



Journal of
**Software
Engineering**

ISSN 1819-4311



Academic
Journals Inc.

www.academicjournals.com

Improvement of DV-Hop Localization Based on Evolutionary Programming Resample

Wanli Zhang, Xiaoying Yang and Qixiang Song

College of Information Engineering, Suzhou University, Suzhou, Anhui, China

Corresponding Author: Wanli Zhang, College of Information Engineering, Suzhou University, Suzhou, Anhui Province, China

ABSTRACT

In this study, the evolutionary programming algorithm with high stability, fast convergence and excellent performance on solving global optimization problem is introduced into DV-Hop localization algorithm and an improved DV-Hop localization algorithm based on evolutionary programming resample (EPRDV-Hop) is proposed to solve the error problem, which is brought by DV-Hop localization algorithm using trilateration or maximum likelihood estimation method to calculate the coordinates of the unknown node. After obtaining the estimated distances between the unknown nodes and anchor nodes, the initial position estimation is achieved based on the sampling. Then, the evolutionary programming based position resample is carried out. Finally, the final estimated position of the unknown node is obtained through an iterative manner. Simulation results show that EPRDV-Hop algorithm effectively improves the positioning accuracy of the node without additional hardware.

Key words: WSN, DV-Hop algorithm, evolutionary programming, resample

INTRODUCTION

Location information of node plays a very important position on various applications of Wireless Sensor Networks (WSN). The current location algorithms can be divided into two major categories according to ranging and without ranging (Sun *et al.*, 2005). DV-Hop algorithm proposed by Niculescu and Nath (2003) is a typical localization algorithm without ranging, with computing simplicity and low hardware requirements, making it widely used in node localization in WSN. However, the algorithm has a large positioning error. To solve this problem, many researchers have done a lot of research, such as in the study of Zhang *et al.* (2012), Wen *et al.* (2014), Xiao and Liu (2012), Feng *et al.* (2012), Wang and Shi (2012), Wang *et al.* (2011), Lu *et al.* (2011), Li and Zhou (2011) and Li *et al.* (2013). They have improved the algorithm and achieved certain results. These improved algorithms basically improved the error caused by the first and second stage of the positioning process. However, there are few works that study the error caused by the third phase of the positioning process.

In this study, combining the principles of evolutionary programming algorithm, a DV-Hop improvement program based on evolutionary programming resample is proposed to solve the error caused by the third phase of the positioning process. Simulation results show that under conditions that without additional hardware, positioning accuracy of EPRDV-Hop algorithm has greatly been improved.

DV-HOP ALGORITHM:

Positioning process: The estimated distance between an unknown node and the anchor node is obtained by hop multiplied by average distance per hop, therefore, to estimate the unknown node coordinates. The positioning process is divided into three steps:

Step 1: Each anchor node broadcasts packet $\{id_i, (x_i, y_i), hop_i\}$ which contains an identification number, its coordinates and number of hops to the network and the value of hop_i is initialized to 0. After receiving the packet, the neighbour nodes check whether the table has the anchor node information or not. If the packet information is recorded, set hop_i+1 and then forward the packet to its neighbor nodes. If there already exists and the hop is larger than the hop recorded in the table, then the packet is ignored. After the broadcast ends, all nodes will get the minimum hops value of each anchor node

Step 2: Using the coordinate position obtained in the first stage and hops, each of the anchor nodes calculate the average distance per hop by the Eq. 1:

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

where, $(x_i, y_i), (x_j, y_j)$ are coordinates of node i, j , respectively and hop_{ij} is the minimum number of hops between the two nodes. Then anchor nodes broadcast $hopsize$ to the network and unknown nodes only write down the first $hopsize$. Then the estimated distance d between each anchor node is obtained by using hop multiplied by $hopsize$

Step 3: The coordinate location of the unknown node is obtained by trilateration or maximum likelihood estimation method

Positioning error analysis: In traditional DV-Hop algorithm, after the unknown node obtains the estimated distances of three or more anchor nodes, the trilateration or maximum likelihood estimation method is used to determine the coordinate location of the unknown nodes. The coordinate position of the anchor nodes is set as $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ and the coordinate position of the unknown node is set as (x, y) , the estimated distances of unknown node and anchor nodes, are d_1, d_2, \dots, d_n , respectively, so Eq. 2 holds:

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

Linear expression of Eq. 2 is $AX = b$, where, $X = (x, y)^T$:

$$A = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix}, \quad b = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_n^2 - d_1^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + d_n^2 - d_{n-1}^2 \end{bmatrix}$$

The standard minimum mean square error is used to obtain coordinate of the unknown node, which is denoted as $X^T = (A^T A)^{-1} A^T b$.

