

An Improved Positioning Algorithm of Wireless Sensor Network Based on Differential Evolution

Wenli Lei^{1,2*} and Fubao Wang¹

¹*School of Electronics and Information, Northwestern Polytechnical University, Xi'an, 710072, China*

²*School of Physics and Electronic Information, Yan'an University, Yan'an 716000 China
lei.wenli@163.com*

Abstract

Localization as a key problem of wireless sensor network technology, has been the subject of widespread attention. This paper presents an improved positioning algorithm of wireless sensor network based on differential evolution(DE). Firstly, the algorithm improves on each jump distance measurement method in the traditional DV-hop algorithm, thereby reducing distance estimation error between anchor nodes and the unknown nodes, and then through the DE algorithm to obtain better positioning accuracy. Simulation results show that improved positioning algorithm has improved significantly in positioning accuracy compared to the previous DE positioning algorithm and least-squares algorithm, under different error factor and the density of anchor nodes. The algorithm has features of good robustness, global search capability and the ability to inhibit the accumulation of errors, suitable for a variety of applications in wireless sensor networks localization.

Keywords: *differential evolution, wireless sensor networks, locate, algorithm*

1. Introduction

Along with the progress of electronic communications and sensor technology, wireless sensor network has been widely used in the modern information society. Node location information is important information of monitoring message contained in wireless sensor networks, and is of great significance to obtain other information of nodes. Traditional positioning technology can be divided into range-free positioning and range-based position two ways in accordance with whether need for distance information. In the process of positioning based on distance positioning algorithm, it needs any additional hardware to complete the measurement of distances or angles between nodes [1,2]. This algorithm has high accuracy, but with the addition of hardware which makes cost higher. Generally based on distance of positioning algorithm has Received Signal Strength Indicator(RSSI) positioning, Time Difference of Arrival(TDOA) positioning, Angle of Arrival(AOA) positioning and Time of Arrival(TOA) positioning. Distance-independent algorithms without hardware support, according to communicate information through certain methods to estimate the distance between nodes, and classic distance-independent algorithm include DV-Hop algorithm, Centroid algorithm, APIT algorithm and so on [3,4]. The advantage of this method is low cost, positioning precision lower compared with the positioning algorithm based on distance, but can be optimized to meet the needs of most of the positioning, and therefore receives extensive attention.

Based on differential evolution algorithm, this paper presented an improved DV-Hop wireless sensor network localization algorithm which makes fuller use of the known anchor node position information to correct the unknown nodes localization information, without adding additional network hardware devices and further improve the positioning accuracy. The simulation results show that the proposed algorithm has better positioning effect [5,6].

2. Problem Description

Wireless sensor network node localization problem in essence is using M known anchor node position information to estimate the remaining N-M unknown nodes location information in the WSN network contained N nodes. In such a two-dimensional positioning problem, unknown node coordinates $V = (X, Y)$ where $X = [x_1, x_2, \dots, x_n]$, $Y = [y_1, y_2, \dots, y_n]$ can use the position coordinates of the anchor node $[x_{n+1}, x_{n+2}, \dots, x_{n+m}]$ and $[y_{n+1}, y_{n+2}, \dots, y_{n+m}]$ to calculate [7,8,9].

Based on positioning algorithm DV-Hop algorithm without ranging, anchor node and unknown node distance is estimated by minimum jumped number and average each jumped distance, which is not actual distance. Thus the estimated distance of unknown nodes and anchor nodes can be treated as an optimization problem, which through looking for minimum distance errors target function of anchor node and unknown node, to further get unknown node coordinate. In this paper, the mean square error of anchor nodes and unknown nodes is used as the fitness function. The fitness function is defined as follows:

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

Where (x, y) is the coordinates of the unknown node estimates, (x_i, y_i) are coordinates of the anchor node, d_i is the estimated distance of unknown nodes obtained by DV-Hop algorithm of the i -th anchor node. $\sqrt{(x_i - x)^2 + (y_i - y)^2}$ is actual distance between estimated coordinates and anchor node. Because in reality the estimated distance between the nodes is not their actual distances, so it should take the form of Gaussian error plus actual distance, namely:

$$d_i = d_{ij}(1 + randn \times \eta) \quad (2)$$

Where d_{ij} is the true value of the distance between two nodes, η is a distance error factor, and $randn$ is with mean 0 and variance 1 standard normally distributed random variables [10,11].

