Ke-ji *et al.* proposed a node localization method based on SVM [9]. The basic idea is dividing the network area into several small aliquots of grids, each representing a certain class of machine learning algorithm. When the machine learning algorithm has learned the classes corresponding to the known beacon nodes, it would classify the unknown nodes' localization and then further determine the position coordinates of the unknown nodes. Their proposal has higher localization accuracy and better tolerance of ranging error. LIN Jin-zhao *et al.* put forward three approaches to improve the poor locating performance of DV-Hop algorithm [10]. Firstly, the average one-hop distance among beacon nodes is refined by means of least squares method (LSM). Secondly, the average one-hop distance used by each locating node for estimating its own location is corrected by weighting the  $N$  received average one-hop distances from the beacon nodes. Finally, the iterative numerical method with the initial values of estimated node locations is presented by setting proper threshold. The improved algorithm has obviously better locating performance in locating precision and precision stability. Zhang Zhaoyang *et al.* proposed two improved DV-Hop algorithms and integrated them reasonably into a self-adaptive positioning algorithm called SAP which includes two modes [11]. Rough-precision mode uses DV-Hop I algorithm and high-precision mode uses DV-Hop II algorithm. This proposal strikes a good balance between positioning accuracy and energy consumption. Chunhua Huang *et al.* put forward an algorithm based on compensation coefficient, which is used to correct the estimated distances between unknown nodes and anchor nodes [12]. Hongyang Chen *et al.* use the average hop distance of the whole network to calculate the distances between unknown nodes and anchor nodes to reduce the errors caused by the

closest wrong anchor nodes. Hyperbolic positioning is utilized to solve the coordinates of the unknown nodes [13]. Mudong Li *et al.* use Artificial Bee Colony (ABC) algorithm in the DV-Hop algorithm [14] due to the strong robustness, fast convergence speed and good global optimization performance of ABC. In the last stage of the DV-Hop algorithm, objective function is set to work out the coordinates of the unknown nodes.

Current researches generally use different methods to correct the average distance per hop but they don't consider the topology of the network. So the current researches cannot apply in the irregular network. So to improve the accuracy of localization algorithm in irregular network, our scheme is proposed in this paper in three aspects.

### 3. Analysis of DV-Hop Algorithm

#### 3.1. Description of DV-Hop Algorithm

DV-Hop algorithm consists of three phases:

1) Phase one: information broadcast

Each beacon node in the network will convey the location information to all neighboring nodes. The information format is  $\{id_i, (x_i, y_i), hop_i\}$ , which contains its own identity  $id_i$ , location coordinates  $(x_i, y_i)$  and the hop count  $hop_i$ . The hop count is initialized at 0. Each node receives this data and records  $hop_i + 1$  to a table, and then continues to broadcast to their neighboring nodes. If a node receives the information with an id that already exists in the table, it compares the new  $hop_i$  with the local  $hop_i$  with the same id. If the new hop count is less than the local hop count, save the new hop count; otherwise the information will be discarded and no longer forwarded.

The beacon nodes can get the coordinates and hops of all other beacon nodes. So the beacon nodes calculate the distance of average per hop by Eq. 1:

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

Where,  $i$  and  $j$  are different beacon nodes;  $(x_i, y_i), (x_j, y_j)$  are the coordinates of beacon node  $i$  and  $j$ ;  $hop_{ij}$  is the hop count between beacon node  $i$  and  $j$  ( $i \neq j$ ).

2) Phase two: calculate distance

Every average distance per hop calculated by each beacon node is broadcast by flooding on the entire network. To make sure the unknown node receive the average distance per hop from the closest beacon node, each unknown node only accepts the first average distance per hop it receives and discards the others. Then each unknown node can calculate the distance to each beacon node by using the product of the network average hop-distance and the number of hops.

3) Phase three: calculate location

After calculating the distance between unknown nodes and beacon nodes, unknown nodes calculate their localization using trilateration or maximum likelihood estimation.

