

Design and Realization of the Advanced Distributed Localization Algorithm Based on MDS-MAP for WSNs

Zhang Lu^{1,*}, Wang Hailun¹ and Yang Mingxia¹

¹Quzhou University, College of electrical and information engineering, Quzhou
Zhejiang 324000, China
Zhanglu90573@163.com

Abstract

Aiming at the deficiencies of the classical MDS-MAP algorithm in Wireless Sensor Networks, an advanced distributed localization algorithm(AMDS-MAP(D)) was designed and realized based on classic MDS-MAP algorithm in this paper. This algorithm is implemented mainly through the node classes, the main variables which is used in the algorithm was defined in the node classes, while the main messages did. The realization of the algorithm was given and the complexity of the algorithm was obtained through the analysis. Theoretical analysis and simulation experimental results show that, the distributed localization algorithm based on MDS-MAP algorithm can realize distributed computing and improve the positioning accuracy of the node.

Keywords: Wireless Sensor Networks (WSNs); Advanced Distributed MDS-MAP localization algorithm(AMDS-MAP(D)); Node class; Algorithm design; Algorithm realization

1. Introduction

Introduction in recent years, with the progress of wireless communication, microelectronics and embedded computing technology, promote the development of low cost, low power consumption and multifunctional sensor nodes, so that it can be in a small volume of integrated information collection, data processing and short distance wireless communication multi-function [1]. Wireless sensor network integrates sensor technology, embedded computing technology, wireless network communication technology, distributed information processing technology and microelectronic technology can through the various integrated micro sensor collaboration to real-time monitoring, sensing and collecting and monitoring object information, through the embedded system of information processing, and through random self-organized wireless communication network in a multi hop relay the information transmitted to the end users, so as to realize the "ubiquitous computing" concept [2].

Although the huge human, material and financial resources have been put into the research of wireless sensor networks, there are still many problems to be solved, especially the problem of node localization [3]. Firstly, node localization is the basis of application. In some applications such as environmental monitoring, target tracking, no node's specific location information, all of which is meaningless; secondly, node localization is also a technical difficulty. In wireless sensor networks, number of nodes is large and aircraft spilled the position of each node is distributed randomly. At the same time, in order to reduce the cost as much as possible, each node processor performance, memory capacity, the communication distance of wireless transceiver and battery energy are extremely limited [4,5]. For each node is equipped with GPS (Global Positioning System) is not reality is not feasible, the first such will greatly increase the cost of the whole network, contrary to the original intention of wireless sensor network with low cost; the second GPS for the use of the environment with certain restrictions,

such as in the water or in the building cannot directly use. Therefore, there must be a high precision, distributed, low complexity and strong fault tolerant ability localization algorithm to obtain all the unknown node location, on the one hand, can improve the performance of wireless sensor networks, on the other hand can also reduce costs, is conducive to its large-scale application.

Columbia University Yi Shang et al. MDS technology is applied to the Ad-Hoc network positioning, put forward MDS-MAP, MDS-MAP (P), MDS-MAP (C) and other positioning algorithms [6]. Because it has no need of anchor nodes, it can obtain very high positioning accuracy, and is widely used in wireless sensor networks. In this paper, the requirements of the project, the positioning accuracy is higher, the amount of calculation is small, so the use of MDS-MAP algorithm. However, this algorithm is a centralized algorithm, the computation and communication costs are very large, so it is not suitable for large scale network, in order to meet the needs of the project, the corresponding improvement of the algorithm is necessary. [7-12].

This paper design and implement a distributed localization algorithm based on MDS-MAP algorithm, the algorithms don't need auxiliary hardware facilities, can be used to compute local to the nodes, so as to reduce the calculation amount and the amount of communication, and is applicable to shaded environment.

2. The Design of AMDS-MAP (D) Location Algorithm

2.1. AMDS-MAP(D) the Establishment of Algorithm

The purpose of the establishment phase is to solve the problem of the need to assist the hardware facilities in the process of distance measurement. In [3], a new measurement algorithm is proposed. It does not require the support of hardware facilities, can be carried out at the local calculation of nodes, thereby reducing the amount of computation and communication, and applies to sheltered environment. To sum up, this method is also used in this paper. This stage only gives a rough estimate of the location of the nodes, in order to get the initial approximation of the network topology.

(1)The algorithm of distance measurement

Step 1, according to the wireless radio frequency transceiver wireless range R to quantify of the distance, and then according to the node's quantization level and neighbor node response information to prepare the node's nearest neighbor vector (CNV, Close-Neighbor Vector) . As shown in Figure 1.

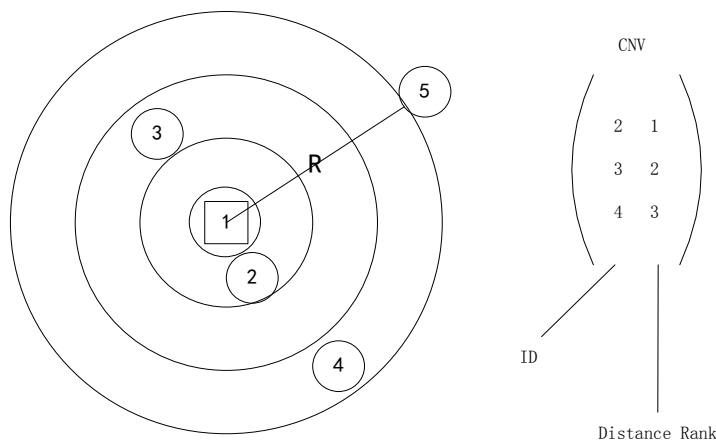


Figure 1. Distance Quantization and Close Neighbor Vector (CNV)

Step 2, the node's CNVs, run the estimation algorithm, according to these CNV compiled into a connected matrix C, and then through the connected matrix C to calculate the distance matrix D. As shown in Figure 2.