Seen from the above formula, the smaller the fitness function value, the smaller the error value, more accurate positioning of the unknown node. By calculating the fitness function minimizes the optimal solution, it can get actual position estimation of an unknown node, reduce error and improve the position accuracy of the unknown node.

3. WSN Node Location Algorithm Based on Differential Evolution Algorithm

3.1. Differential Evolution Algorithm

Differential evolution algorithm proposed in 1995 by two academics, the Storn and Price, it is a heuristic random searching algorithm based on group differences. Because of its ease of use, robustness and has strong global search capability, DE algorithm has become an effective tool for solving many real-world optimization problems. Like other evolutionary algorithms, the DE algorithm starts with a

random selection of initial population, its main process include mutation, crossover and selection three steps [12-13].

The basic idea of DE is to start from a set of random initial population, randomly select two different vector subtraction to generate difference vector, and then add difference vector which be given assigning weights to the third randomly selected individual vectors to produce mutation vector. Then mutation vector and predetermined parent individuals vector crossover to produce test vectors according to certain rules. If the fitness values of test vectors is better than the parent individuals fitness values, test vectors is selected into the next generation, otherwise leave the parent individual vector. Through continuous evolution, keep winning individual, direct your search to the optimal approximation.

Let D-dimensional continuous space vector $X = (X_1, X_2, \dots, X_D) \in S (S \subseteq R^D)$ is the decision vector of an optimization problem $f(x)$, and the optimization goal is find the optimal x^* making the objective function (or fitness function) $f(x)$ minimum, that is, to $\forall X \subseteq S$, there are $f(x^*) \leq f(x)$. The detailed procedure of DE algorithm is described as follows.

(1)initial population. DE algorithm is to search the global optimal solution in the D-dimensional real-valued continuous space. First, it need to select individuals contain NP initial population. Individuals vector in the population is called the destination vector, it use the following notation for the current generation of i-th objective vector:

$$X_{i,G} = (X_{1i,G}, X_{2i,G}, \dots, X_{Di,G}) \quad (3)$$

Where $G=0,1,\dots,G_{max}, G=0$ is the initialization vector populations, G_{max} is maximum number of generations.

In the initialization process, we often want the target vector can be distributed as uniformly as possible in the solution space. Therefore, the i-th target initialization vector of the j-th parameter can be written as:

$$X_{j,i} = X_{j,min} + rand_{j,i}(0,1) * (X_{j,max} - X_{j,min}) \quad (4)$$

Where $rand(0,1)$ is random numbers in $(0,1)$ which evenly distributed. $X_{min} = \{X_{1,min}, X_{2,min}, \dots, X_{D,min}\}$ representing the target vector lower boundary, $X_{max} = \{X_{1,max}, X_{2,max}, \dots, X_{D,max}\}$ representing the target vector upper boundary.

(2)Mutation operator. After the population initialization, DE through mutation operator to generate mutation vector V_{iG} . For each target vector X_{iG} , adopt certain mutation strategy generates the corresponding mutation vector $V_{iG} = (V_{1ij}, V_{1ij}, \dots, V_{Dij})$. In this paper, "DE / rand / 1" strategy is used to generate mutation vector.

$$V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}) \quad (5)$$

Among them, the $r1, r2, r3$ is a positive integer from $[1, NP]$ randomly selected and $r1 \neq r2 \neq r3$, for each mutation vector, which will randomly generate a positive integer. Scale factor F is a positive real constant, it controls the difference vector scaling, $F \in [0,2]$. If F value is large, it will result in the formation of the population has a higher diversity; If F value is small, algorithm can quickly converge.