In the first stage of traditional DV-Hop algorithm, when getting the minimum number of hops between nodes, distance is recorded as a jump as long as it lies in inner radius, no matter how near and far it is, which cannot reflect the size of the actual distance. There must exist errors when the average distance per hop is calculated by using Eq. 1. Results of d_1, d_2, \dots, d_n will inevitably lead to the accumulation of errors. Therefore, unknown node coordinate obtained by Eq. 2 has large error, which affects the positioning accuracy of the algorithm.

Set error of actual distance and estimated distance of anchor node and unknown node as ϵ , Eq. 2 can be changed into Eq. 3:

$$\begin{cases} \sqrt{(x-x_1)^2+(x-y_1)^2}-d_1=\epsilon_1 \\ \vdots \\ \sqrt{(x-x_n)^2+(x-y_n)^2}-d_n=\epsilon_n \end{cases} \quad (3)$$

When solving the unknown node's coordinates (x, y) , Eq. 4 obtains the minimum value as long as the minimum value of ϵ is achieved and the obtained coordinates of unknown node at this time by the influence of the error is minimized, therefore, the accuracy of the coordinate calculation is improved:

$$Fit(x, y) = \sum_{i=1}^n \sqrt{(x-x_i)^2+(x-y_i)^2}-d_i \quad (4)$$

METHODOLOGY

EPDV-Hop algorithm

EP algorithm: Evolutionary Programming (EP) is an evolutionary model of a finite state machine which is proposed by Fogel and Fogel (1996). It solves prediction problem by imitating process of biological evolution in nature. EP Algorithm is robust, intelligent, etc. and is a very broad application optimization search tool.

In evolutionary programming model, it uses potential problem of finite state machine to represent individuals. It first generates a finite state machine groups that are the parent groups. Each of these individuals produces offspring individuals by a mutation. Then the offspring of individuals are assessed, the best individuals from parent groups and mutated progeny groups are chosen to make new group. Evolutionary programming algorithm flow is shown in Fig. 1.

As can be seen from Fig. 1, the realization of EP algorithm mainly includes: determining expression of the problem; randomly generate initial population; calculating fitness value of the initial individual, calculating the fitness of a new individual and selecting these operation to generate a new group by mutation. Selecting the best individual as the optimal solution until a satisfactory solution is obtained or termination condition is satisfied.

Improvement process of algorithm: The main idea of the EPDV-Hop algorithm is: using evolutionary programming algorithm in place of trilateration to calculate the coordinate position of unknown nodes. The specific process is elaborated as follows:

