An Improved DV-Hop Wireless Sensor Network Localization Algorithm

Xiaotie Qin[*]

Library, Chongqing University of Science and Technology, Chongqing, 401331, China

Abstract: The positioning accuracy is not high for the range-free localization algorithm DV-Hop in wireless sensor networks and the Particle Swarm Optimization algorithm is proposed to optimize the positioning accuracy of the DV-Hop and the IDV-Hop localization algorithm is also proposed. Firstly, the IDV-Hop utilizes DV-Hop to estimate the measuring distances between nodes to be positioned and the beacon node, then the Particle Swarm Optimization algorithm is used to maximize the IDV-Hop fitness function to obtain the location coordinates of each node to be positioned. Simulation results show that, compared to the DV-Hop algorithm and GADV-Hop algorithm, IDV-Hop algorithm provides higher positioning and can meet the requirements of most wireless sensor network applications.

Keywords: DV-Hop, IDV-Hop, Location Algorithm, Particle Swarm Optimization.

1. INTRODUCTION

In the wireless sensor network, monitoring application of network heavily depends on the geographical position of nodes. There are a large number of nodes in WSN, which are randomly arranged. It is very difficult to get the location information of nodes. Although it is possible to get location information of nodes by GPS, but the cost is too high, which is not suitable for large-scale use. Therefore, we need to adopt the appropriate localization algorithm to obtain location information of nodes [1].

 In the wireless sensor network, there are generally two classes of the localization algorithm of nodes: the range-based and the range-free [2]. The former needs to get information such as the orientation between neighbor nodes or absolute distance so that it can make use of the actual distance between nodes to estimate the position of the to-be-located nodes; the latter doesn't need to get information such as the orientation between neighbor nodes or absolute distance and instead it makes use of the estimated distance between nodes to estimate the position of nodes to be located.

Range-based localization algorithm, such as localization mechanism based on received signal strength (RSSI), angle of arrival (AOA), time of arrival (TOA) and time difference of arrival (TDOA), etc., needs to be equipped with additional hardware, which results in the increase in both orientation cost and energy consumption. However, range-free localization algorithm will not cause the increase of additional hardware cost, though its localization accuracy is not as high as that of the range-based localization algorithm. Yet its localization accuracy can satisfy the application of most wireless sensor networks [3].

In range-free localization algorithm, most scholars agree that it is a better way to estimate the distance between beacon nodes and unknown nodes by means of light-flooding. DA-Hop is one of the algorithms which have been studied most. In this paper, an improved DV-Hop localization algorithm is put forward, based on the research which focuses on optimizing the localization accuracy of DV-Hop algorithm by using Particle Swarm Optimization (PSO).

2. DESCRIPTION OF LOCALIZATION PROBLEMS

Suppose there are N unknown nodes and M beacon nodes in WSN, $q = [z_1, z_2 L z_{M+N}]$, standing for the initial position of nodes, wherein $z_i = [x_i, y_i]^T$, then the position of the M beacon nodes is $(x_1, y_1), (x_2, y_2) \cdots (x_M, y_M)$. So the localization problem means to estimate the position information of the unknown nodes in the network by using the known beacon nodes.

Suppose the communication model of the node is concentric zone model whose radius is R which is also the communication radius of R_f transceiver [4-6], the localization problem is the position of the unknown nodes $q = [q_x, q_y]$, according to the M Beacon nodes whose position is known, wherein

$$q_x = [x_{M+1}, x_{M+2} \underline{0} x_{M+N}], q_y = [y_{M+1}, y_{M+2} \underline{0} y_{M+N}].$$

In order to evaluate the performance of the localization algorithm, the localization error is described as follows:

$$f_i(\hat{x}, \hat{y}) = d_i - \sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2} \qquad (1)$$

Wherein $(\hat{x}, \hat{y})$ is the coordinate of the to-be-located node, the coordinate of the No.i beacon node is $(x_i, y_i)(i=1, 2 \bullet M)$, the measuring distance and the estimated distance between the to-be-located node and the No. i bea-

Fig. (1). Process of sending data packet by the beacon node.

con node are d_i and $\sqrt{(x_i - \hat{x})^2 + (x_i - \hat{y})^2}$, respectively. The localization problem is to minimize the localization error $f_i(\hat{x}, \hat{y})$ [7].