(3)Crossover operator. After the mutation, crossover operator is introduced in order to enhance the diversity of the population. The crossover operation is to generate a new test vector $U_{iG} = (U_{1i,G}, U_{1i,G}, \dots, U_{Di,G})$ according to certain rules, for mixing each dimension parameter of each mutation vector and the target vector. In the DE algorithm, Binomial Crossover is often used, the mathematical expression is:

$$U_{ji,G} = \begin{cases} V_{ji,G}, & \text{if } \text{rand}_j(0,1) \leq CR \text{ or } j = k \\ X_{ji,G}, & \text{otherwise} \end{cases}, j = 1, 2, \dots, D \quad (6)$$

Where, $CR \in [0,1]$ is a real crossover probability constant value provided by the user, same with difference zoom factor F , is a control parameter of DE algorithm. $\text{Rand}(0,1)$ is a uniform random number in $(0,1)$; $k \in [1, D]$ and randomly selected to ensure that the test vector U_{iG} at least has one dimension which is contributed by mutation vector V_{iG} .

(4) Selection operator. DE algorithm uses the "greedy" selection strategy to generate offspring. After crossover and mutation operation, test vector U_{iG} and the target vector X_{iG} are competing, select the individual as the offspring with better fitness value. If the objective function is minimized, the selection procedure can be written as mathematical representation:

$$X_{i,G+1} = \begin{cases} U_{i,G}, & \text{if } f(U_{i,G}) \leq f(X_{i,G}) \\ X_{i,G}, & \text{otherwise} \end{cases} \quad (7)$$

Where $f(\cdot)$ represents the fitness function.

DE algorithm is mainly involved in population size NP , differential scale factor F and crossover probability CR three control parameters. Select the appropriate control parameters is critical to the performance of the algorithm. So far the research show that the population size $NP \in [4D, 10D]$ is more appropriate, scale factor and crossover probability in the range $F \in [0.4,1]$ and $CR \in [0.5,0.95]$ are more effective. $F=0.5$ and $CR=0.9$ can balance the contradiction between exploration and exploitation.

3.2. Positioning Algorithm

According to the foregoing description of positioning problem, traditional DV-Hop algorithm with jumped paragraph distance instead of actual distance, inevitably will appeared errors. In the process of anchor node calculation itself average each jumped distance, it calculates the quotient of the distance from the anchor node to other anchor nodes and jumped number, although participation calculation of distance value is accurate of actual distance, but this simple method has blurred the differences between individuals, when the uneven distribution of network nodes, obtained average hop distance would be difficult to reflect the overall situation of the network. In order to reduce positioning errors due to uneven distribution of network, this paper presents different anchor nodes, adaptive adjustments to the average value of each jump distance. Specific improvement methods are as follows:

(1) Firstly, according to the traditional DV-Hop algorithm, the average hop distance of each beacon node is calculated according to the following formula:

$$C_M = \frac{\sum_{N \neq M} \sqrt{(x_M - x_N)^2 + (y_M - y_N)^2}}{\sum_{N \neq M} h_{MN}} \quad (8)$$

C_M represents the average jump distance value of beacon node M , N is beacon node in addition to beacon node M , (x_M, y_M) , (x_N, y_N) respectively for the actual location coordinates of M and N ; h_{MN} is the smallest hop count of beacon node M and N which is obtained by the before stage seeking.

For unknown node k, beacon node M is closest beacon nodes, the k receives an average jump distance from M, remember:

$$C_{sk} = C_M \quad (9)$$

Among them, the C_{sk} is the average each jump distance based on unknown node k.

(2) In calculating the distance of unknown nodes k and beacon nodes M, as the two-node distance, still follows the classic jump distance calculation method in the DV-Hop method, that is:

$$d_{kM} = C_{sk} \times h_{kM} \quad (10)$$

Among them, the d_{kM} is hops distance between k and M nodes, h_{kM} is hops between two points.