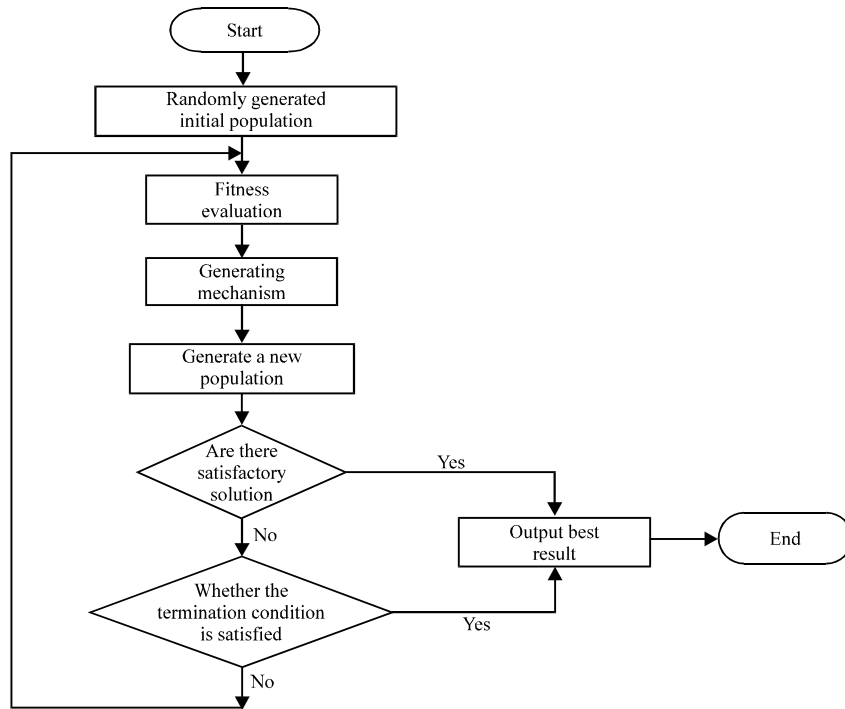


Fig. 1: Flowchart of EP algorithm

Position sampling: After the second stage, unknown nodes get the estimated distance to the anchor nodes by $\text{hop} \times \text{hopsizes}$. The initial estimate of the coordinate position of the unknown node is obtained by position sampling. Suppose the sampling position coordinates of the unknown node is $S(x_{\text{sample}i}, y_{\text{sample}i})$, then:

$$\begin{cases} x_{\text{sample}i} = x + r\delta_i \cos\theta_i \\ y_{\text{sample}i} = y + r\delta_i \sin\theta_i \end{cases} \quad (5)$$

where, δ_i, θ_i represent random parameters of the sampling radius and sampling angle, respectively, value intervals are $[0,1], [0, 2\pi]$, respectively. Select the sampling location which satisfies the Eq. 6:

$$d_j - \sqrt{(x_{\text{sample}i} - x_j)^2 + (y_{\text{sample}i} - y_j)^2} \leq \varepsilon \quad (6)$$

where, d_j is the distance between the anchor node j and the unknown node (x_j, y_j) is the coordinate of the position of the anchor node j . ε is the distance error, which can be artificially set, the value of which is set to be 3 m. If sampling location $S(x_{\text{sample}i}, y_{\text{sample}i})$ satisfies the Eq. 6, it is retained otherwise discarded. When the number of sampling locations reaches the threshold N_{eff} or reaches the pre-set maximum sampling number N_{max} , sampling stops, getting the number of samples N_s . The estimate of initial position of the unknown node can be calculated by the following equation:

$$\begin{cases} x = \frac{1}{N_s} \sum_{j=1}^{N_s} x_{samplej} \\ y = \frac{1}{N_s} \sum_{j=1}^{N_s} y_{samplej} \end{cases} \quad (7)$$

Location resample based on evolutionary programming: The initial position (x, y) of the unknown node is optimized by evolutionary programming algorithm. Selecting the appropriate position according to error analysis to make Eq. 4 take the minimum value. The objective function is:

$$G(x_i, y_i) = \min \sum \left(d_{ij} - \sqrt{(x - x_i)^2 + (y - y_i)^2} \right)^2 \quad (8)$$

Therefore, estimates of the position are converted into seeking the minimum problem. So the fitness function is chosen as:

$$Fit(x_i, y_i) = \frac{1}{[\alpha + \beta G(x_i, y_i)]} \quad (9)$$

where α, β is a constant coefficient fitness function, the value of which are set as $\alpha = \beta = 1$.

Setting the sampling location $S(x_{samplei}, y_{samplei})$ of unknown node as the initial population and resample location of unknown nodes is produced by evolutionary programming. Resample position of unknown nodes is assumed as $S(x_{samplei}, y_{samplei})$, then:

$$\begin{cases} S'(t+1) = EP(S, E, S'(t)) \\ S'(t) = S = (x_{samplei}, y_{samplei}) \end{cases} \quad (10)$$

where, EP is evolutionary programming, S is the selection operator, E is the mutation operator:

Designing mutation operator: First, each individual (x, y) is mutated by Eq. 11 using mutation operation of standard evolutionary programming:

$$\begin{cases} x(t+1) = x(t) + \sqrt{fit(t)} N_x(0,1) \\ y(t+1) = y(t) + \sqrt{fit(t)} N_y(0,1) \end{cases} \quad (11)$$

where, both (0,1) and $N_y(0,1)$ are normally distributed, random numbers which generated for direction of (x, y) and fit(t) is the fitness of older individuals.

Designing selection operator: Select operations using tournament selection method. I is referred to as set of $P(t) \cup P'(t)$, P(t) is a population which consists of μ parent individuals.