When the distance  $d$  between all beacon nodes and the unknown node  $p(x, y)$  is known, we have  $n$  equations as Eq. 2:

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = d_1^2 \\ \dots \\ (x_n - x)^2 + (y_n - y)^2 = d_n^2 \end{cases} \quad (2)$$

Eq. 2 can be expressed as Eq. 3:

$$\begin{cases} x_1^2 - x_n^2 + 2(x_1 - x_n)x + y_1^2 - y_n^2 - 2(y_1 - y_n)y = d_1^2 - d_n^2 \\ \dots \\ x_{n-1}^2 - x_n^2 + 2(x_{n-1} - x_n)x + y_{n-1}^2 - y_n^2 - 2(y_{n-1} - y_n)y = d_{n-1}^2 - d_n^2 \end{cases} \quad (3)$$

Eq. 3 can be expressed as linear Eq. 4:

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

Where,  $X = [x, y]^T$  is the coordinates of the unknown node  $p$ ,  $A$  and  $B$  can be expressed as Eq. 5 and Eq. 6.

$$A = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ \dots & \dots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix} \quad (5)$$

$$B = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_n^2 - d_1^2 \\ \dots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + d_n^2 - d_{n-1}^2 \end{bmatrix} \quad (6)$$

The coordinates of the unknown node  $p$  can be worked out using estimation methods of the standard minimum mean square with Eq. 7:

$$X = (A^T A)^{-1} A^T B \quad (7)$$

### 3.2. Error Source Analysis

According to the process of the DV-Hop algorithm, the error mainly comes from three aspects:

1) Error from the average distance per hop

The DV-Hop algorithm firstly gets the minimum hop between beacon node and unknown node, and uses the product of average distance per hop calculated by beacon nodes and the numbers of hops between beacon nodes and unknown node to estimate the distance of beacon nodes and unknown node. The average distance per hop is easily influenced in different networks, especially in irregular networks. The average distance per hop calculated by single beacon node cannot reflect the resulting average distance per hop of the entire network.

2) Error from the accumulated distance and actual distance

The number of hops between the unknown node and beacon node is larger when the beacon node is farther away from the unknown node. Error exists in the average distance per hop itself, so if the hop count is larger, the error of estimated distance will be much larger than the actual distance.

3) Error from trilateration or maximum likelihood estimation

After calculating the distance between each unknown node and beacon nodes, unknown nodes calculate their localization by trilateration or maximum likelihood estimation. Error also exists in the method of both trilateration and maximum likelihood estimation.

## 4. An Improved DV-Hop Algorithm based on Steep Decent Method (IDV-Hop)

### 4.1. Correct the Average Distance per hop of Beacon Nodes

To reduce the first type of error, the proposed scheme corrects the average distance per hop of the beacon nodes. The corrected average distance per hop in the proposed scheme considers not only the average distance per hop of single beacon node, but also the average distance per hop of all the beacon nodes in the entire network. Correction of average distance per hop is implemented as follows.

After each beacon node works out their average distance per hop, calculate the average value of the average distance per hop of the entire network with Eq.8:

$$c_{ave} = \frac{1}{n} \sum_{i=1}^n c_i \quad (8)$$

Where  $n$  is the number of beacon nodes,  $c_i(i=1,\dots,n)$  is the average distance per hop of each beacon node,  $c_{ave}$  is the average value of the average distance per hop of the entire network.

Get the new average distance per hop  $c'_i$  with Eq.9:

$$c'_i = \frac{1}{2}(c_i + c_{ave}) \quad (9)$$

According to the new average distance per hop  $c'_i$ , beacon node  $i$  can use the product of the new average distance per hop  $c'_i$  and the numbers of hops with beacon node  $j$  as the distance  $d'_{ij}$ , calculated by Eq.10:

$$d'_{ij} = c'_i * hop_{ij} \quad (10)$$

Where,  $i$  and  $j$  are both ids of beacon nodes,  $hop_{ij}$  is the number of hops between beacon node  $i$  and  $j$ .

There exists some error in the distance  $d'_{ij}$  estimated by Eq.10, comparing with the actual distance  $d_{ij}$ . The error of average distance per hop  $err_i$  can be calculated with Eq.11:

$$err_i = \frac{1}{n-1} \times \frac{\sum_{i \neq j} d_{ij} - d'_{ij}}{\sum_{i \neq j} hop_{ij}} \quad (11)$$

Where  $d'_{ij}$  is the estimated distance between beacon node  $i$  and beacon node  $j$ ;  $d_{ij}$  is the actual distance between beacon node  $i$  and beacon node  $j$ ,  $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ .

The error  $err_i$  was used to correct  $c'_i$  and get the corrected average hop distance  $hopsiz_e_i$  with Eq.12:

$$hopsiz_e_i = c'_i + \frac{1}{n} \sum_{i=1}^n err_i \quad (12)$$

## 4.2. Weight the Average Distance per Hop

In the second phase of the DV-Hop algorithm, each unknown node only accepts the average distance per hop of the beacon node that is the closest to it. However, as to the entire network, the average distance per hop of a single beacon node usually exist big error and cannot reflect the resulting average distance per hop of the entire network. So in the proposed scheme of this paper, we weight the average distance per hop of several beacon nodes to reduce this type of error.

Assume that an unknown node  $p$  receives a corrected average distance per hop  $hopsiz_e_i$  from  $n$  beacon nodes. The weight value  $w_i$  of beacon node  $i$  is calculated with Eq. 13:

$$w_i = \frac{hopsiz_e_i}{\sum_{k=1}^n hopsiz_e_k} \quad (13)$$

The unknown node  $p$  can calculate its average distance per hop  $c_p$  with Eq.14:

$$c_p = \sum_{i=1}^n w_i hopsiz_e_i \quad (14)$$

Then the unknown node  $p$  can use  $c_p$  and the numbers of hops with each beacon node to get the distance  $dis_{pi}$  with each beacon node, with Eq. 15.

$$dis_{pi} = c_p * hop_{pi} \quad (15)$$

Where,  $hop_{pi}$  is the hop count between beacon node  $i$  and the unknown node  $p$ .

## 4.3. Optimize the Localization Result

In the third phase of the DV-Hop algorithm, unknown nodes calculate their localization using trilateration or maximum likelihood estimation. Trilateration method is sensitive to distance measurement error. When distance measurement error exists, the coordinates

worked out by Trilateration method usually have a big deviation. So the steepest descent method is used to optimize the results.

Steepest descent algorithm [15] is a minimize optimization method, also known as the gradient method. Steepest descent algorithm is modified on the basis of Newton's method. Although Newton's method has relatively faster convergence speed, its calculation is complicated, which can be challenging for sensor nodes because of their limited computing resource. The steepest descent algorithm is easy in calculation and programming, so that it is suitable to apply to WSNs.

Maximum likelihood estimation algorithm can only obtain an approximation of the unknown node location, while the steepest descent algorithm can search for the optimal solution within a certain range on the basis of approximation. It is possible to combine these two methods together theoretically. The maximum likelihood estimation algorithm provides initial information for the steepest descent algorithm. It does not need any extra communication overhead. The process using the steepest descent algorithm to improve the positioning accuracy is as follows.

$(x, y)$  is the coordinates of unknown node,  $dis_{pi}$  is the distance between beacon node  $i$  and the unknown node  $p$ .  $(x_i, y_i)$  is the coordinates of beacon node  $i$ . The positioning error of the unknown node  $E_i$  can be expressed as Eq.16.

$$E_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} - dis_{pi} \quad (16)$$

To minimize the positioning error, we construct the object function on the unknown node as Eq.17:

$$F = \sum_{i=1}^n E_i^2 = \sum_{i=1}^n (\sqrt{(x_i - x)^2 + (y_i - y)^2} - dis_{pi})^2 \quad (17)$$

The minimized value of Eq. 17 is the location of the unknown node. The steepest descent method is used in the iterative calculation of Eq.17, in order to solve the targeted node location. The steps are as follows:

step1: Let  $x^{(0)}, y^{(0)}, dis_{p1}^{(0)}, dis_{p2}^{(0)}, \dots, dis_{pi}^{(0)}, \dots, dis_{pm}^{(0)}$  be the initial values of the objective function.  $x^{(0)}, y^{(0)}$  is the solution of Eq.7, while  $I_{ki}^{(0)}$  is the solution of Eq.15.

step2: Carry out the iterative calculation of Eq.17. In the  $k^{th}$  iteration, the solution is  $x^{(k)}, y^{(k)}, dis_{p1}^{(k)}, \dots, dis_{pi}^{(k)}, \dots, dis_{pm}^{(k)}$ , and the values of the objective function are  $F = F(x^{(k)}, y^{(k)}, dis_{p1}^{(k)}, dis_{p2}^{(k)}, \dots, dis_{pi}^{(k)}, \dots, dis_{pm}^{(k)})$ .

step3: If  $|F| < \varepsilon$ ,  $x^{(k)}, y^{(k)}$  is the solution we want. Otherwise, go to step4.

step4: Calculate the difference quotient with Eq.18.