(3)When calculating hop distance between k and other beacon nodes, introducing an average jump distance adjustment, and after introducing adjustments each jump distance calculation formula is as follows:

$$C_{kp} = \sqrt{C_{sk} \times \frac{d_{MP}}{h_{MP}}} \quad (11)$$

Wherein, C_{kp} represents the average distance per hop when calculated the distance between unknown nodes k and beacon nodes P used, d_{MP} and h_{MP} represent the actual distance and the smallest hop count between beacon node M and P.

As can be seen from the above equation, the algorithm in this paper based on the traditional DV-Hop algorithm, increasing the adaptive factor $\frac{d_{MP}}{h_{MP}}$, in all beacon node ,

the most close target unknown node k surrounding environment of beacon node is the nearest distance beacon node M; Therefore, the positional relationship between the M and the other beacon nodes close to the positional relationship between the unknown node with other nodes beacon. if in the process of estimation distance between the unknown nodes k and beacon nodes P, with accurate average distance per hop to adjusted based average distance per hop C_{sk} between M and P, according to the above formula to calculate a new average hop distance value, it can effectively play the effect of reducing the error results. This method can set the system to balance the overall situation and individual special circumstances, make the jump distance estimate more accurate.

Unknown node k to beacon node p jump distance estimates as:

$$d_{kp} = C_{kp} \times h_{kp} \quad (12)$$

4. Node Location Improved Algorithm Steps based on Differential Evolution

In summary, this paper proposes a new improved algorithm based on differential evolution algorithm of wireless sensor network node localization. The method implementation process is as follows:

1th step initialize population scale NP, set the DE algorithm parameters. Individual's initial position is set to nearest the unknown node that corresponds to the location of the beacon nodes. Random deployment of N unknown nodes and M anchor nodes in a network region, the scaling factor $F=0.5$ and crossover probability $CR=0.9$.

2nd step used the improved algorithm formula (8)-(12) in this paper to calculate the distance matrix D between the unknown nodes N and the anchor nodes M.

3rd step calculate the fitness value of each individual by the formula (1)-(2).

4th step according to the formula (3)-(7), all individuals in the group perform DE mutation, DE crossover, DE the selection operation.

5th step to determine whether reach the maximum number of iterations, if the condition is met, then output the best solution of corresponding individual, which is an unknown node position estimation coordinates, otherwise continue.

5. Simulation and Analysis

5.1. Simulation Description

In this paper, MATLAB 7.0 environment simulation experiment is used, set the total number of network nodes is 100 nodes to communicate with a radius of 30m. Unknown nodes and anchor nodes randomly generate and deploy in the 100m × 100m square area. Setting DE algorithm simulation parameters: population size NP = 20, the maximum number of iterations $t_{max} = 100$, the scaling factor F = 0.5 and crossover probability CR = 0.9.

5.2. Simulation Analysis

In the simulation, the effectiveness and practicability of the algorithm is verified by changing beacon nodes density and error factors. Experimental evaluation using the average location error and positioning accuracy to evaluate the performance of the algorithm, the formula is as follows:

Average location error

$$averageerror = \frac{\sum_{i=1}^N \sqrt{(x_i - x_i^*)^2 + (y_i - y_i^*)^2}}{N} \quad (13)$$

Positioning accuracy

$$accuracy = \frac{\sum_{i=1}^N \sqrt{(x_i - x_i^*)^2 + (y_i - y_i^*)^2}}{N \times R} \times 100\% \quad (14)$$

5.2.1. Old and new two algorithm performance comparison: In the second part of this paper, where introduces an improved new algorithm of positioning mechanism, performance verification via simulation. Figure 1 is performance comparison chart by using the new improved location algorithm and DE algorithms based on traditional DV-hop algorithm, where anchor nodes density is 30%, error factor is 25%.