3. DESCRIPTION OF IDV-HOP ALGORITHM

The localization problem in wireless sensor networks is a complicated NP problem concerning the optimization of the distance value obtained in different ways. Many scholars at home and abroad have recommended Heuristic method, such as genetic algorithm, to optimize and increase the localization accuracy of nodes. Given that Particle Swarm Optimization (PSO) has fewer parameters to be adjusted, rapid convergence and more importantly facilitates parallel realization. In this paper, an improved DV-Hop localization algorithm is put forward, based on the research which focuses on optimizing the localization accuracy of DV-Hop algorithm by using Particle Swarm Optimization (PSO). The basic idea of IDV-Hop algorithm is to estimate the measuring distance between the unknown node and the beacon node by using IDV-Hop algorithm, make optimization in later period by using Particle Swarm Optimization (PSO) and then maximize the fitness function of IDV-Hop, thus fulfilling the optimization of the measuring method based on different distance or path.

3.1. Description of DV-HOP Algorithm

IDV-Hop algorithm [8] generally involves the following three steps:

(1) The to-be-located node obtains the minimum hop value.

Via sending the broadcast packet to the network, the beacon node forwards it to the neighbor node after receiving the hop counts of the node and adding 1 to the hop value. Thus, we can get the information about the minimum hop value between the to-be-located node and the beacon node in the network. As can be seen from Fig. (**1**), the beacon node A sends the broadcast packet to the network by means of concentric circles. The number in the figure represents the hop value of distance A.

(2) Count the number of light-flooding between the to-be-located node and the beacon node. According to the relevant information obtained from Step (1), every beacon node estimates the actual distance per average hop by using Formula (2):

$$HopSize_i = \frac{\sum_{j\neq i}\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j\neq i}h_j} \tag{2}$$

Wherein, $(x_i, y_i), (x_j, y_j)$ are the coordinates of the beacon nodes i and j, respectively, h_j is the number of light-flooding between the beacon nodes i and j $(j \pi\ i)$.

And then, the beacon node sends the broadcast packet containing the distance per average hop to the network. The to-be-located node receives and records the data package and then forwards it to the neighbor node, thus making most nodes get the broadcast packet from the beacon node closest to it. After the to-be-located node receives the data package, according to the hop counts recorded by itself, the light-flooding distance between the beacon nodes can be obtained.

(3) Count its own location

According to the light-flooding distance of each beacon node recorded in Step (2), the to-be-located node can calculate the position coordinate of the to-be-located node by using maximum likelihood method.

Suppose $A_1(x_1, y_1), A_2(x_2, y_2) \unicode{x00BA}\ A_n(x_n, y_n)$ stands for the position coordinate of n beacon nodes, the estimated distance between the to-be-located node D and each beacon node is $d_1, d_2 \bullet d_n$ respectively.

So, we can get the following formula:

$$\begin{aligned}(x_1-x)^2+(y_1-y)^2&=d_1^2 \\ &\vdots \\ (x_n-x)^2+(y_n-y)^2&=d_n^2\end{aligned} \tag{3}$$

In Formula (3), if the first formula subtracts the last one, then we have:

$$\begin{cases} x_1^2 - x_n^2 - 2(x_1-x_n)x + y_1^2 - y_n^2 - 2(y_1 - y_n)y = d_1^2 - d_n^2 \\ \qquad\qquad\qquad M \\ x_{n-1}^2 - x_n^2 - 2(x_{n-1}-x_n)x + y_{n-1}^2 - y_n^2 - 2(y_{n-1} - y_n)y = d_{n-1}^2 - d_n^2 \end{cases} \tag{4}$$

Formula (4) can be expressed as $AX = b$, wherein:

$$X = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$A = -2 \begin{bmatrix} (x_1 - x_n) & (y_1 - y_n) \\ (x_2 - x_n) & (y_2 - y_n) \\ M & M \\ (x_{n-1} - x_n) & (y_{n-1} - y_n) \end{bmatrix}$$
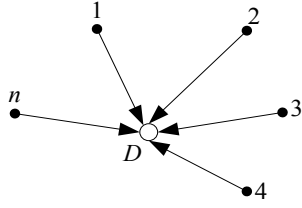
Fig. (2). Sketch map of maximum likelihood method.

$$b = \begin{bmatrix} d_1^2 - d_n^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 \\ d_2^2 - d_n^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 \\ M \\ d_{n-1}^2 - d_n^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{bmatrix}$$

And then the coordinate calculation formula of the node D is $\hat{X} = (A^T A)^{-1} A^T b$.