P'(t) is populations which consist of μ offspring individuals gotten after a mutation operation. For each individual $P_k = (x_k, y_k)$ of the set I, choose q other individuals from set I, compare the

fitness of P_k and fitness of q individuals, the number of individuals whose fitness is higher than fitness of P_k is referred to as score w_k of P_k . Choose μ highest scoring individuals from set I to form the new generation of population $P(t+1)$ after selecting.

Iterative optimization: When the evolution is completed, then enter the iterative optimization stage. The resample position are substituted into Eq. 8, to calculate a new estimate of the position $(x(t+1), y(t+1))$. Set $t \leftarrow t+1$, get into the next iteration. Terminate the iterative process until reaching the maximum number of iterations or satisfying the Eq. 12:

$$\sqrt{(x(t+1) - x(t))^2 + (y(t+1) - y(t))^2} \leq \tau \tag{12}$$

where, τ is optimization of residuals, which is set as 1 m in EPDRV-Hop algorithm.

After the node position estimations constantly adjust to each other, such that the objective function reaches a minimum value which can be expected to obtain more accurate location estimate.

Implementation steps of EPDRV-Hop algorithm: The implementation process of EPDRV-Hop algorithm is:

- Step 1:** Like the first stage of traditional DV-Hop algorithm, obtain the minimum number of hops between nodes
- Step 2:** Like the second stage of traditional DV-Hop algorithm, calculate the estimate of the distance between an unknown node and each of the anchor nodes
- Step 3:** It randomly generates the location of sampling point $S(x_{\text{sample}i}, y_{\text{sample}i})$, calculates the sampling points which satisfy Eq. 6 and obtains an initial estimate of the position (x, y)
- Step 4:** Sampling points generate a new re-sampling location $S'(x'_{\text{sample}i}, y'_{\text{sample}i})$ by mutation and selection. The new individuals are substituted into the fitness Eq. 9 to give new fitness; a new round of evolution is started, until a specified number of evolutions are completed
- Step 5:** Resample position which is generated by evolution is substituted into the objective function of Eq. 8. When the minimum value of the objective function is obtained, the old location estimate is replaced by the new position estimate producing the new objective function and do the next round of iteration until the termination condition is met
- Step 6:** The iteration is end and the final coordinate position of the unknown node is obtained

RESULTS AND DISCUSSION

Positioning performance simulation

Setup of parameters: In order to verify the positioning performance of EPDRV-Hop algorithm, we use MATLAB platform to do simulations. Suppose that nodes are randomly distributed in the 100×100 m square area. Distance difference ϵ is 5 m. The maximum sampling frequency is 1000. The threshold of the number of sampling locations is 100. Optimization of residuals τ is 1 m. Population size is 100. The value of parameter q is 30 and the communication radius is 15 m. Positioning error is used as experimental evaluation in this study. Assuming true coordinates and estimated coordinates of unknown node respectively are (x_t, y_t) , (x_e, y_e) , respectively and communication radius is R . The positioning error is defined as:

$$\text{err} = \frac{\sqrt{(x_e - x_t)^2 + (y_e - y_t)^2}}{R} \times 100\% \tag{13}$$

Comparing positioning results under different numbers of anchor nodes: Getting a high precision positioning effect requires sufficient number of anchor nodes but because of cost factors, the number of anchor nodes is generally limited. In the case that the total number of nodes is 200, the number of anchor node gradually increases from the 5 to 40. We compare the performances of EPRDV-Hop algorithm and DV-Hop algorithm, the simulation results of which are shown in Fig. 2.

As can be seen from Fig. 2, positioning errors of the two algorithms are gradually reduced with the number of anchor node increasing. When the coordinates of the unknown nodes are calculated by the DV-Hop algorithm, the resulting position is not further optimized, so the error is larger than that of EPRDV-Hop algorithm. The EPRDV-Hop algorithm uses evolutionary programming resample algorithm to estimate the coordinates of the unknown node, the error which caused by the number of hops and the average distance per-hop is reduced to a minimum and the result of the initial coordinate position is further iteratively optimized, so we can get the optimal location estimate, which makes positioning performance of EPRDV-Hop algorithm significantly better than that of the traditional DV-Hop algorithm. Comparing with DV-Hop algorithm under the same conditions, the positioning errors of EPRDV-Hop algorithm are reduced by between thirteen percent and eighteen percent.

Comparing positioning results of different nodes: Proportion of anchor nodes is maintained as unchanged ten percent. Compare positioning performance of EPRDV-Hop algorithm and DV-Hop algorithm by changing the total number of nodes. The simulation results are shown in Fig. 3.