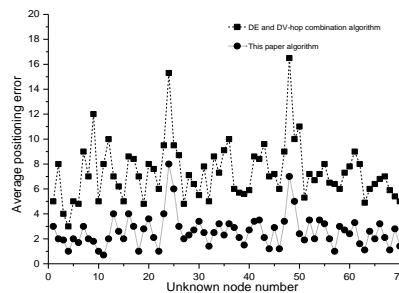


Figure 1. Old and New Two Algorithm Performance Comparison

By figure visible, improved new algorithm of positioning mechanism compared with the traditional position DE algorithm based on DV-hop algorithm, reducing positioning errors of the unknown node.

5.2.2. Comparison of different anchor node density: Anchor nodes is an important indicator affecting the positioning accuracy, in the simulation process, while keeping the other parameters do not change under anchor node density at intervals of 5% from 10% to 40%, observing its impact on positioning algorithm error.

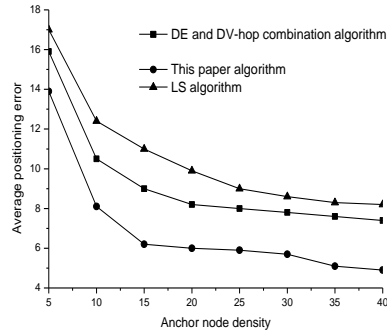


Figure 2. Comparison of Different Anchor Node Density

By Figure 2 visible, positioning errors of unknown node always decreased with anchor node proportion increased, in the same average location error, the new positioning mechanism algorithm improved compared to traditional position DE algorithm during DV-hop algorithm in use, requires fewer anchor nodes, thus, on anchor node of effective using aspects, due to the improved in this paper, algorithm make more fully effective use of anchor node, to save positioning cost.

5.2.3. Comparison of different error factor: Error factor has direct impact on the estimating distance. In the process of simulation, comparing performance of proposed improvement algorithm and other relevant location algorithm by changing the error factor from 0% to 20%, stride length selected 5%. Results as shown in the Figure 3, it can be seen from the figure under the same error factor, positioning error of the location method based on improved DE algorithm is smaller, positioning accuracy is higher, when error factor increases, the effect is more obvious.

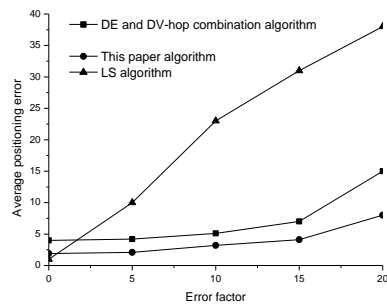


Figure 3. Comparison of Different Error Factor

5.2.4. Performance comparison of three kinds of positioning algorithm: Performance comparison of this paper algorithm, least squares algorithm and traditional DE algorithm based on DV-hop shown in the Figure 4, where anchors node density is 30%, error factor is 25%. By figure visible, the algorithm proposed in this paper is obtained the average positioning error is minimized, positioning accuracy is the highest.

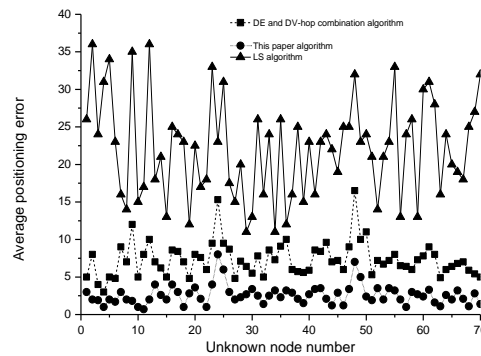


Figure 4. Performance Comparison of Three Kinds of Positioning Algorithm

6. Conclusion

This paper presents an improved positioning algorithm of wireless sensor network based on differential evolution algorithm. Firstly, the algorithm improves on each jump distance measurement method in the traditional DV-hop algorithm, thereby reducing distance estimation error between anchor nodes and the unknown nodes, and then through the DE algorithm to obtain better positioning accuracy. Simulation results show that improved positioning algorithm has improved significantly in positioning accuracy compared to the previous DE positioning algorithm and least-squares algorithm, under different error factor, the density of anchor nodes. The algorithm has features of good robustness, global search capability and the ability to inhibit the accumulation of errors, can effectively reduce the number of anchor nodes in wireless sensor networks deployment, control network system costs, and has a certain reference value for the practical application of wireless sensor networks positioning.