$$\begin{cases} \frac{F^{(k)}}{x} = \frac{F(x^{(k)} + x, y^{(k)}, dis_{p1}^{(0)}, \dots, dis_{pm}^{(0)}) - F(x^{(k)}, y^{(k)}, dis_{p1}^{(0)}, \dots, dis_{pm}^{(0)})}{x} \\ \frac{F^{(k)}}{y} = \frac{F(x^{(k)}, y^{(k)} + y, dis_{p1}^{(0)}, \dots, dis_{pm}^{(0)}) - F(x^{(k)}, y^{(k)}, dis_{p1}^{(0)}, \dots, dis_{pm}^{(0)})}{y} \end{cases} \quad (18)$$

step5 : Calculate  $x^{(k+1)}$  and  $y^{(k+1)}$  with Eq.19.

$$\begin{cases} x^{(k+1)} = x^k - M^k \left( \frac{F}{x} \right)^k \\ y^{(k+1)} = y^k - M^k \left( \frac{F}{y} \right)^k \end{cases} \quad (19)$$

$M^k$  is calculated with Eq.20.

$$M^k = \frac{F(x^{(k)}, y^{(k)}, dis_{p1}^{(0)}, \dots, dis_{pm}^{(0)})}{\left[ \frac{F^{(k)}}{x} \right]^2 + \left[ \frac{F^{(k)}}{y} \right]^2} \quad (20)$$

Go to step 2 until the objective function meets  $|F| < \varepsilon$ .

#### 4.4. The Process of the Proposed Algorithm

Improvement is made from the following three aspects:

1) Correct the average distance per hop: replace the average distance per hop of each beacon node with the average distance per hop of all the beacon nodes in the network. The corrected average distance per hop can reflect both the status of the closest beacon nodes and the entire network.

2) After the unknown node receives the average distance per hop from all the beacon nodes, weight them and get the new average distance per hop.

3) Use the steepest descent method to optimize the localization result.

The procedure of the proposed IDV-Hop algorithm is shown in Figure1, and is described as follows:

Step1: Network Initialization.

Step2: Each node in the network receives and forwards the information of the beacon nodes.

Step3: Each node checks whether the packet is from the same beacon node. If the node hasn't received its packet yet, save the packet; otherwise, compare the hop values and save the smaller one.

Step4: If the broadcast is not over, go to step1; otherwise go to the next step.

Step5: If an unknown node has received information from more than three beacon nodes, go on the next step; otherwise, mark the unknown node as an isolated node.

Step6: The beacon nodes calculate their average distance per hop with Eq.1.