3.2. Particle Swarm Optimization

Particle Swarm Optimization (PSO) means to treat the individual in the group as a special particle, which has no volume or weight, and flies at a certain speed in the search space of solution. The individual makes a dynamic adjustment according to the flying experience of its own and the particle groups. In N dimensional space, the position and speed of the No. i particle is expressed by the following vector:

$$X_i = (x_1, x_2, \ldots, x_N) \quad V_i = (v_1, v_2, \ldots, v_N) \tag{5}$$

When the algorithm begins, the particle is initialized randomly in the solution space, and then it constantly searches and follows the optimal particle among particle groups in the solution space, and more importantly it updates itself by constantly tracking the optimal value pb_i the particle itself has experienced and the optimal value gb found in particle groups.

The position and speed of the particle i in the No. k iteration can be expressed as x_i^k and v_i^k, respectively. The speed and position update formula is as follows:

$$v_i^{k+1} = \omega \times v_i^k + c_1 \times r_1 \times (pb_i - x_i^k) + c_2 \times r_2 \times (gb - x_i^k) \tag{6}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{7}$$

In the formula, ω is inertia weight; c_1 and c_2 are acceleration constants, and generally the data range is $0 \sim 2$. r_1 - $U(0,1)$ and r_2 - $U(0,1)$ are two random functions independent of each other. pb_i is the best position that the particle i has experienced; gb is the best position the particle group has experienced. Particles keep learning and updating, and output the global optimum of the group when the search process terminates.

3.3. The Encoding of Algorithm and the Selection of Fitness Function

The encoding of problem solution and the selection of fitness function are the critical problems to be solved when using Particle Swarm Optimization (PSO) to optimize problems. In IDV-Hop algorithm, the encoding of problem solution is based on the position of the unknown node, i.e. each particle is the position of the N unknown nodes. Suppose the dimension of the sample vector is d, and in this paper, d is 2, then we'll use the following encoding structure:

$$x_{11} x_{12} L \ x_{1d} x_{21} x_{22} L \ x_{2d} L \ x_{N1} x_{N2} L \ x_{Nd}$$

$$v_{11} v_{12} L \ v_{1d} v_{21} v_{22} L \ v_{2d} L \ v_{N1} v_{N2} L \ v_{Nd}$$

$$f(\hat{x}, \hat{y})$$

In the above encoding structure, the first part denotes the position of the N unknown nodes, the second part denotes the speed of the N unknown nodes, and the third part denotes fitness function.

In this paper I propose the IDV-Hop algorithm by using Particle Swarm Optimization (PSO) to improve IDV-Hop. Its fitness function can be defined as:

$$f(\hat{x}, \hat{y}) = \frac{1}{1 + \sum_{i=1}^{M} \alpha_i^2 f_i^2(\hat{x}, \hat{y})} \tag{8}$$

In the formula, the weight of the measurement accuracy between the to-be-located node and the No. i beacon node is denoted by α_i, which is in inverse proportion to the hop value between the to-be-located node and the beacon node. This value is obtained from the DV-Hop measuring stage. Here set the value of α_i as the reciprocal value of the hop value. The estimated position of the to-be-located node can be fulfilled by maximizing the value of $f(\hat{x}, \hat{y})$, which helps minimize the localization error. Now the position of $(\hat{x}, \hat{y})$ is the localization result by using IDV-Hop algorithm.

IDV-Hop algorithm can be described as follows:

(1) Use the first and second stage in IDV-Hop algorithm to estimate the measuring distance and hop count between the unknown node and the beacon node;

(2) Set the initial r particle groups, calculate the value of fitness function $f(\hat{x}, \hat{y})$ of each particle, and initialize pb_i and gb;

(3) Compare the fitness of each particle with the fitness of the best position pb_i it has ever experienced. If it is better, then update pb_i;

(4) Compare the fitness of each particle with the fitness of the best position gb the particle group has ever experienced. If it is better, then update gb;