Acknowledgments

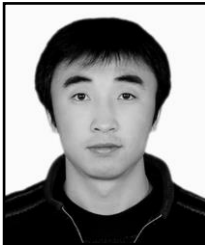
The authors would like to thank all the reviewers for their constructive comments. The authors acknowledge with thanks the Scientific Research Program Funded by Shaanxi Provincial Education Department (Program No. 14JK1829), Yan'an Municipal Science and Technology Research and Development Program (Program No. 2014KG-04), the Science and Technology Research and Development Program Project in Shaanxi Province (Industrial Research) (Grant No.2014K05-61), and the National Nature Science Foundation of China (Grant No. 61379026).

References

- [1] H. Chen, K. Sezaki, P. Deng and H. C. So, "An improved DV-Hop localization algorithm for wireless sensor networks", ICIEA 2008, 3rd IEEE Conference on Industrial Electronics and Applications, (2008), pp. 1557-1561.
- [2] J. Brest, S. Greiner, B. Boskovic, M. Mernik and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems", IEEE Transactions on Evolutionary Computation, vol. 10, no. 6, (2006), pp. 646-657.
- [3] H. Chen, K. Sezaki, P. Deng and H. C. So, "An improved DV-hop localization algorithm for wireless sensor networks", IEEE Conference on Industrial Electronics and Applications(ICIEA2008), (2008), pp. 1557-1561.
- [4] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive", IEEE Transactions on Evolutionary Computation, vol. 13, no. 5, (2009), pp. 945-958.
- [5] T. He, C. Huang, B. M. Blum, J. A. Stankovic and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks", Proceedings of the 9th annual international conference on Mobile computing and networking, ACM, (2003), pp. 81-95.

- [6] W. W. Ji and Z. Liu, "An improvement of DV-Hop algorithm in wireless sensor networks", *Networking and Mobile Computing, WiCOM 2006, International Conference on Wireless Communications*, IEEE, (2006), pp. 1-4.
- [7] N. B. Priyantha, H. Balakrishnan, E. Demaine and S. Teller, "Anchor-free distributed localization in sensor networks", *Proceedings of the 1st international conference on Embedded networked sensor systems, ACM*, (2003), pp. 340-341.
- [8] S. Kumar and D. K. Lobiyal, "An advanced dv-hop localization algorithm for wireless sensor networks", *Wireless personal communications*, vol. 71, no. 2, (2003), pp. 1365-1385.
- [9] R. Nagpal, H. Shrobe and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network", *Information Processing in Sensor Networks*, Springer Berlin Heidelberg, (2003), pp. 333-348.
- [10] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks", *Telecommunication Systems*, vol. 22, no. 1-4, (2003), pp. 267-280.
- [11] M. Li and Y. Liu, "Rendered path: range-free localization in anisotropic sensor networks with holes", *Proceedings of MOBICOM*, (2007), pp. 51-62.
- [12] R. Mallipeddi and P. N. Suganthan, "Empirical study on the effect of population size on differential evolution algorithm", *CEC 2008, (IEEE World Congress on Computational Intelligence), IEEE Congress on Evolutionary Computation*, IEEE, (2008), pp. 3663-3670.
- [13] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis", *Artificial Intelligence Review*, vol. 33, no. 1-2, (2010), pp. 61-106.

Authors



Wenli Lei, he is a lecturer and a Ph. D candidate. He received the M.S degree from Xi`an University of Science and Technology in 2008. His research interests include wireless sensor network, artificial intelligence, and so on.



Fubao Wang, he is a doctor, professor, Ph.D. supervisor. His research interests include computer networks, network and information security, wireless communication networks, digital multimedia communications and so on.