As can be seen from Fig. 3, the positioning errors of the two algorithms gradually decrease with the number of nodes increasing. In a smaller number of nodes, EPRDV-Hop algorithm can achieve a high positioning accuracy and the positioning performance is better than DV-Hop algorithm. Under the same conditions, comparing with DV-Hop algorithm, the positioning errors of EPRDV-Hop algorithm are reduced by between fourteen percent and nineteen percent.

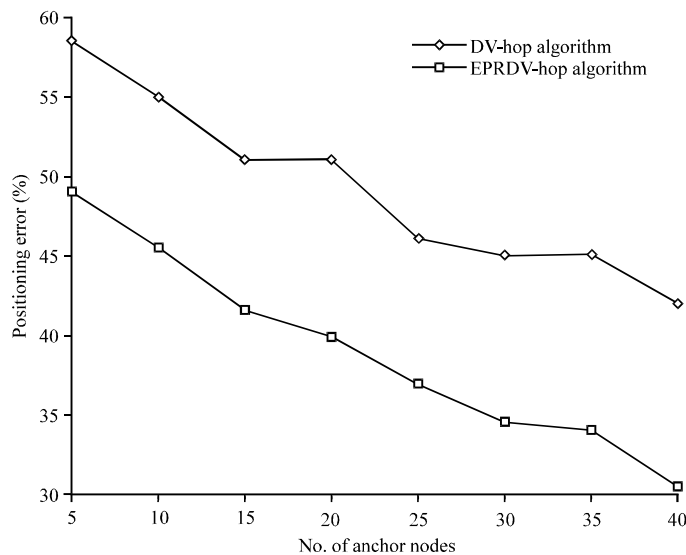


Fig. 2: Comparison of positioning errors of different anchor nodes

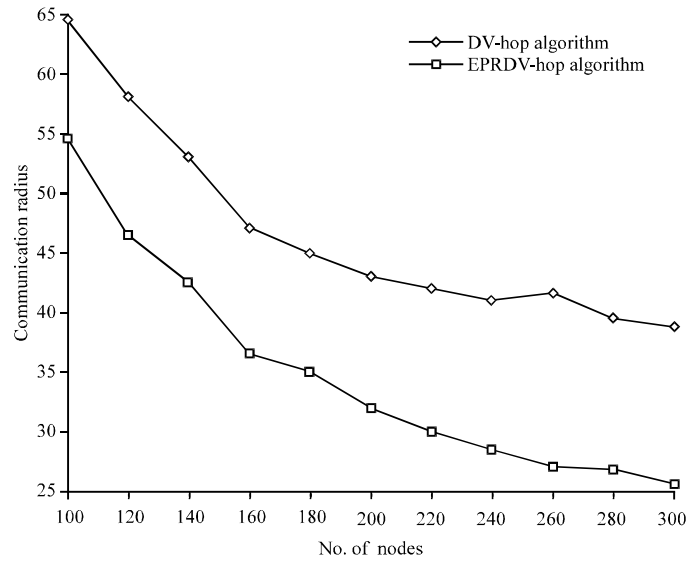


Fig. 3: Comparison of positioning error of different nodes

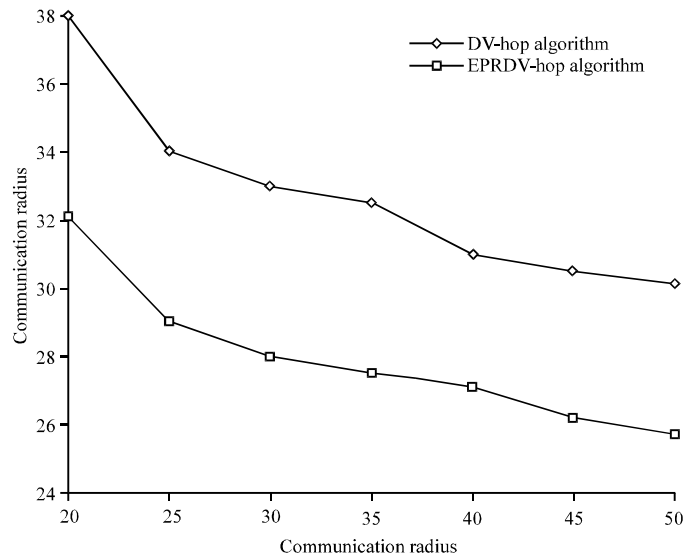


Fig. 4: Comparison of positioning errors under different communication radius

Comparing positioning results of different communication radius: Under the same conditions that the number of nodes is 200 and the ratio of anchor node is ten percent, we compare positioning performance of the two algorithms under the changing communication radius of nodes, the simulation results are shown in Fig. 4.