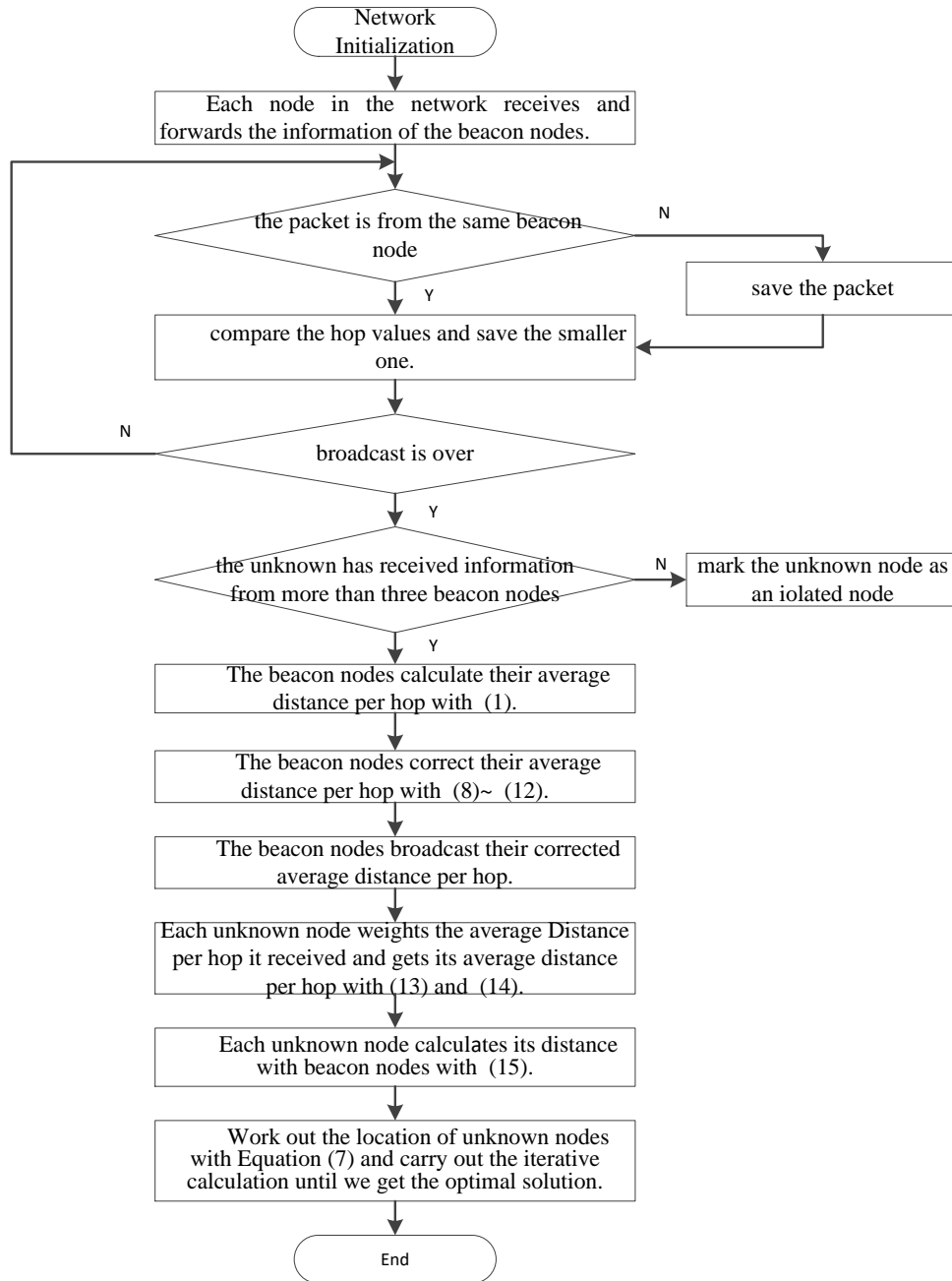
Step7: The beacon nodes correct their average distance per hop with Eq.8 ~ Eq.12.

Step8: The beacon nodes broadcast their corrected average distance per hop.

Step9: Each unknown node weights the average distance per hop it received and gets its average distance per hop with Eq.13 and Eq.14.

Step10: Each unknown node calculates its distance with the beacon nodes using Eq.15.

Step11: Work out the location of unknown nodes with Eq.7 and carry out the iterative calculation with Eq.18~ Eq. 20 until we get the optimal solution.



**Figure 1. The Process the IDV-Hop Algorithm**

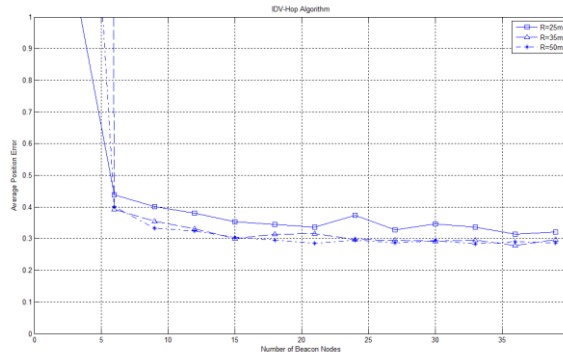
## 5. Simulation and Analysis

To verify the performance of the proposed algorithm, we use Matlab2009b to perform simulations. In the simulation, all the nodes are randomly deployed in a square region of 100m\*100m. There are totally 100 nodes in the network including both beacon nodes and unknown nodes. The simulation results of the proposed IDV-Hop algorithm are compared with DV-Hop algorithm and CDV-Hop (Correct average distance per hop DV-Hop) algorithm.

Under the different communication radius, the average position error of the proposed algorithm is shown in Figure2. The average position error decreases with the

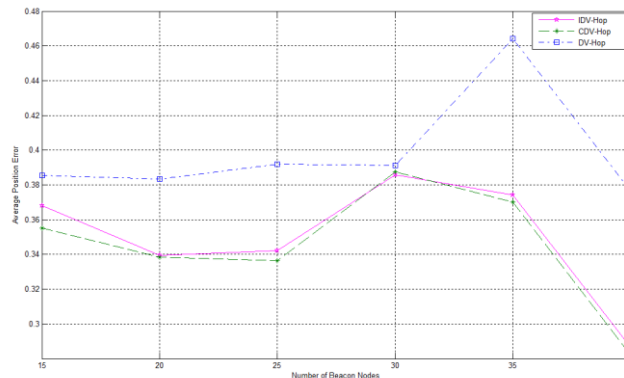


increase of the communication radius R. When R is set as 50m, the positioning error is the lowest.