(5) Update the speed and position of the particle according to Formula (6) and (7);
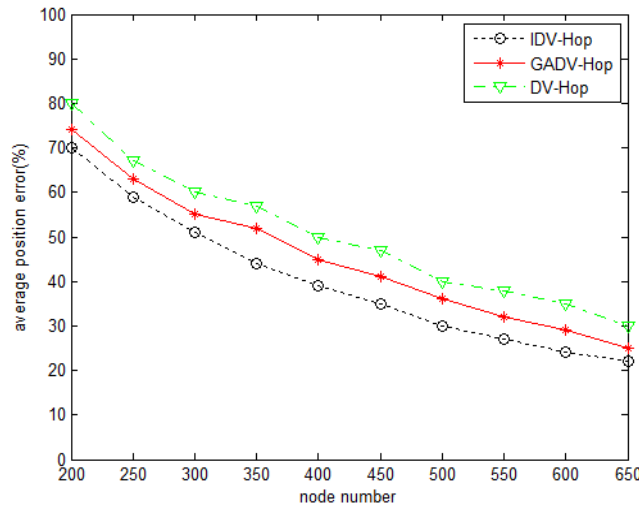
Fig. (3). Total numbers of to-be-located nodes-average localization error.

Fig. (4). The percentage of beacon nodes-average localization error.

(6) If terminating condition is satisfied, i.e. the predetermined maximum iteration number or good fitness, then finish it otherwise, turn to Step (3).

4. SIMULATION EXPERIMENT

In order to test the performance of IDV-Hop algorithm based on DV-Hop (GADV-Hop) localization algorithm in the genetic algorithm, a simulation comparison between IDV-Hop and DV-Hop is made by MATLAB [9]. In the simulation scene, which is set as an area of $100¥100$, the to-be-located node will be laid out randomly. As the performance of the beacon node located at the boundary is superior to the random layout, the beacon node is laid out at the boundary of the area. The emission radius of the radio frequency (RF) is set as 10 Hz. In this paper, average localization error is adopted to assess the performance of localization algorithm. Its definition is as follows:

$$error = \frac{100}{N \times R} \sum_{i=1}^{N} \sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2} \% \qquad (9)$$

In the formula, N is the number of the to-be-located node, the estimated position of the to-be-located node i is $(\hat{x}_i, \hat{y}_i)$, (x_i, y_i). $(i = 1, 2^{\underline{o}}\ N)$ is the actual position of the to-be-located node, and R is the RF radius.

This paper compares the performance of IDV-Hop localization algorithm and DV-Hop as well as DV-Hop (GADV-Hop) localization algorithm based on the genetic algorithm and analyzes the average localization error of various localization algorithms in different conditions.

(1) Change the number of the to-be-located node

Lay out some to-be-located nodes randomly in the area, and 8 beacon nodes are laid out evenly at the boundary of the area. We can see from Fig. (3), the average localization error of the three algorithms decreases as the number of the to-be-located nodes increases. This is because the connectivity of the network is increased due to the increase of the number of the to-be-located nodes, thus making the locali-

zation algorithm based on light-flooding more accurate. However, under the same conditions, the performance of IDV-Hop localization algorithm is superior to the other two algorithms.
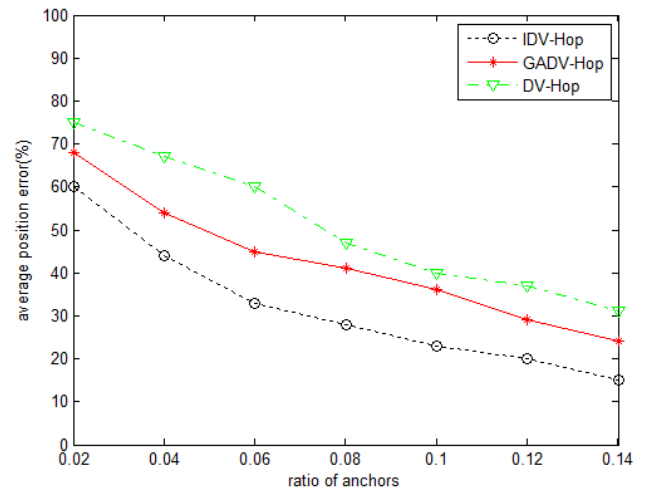
(2) Change the percentage of beacon nodes in the total number of nodes

Lay out the 150 to-be-located nodes randomly in the square area, and assess the performance of each localization algorithm by changing the percentage of beacon nodes in the total number of nodes. We can see from Fig. (4), with the increase of the beacon node percentage, the localization error of different algorithms decreases rapidly. However, compared with the other two algorithms, the average localization error of IDV-Hop localization algorithm is the smallest, and its performance can better satisfy the demand for localization accuracy applied in wireless sensor networks.