As can be seen from Fig. 4, communication opportunity to inter-node increases with the radius of communication increasing and the positioning errors of the two algorithms are gradually decreasing. But the positioning errors of EPRDV-Hop algorithm are significantly lower than DV-Hop algorithm (Zhang *et al.*, 2012; Wen *et al.*, 2014; Xiao and Liu, 2012; Feng *et al.*, 2012;

Wang and Shi, 2012; Wang *et al.*, 2011; Lu *et al.*, 2011; Li and Zhou, 2011; Li *et al.*, 2013); therefore, positioning errors of EPRDV-Hop algorithm are reduced by between thirteen percent and eighteen percent compared with DV-Hop algorithm.

CONCLUSIONS

By analyzing positioning errors caused by the third phase of DV-Hop localization algorithm, an improved DV-Hop algorithm based on evolutionary programming resample (EPRDV-Hop) is proposed aimed at solving the problem of greater error of using the least squares method to estimate the unknown node. EPRDV-Hop algorithm uses evolutionary programming resample to replace the least squares method to estimate the coordinates of unknown nodes. Evolutionary programming resample with ability of optimization search to make error of real-world coordinates and estimated coordinates to be a minimum, which significantly improves the performance of the positioning algorithm. This algorithm with no additional hardware equipment still has the advantage of low cost. In order to meet higher positioning accuracy requirements in WSN, amendments of hops and the average distance per-hop combined with evolutionary programming are considered to obtain higher accuracy in the future.

ACKNOWLEDGMENTS

The study of our work is supported in part by Young Talents Fund Project in Anhui Province of China (No. 2013SQRL083ZD), Anhui University Provincial Natural Science Research Project (No. KJ2014A247), Open platform topics of Suzhou University of China (No. 2012YKF38) and University student's Innovative Training Project (201310379019).

REFERENCES

- Feng, J., Q. Zhu and C.C. Wu, 2012. Research on improved DV-hop localization algorithm. *Comput. Eng.*, 38: 74-77.
- Fogel, D.B. and L.J. Fogel, 1996. An introduction to evolutionary programming. *Artif. Evol.*, 1063: 21-33.
- Li, M., W. Xiong and L. Guo, 2013. Improvement of DV-hop localization based on artificial bee colony algorithm. *Comput. Sci.*, 40: 33-36.
- Li, W. and W. Zhou, 2011. Genetic algorithm base localization algorithm for wireless sensor network. *Proceedings of the 7th International Conference on Natural Computation*, July 26-28, 2011, IEEE Press, Shanghai, China, pp: 2096-2099.
- Lu, Q., M. Bai, W. Zhang and E. Lian, 2011. A kind of improved DV-hop algorithm. *Proceedings of the 2nd International Conference on Intelligent Control and Information Processing*, July 25-28, 2011, Harbin, China, pp: 867-869.
- Niculescu, D. and B. Nath, 2003. DV based positioning in ad hoc networks. *Telecommun. Syst.*, 22: 267-280.
- Sun, L., J. Li and Y. Chen, 2005. *Wireless Sensor Networks*. Tsinghua University Press, Beijing.
- Wang, X.S., Y.J. Zhao and H.T. Li, 2011. Improved study based on DV-hop localization algorithm. *Comput. Sci.*, 38: 76-78.
- Wang, Y. and H.Y. Shi, 2012. Improved DV-hop localization algorithm for wireless sensor network. *Comput. Eng.*, 38: 66-69.

- Wen, J., X. Fan and X. Wu, 2014. Improved DV-Hop location algorithm based on hop correction. *Chin. J. Sens. Actuators*, 27: 113-117.
- Xiao, L. and X. Liu, 2012. DV-hop localization algorithm based on hop amendment. *Chin. J. Sens. Actuators*, 25: 1726-1730.
- Zhang, A., X. Ye, H. Hu and X. Ding, 2012. Improved DV-HOP positioning algorithm based on one-hop subdivision and average hopping distance modification. *Chin. J. Sci. Instrum.*, 33: 2552-2558.