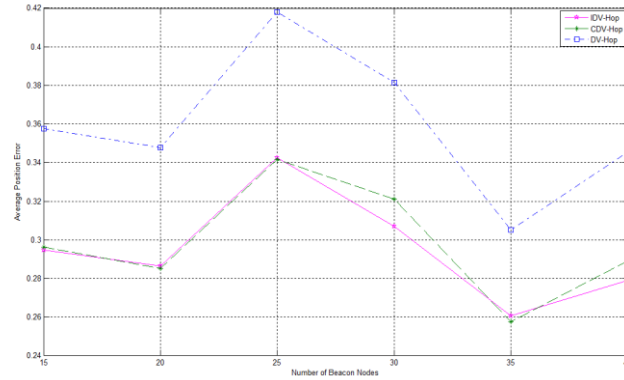
**Figure 2. The Average Position Error in Different Communication Radius**

Figure 3 shows the comparison of the average position error of three algorithms in different proportions of beacon nodes (when R=25m). It is obvious that under the same communication radius and the same proportion of beacon nodes, the proposed algorithm outperforms other two algorithms, judging from the metrics of the average position error. The proposed algorithm is able to reduce 2%~10% of the average position error.



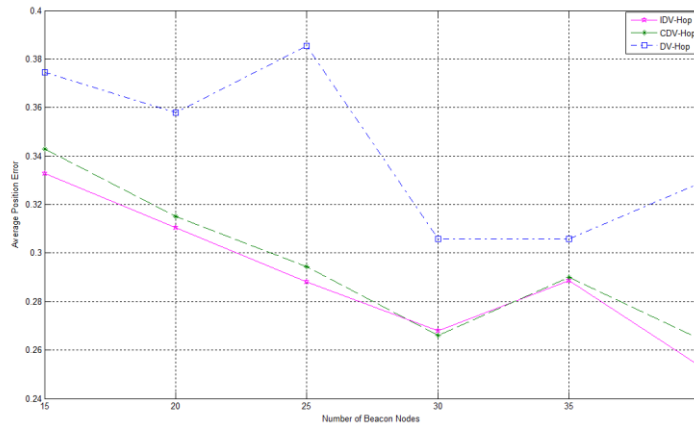
**Figure 3. The Average Position Error in Proportion of Beacon Nodes(R=25m)**

Figure 4 shows the comparison of the average position error of three algorithms in different proportions of beacon nodes (when R=30m). Results show that the proposed algorithm reduces 6% of the average position error, and outperforms other two algorithms again.



**Figure 4. The Average Position Error in Proportion of Beacon Nodes(R=30m)**

Figure 5 shows the comparison of the average position error of three algorithms in different proportions of beacon nodes (when R=35m). Similar results are obtained. The proposed algorithm reduces 5% of the average position error.



**Figure 5. The Average Position Error in Proportion of Beacon Nodes(R=35m)**

It is obvious that the proposed algorithm improves the position accuracy, comparing with the DV-Hop algorithm and CDV-Hop algorithm. Due to the random distribution of nodes in the simulation, the topology structure of the simulated network is different every time. So the average position error appears to have certain fluctuation. However, the average position error of the proposed algorithm, compared with other two algorithms, has better stability.

## 6. Conclusions

In order to improve the localization algorithm of DV-Hop algorithm, the paper summarizes and analyzes the error types of the DV-Hop algorithm, and proposes an improved IDV-Hop algorithm. The improvements of the IDV-Hop algorithm are threefold. Firstly, we correct the average distance per hop. Secondly, we weight the average distance per hop and get the new average distance per hop. Finally, we use the steepest descent method to optimize the localization result. The simulation results show that the proposed algorithm improves significantly in both positioning accuracy and stability, comparing with DV-Hop algorithm and CDV-Hop algorithm.

## Acknowledgments

This work is supported by the National Natural Science Foundation for Youth of China under Grant no. 61202351 and the National Postdoctoral Fund under Grant no. 2011M500124.