(3) Change the emission radius R of the node radio frequency

Lay out the 150 to-be-located nodes randomly in the square area, set the number of beacon nodes as 8, and assess the performance of each localization algorithm by changing the communication radius R of the node radio frequency. We can see from Fig. (4), the increase of the communication radius R makes better the connectivity of the network, thus decreasing the average localization error of different algorithms. However, under the condition that the same simulation parameters are set, the performance of IDV-Hop localization algorithm is superior.

CONCLUSION

In this paper, we propose an improved IDV - Hop localization algorithm, which utilizes DV-Hop to estimate the measuring distances between nodes to be positioned and the beacon node Then the Particle Swarm Optimization algorithm is used to maximize the IDV-Hop fitness function to obtain the location coordinates of each node to be positioned. Simulation results show that, compared to the
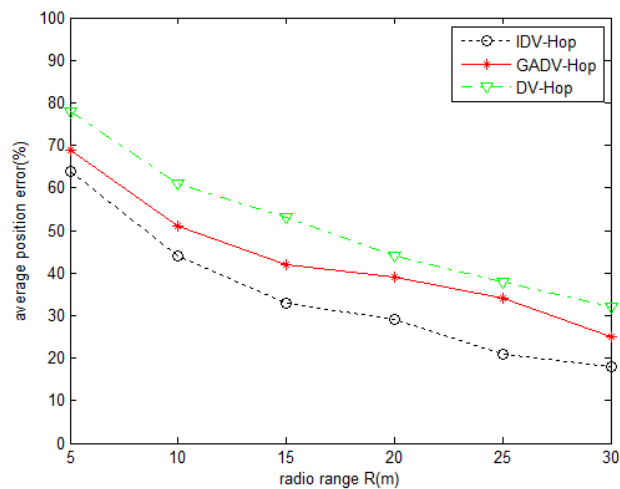
Fig. (5). Emission radius- average localization error.

DV-Hop algorithm and GADV-Hop algorithm, IDV-Hop algorithm provides higher positioning and can meet the requirements of most wireless sensor network applications.

CONFLICT OF INTEREST

The author confirms that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

REFERENCES

[1]   Z. Shijun, S. Meiling, and T. Yifang, "Gasa-based Localisation algorithm forwireless sensor networks", *Computer Applications and Software*, vol. 26, no. 10, pp.189-192, 2009.

[2]   N.B. Priyantha, H. Balakrishnan, and E. Demaine, "*Anchor-free distributed localization in sensor networks*", Technical Report MIT-LCS-TR-892 , MIT Lab for Computer Science , April 2003.

[3]   S. Zhi-Jun, Z. Peng, and Y. Haibin, "*Node localization problem in wireless sensor networks*", Computer Science, vol. 31, no.10, pp.35-38, 2004.

[4]   T. He, C. Huang , and B. Brain, *Range-Free Localization Schemes in Free in Large Scale Sensor Networks Large Scale Sensor Networks, Proc. 9th annual international conference on Mobile computing and networking (MobiCom), San Diego, California*, USA: ACM Press, 2003.

[5]   eu D. Nieules, B. Nath, Ad-hoc positioning system [A] //In: *Proceeding of IEEE Global Communications Conference (GLOBECOM)[C]* , pp. 2926-2931, 2001.

[6]   B. Feng'e, J. Xiaorong, and M. Huihui, "Research on DV-Hop Algorithm for Wireless Sensor Network", *Computer & Digital Engineering*, vol. 3, pp. 34-36, 2010.

[7]   C. Hao, L. Hong, "Research of particle swarm optim ization controller in brushless DC motor system", *Journal of Beijing Institute of Machinery*, vol. 23, no. 4, pp. 53-57, 2008.

[8]   D. Li, "Wireless sensor network based on genetic algorithm of localization algorithms", *Computer Simulation*, vol. 9, pp. 161-164, 2011.

[9]   F. Benbadis, K. Obraczka, and J. Cortes, Exploring landmark placement strategies for self-localization in wireless sensor networks[C] //*Proceedings of Personal, Indoor and Mobile Radio Communications*, 2007.