## References

- [1] Y.-H. Zhao, Z.-H. Qian and X.-H. Shang, "PSO localization algorithm for WSN nodes based on modifying average hop distances", *Journal on Communications*, vol. 34, no. 9, (2013), pp. 105-14.
- [2] Q. Qian, X. Shen and H. Chen, "An improved node localization algorithm based on DV-Hop for wireless sensor networks", *Computer Science and Information Systems*, vol. 8, no. 4, (2011), pp. 953-72.
- [3] D. Wang, H. Jia and F. Chen, "An improved DV-distance localization algorithm for wireless sensor networks", *Proceedings of the Advanced Computer Control (ICACC), 2nd International Conference*, F, IEEE, (2010).
- [4] R. Nagpal, "Organizing a global coordinate system from local information on an amorphous computer", (1999).
- [5] R. Nagpal, H. Shrobe and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network", *Proceedings of the Information Processing in Sensor Networks*, F, Springer, (2003).
- [6] N. Bulusu, J. Heidemann and D. Estrin, "GPS-less low-cost outdoor localization for very small devices", *Personal Communications, IEEE*, vol. 7, no. 5, (2000), pp. 28-34.
- [7] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks", *Telecommunication Systems*, vol. 22, no. 1-4, (2003), pp. 267-80.
- [8] J. H. Tay, V. R. Chandrasekhar and W. K. G. Seah, "Selective Iterative Multilateration for Hop Count-Based Localization in Wireless Sensor Networks", *Proceedings of the Mobile Data Management, MDM 7th International Conference*, F, IEEE, (2006).
- [9] K.-J. Mao, C.-L. Fan and F. Ye, "Node Localization Algorithm in Wireless Sensor Networks Based on SVM", *Journal of Computer Research and Development*, vol. 51, no. 11, (2014), pp. 2427-36.
- [10] J.-Z. Lin, X.-B. Chen and H.-B. Liu, "Iterative algorithm for locating nodes in WSN based on modifying average hopping distances", *Journal on Communications*, vol. 10, (2009), pp. 107-13.
- [11] Z. Zhao-Yang, G. Xu and L. Ya-Peng, "DV-hop based self-adaptive positioning in wireless sensor networks", *Proceedings of the Wireless Communications, Networking and Mobile Computing, WiCom 5th International Conference*, F, IEEE, (2009).
- [12] C.-H. Huang and J. Shen, "Improved DV-Hop Positioning Algorithm Based on Compensation Coefficient", *Computer Technology and Development*, vol. 21, no. 6, (2011), pp. 81-4.
- [13] H. Chen, K. Sezaki and P. Deng, "An improved DV-Hop localization algorithm for wireless sensor networks", *Proceedings of the Industrial Electronics and Applications, ICIEA 3rd IEEE Conference*, F, IEEE, (2008).
- [14] M.-D. Li, W. Xiong and L. Guo, "Improvement of DV-Hop localization Based on Artificial Bee Colony Algorithm", *Computer Science*, vol. 40, no. 1, (2013), pp. 33-6.
- [15] Q.-Q. Shi, H. Huo, T. Fang and D.-R. Li, "Using steepest descent method to improve node localization accuracy of maximum likelihood estimation", *Application Research of Computers*, vol. 25, no. 7, (2008), pp. 2037-2041.
- [16] X. Chen and B. Zhang, "Improved DV-Hop node localization algorithm in wireless sensor networks", *International Journal of Distributed Sensor Networks*, (2012).
- [17] A. Kumar, A. Khosla and J. S. Saini, "Stochastic algorithms for 3D node localization in anisotropic wireless sensor networks", *Proceedings of the Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, F, Springer, (2013).
- [18] M.-D. Li, W. Xiong and Q. Liang, "Wireless Sensor Networks Node Localization Algorithm Based on Improved ABC Algorithm", *Chinese Journal of Sensors & Actuators*, vol. 26, no. 2, (2013), pp. 241-245.
- [19] S. Kumar and D. K. Lobiyal, "An Advanced DV-Hop Localization Algorithm for Wireless Sensor Networks", *Wireless Personal Communications*, vol. 71, no. 2, (2013), pp. 1365-1385.
- [20] W. Hu and X. L. Zeng, "Three-dimensional DV-Hop Localization Algorithm Based on Correction of Average Hop Distance", *Wireless Communication Technology*, (2013).