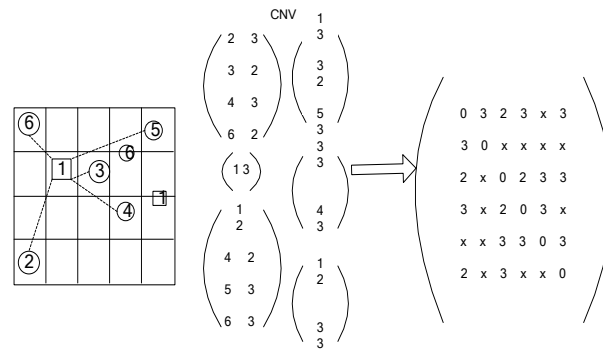


Figure 2 Generation Distance Matrix D

(2) Distributed arithmetic

Assume that the network has a node triggered the whole calculation process, the node acts as the origin of the role, also according to the CNV in around it n neighbor node can calculate the coordinates of the neighbor nodes, where n is a threshold. Based on these CNVs, the origin node can calculate its coordinates in the local. Obviously, when the number of threshold n is large enough calculated results will be accurate enough, but the threshold n increases, the bandwidth consumed more. Therefore, it is necessary to have a suitable number to compromise accuracy and bandwidth consumption, through simulation analysis, it is found that, when the threshold $n = 30$ can obtain a good result. The flow of the distributed algorithm is shown in Figure 4, Figure 3.

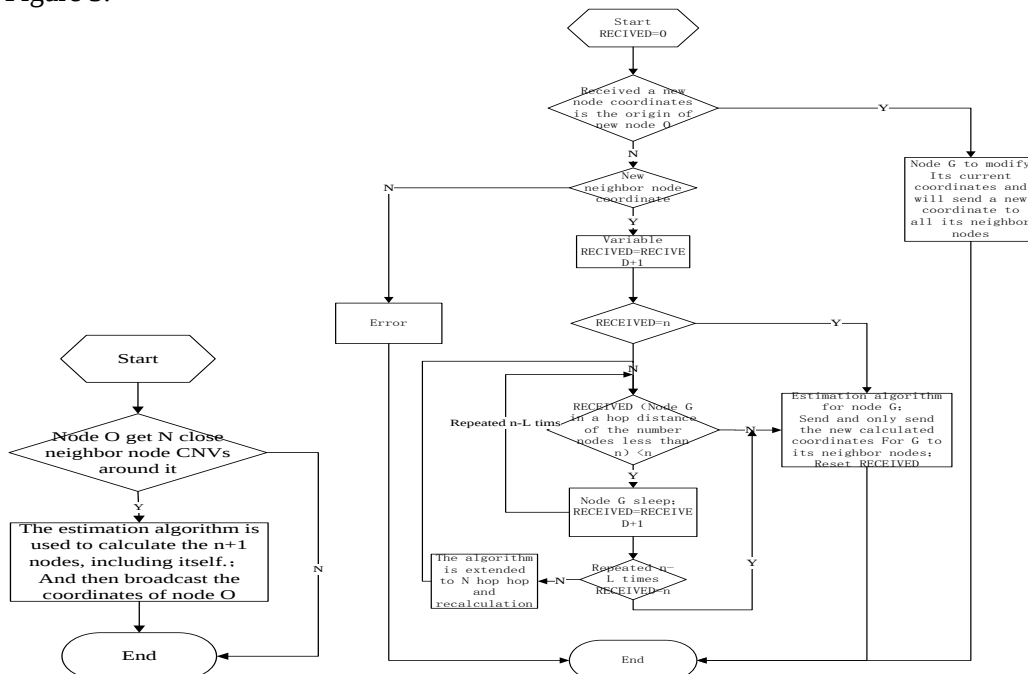


Figure 3. Flow Chart of the Origin Node Algorithm

Figure. 4 Flow Chart of General Node Algorithm

2.2. Refinement Algorithm

The previous stage just initial values of the node position, in order to get more accurate position value, the introduction of refinement step of the algorithm, the purpose is the use of the estimated distances between nodes get more precise location value [8]. Refinement is performed in an ad-hoc network environment, so only to the distance of nearest neighbor (one hop distance) is considered. The restrictions make refinement algorithm can be extended to networks of arbitrary size and in does not support multi hop routing (requires only local radio) in the underlying network operation.

Refinement is a recursive algorithm, the nodes through a series of steps to update its location. In at the beginning of each step, nodes broadcast their location estimates and from neighbor nodes receive location value and the corresponding estimated distance and by solving the least square method and triangulation algorithm to decide its new position. In most cases, the constraint imposed by the distance to the neighbor's position will force the new position closer to the true location of the node. After many cycles, the location update becomes small, and the actuarial method terminates and reports the final location.

Refinement algorithm has the advantages of relatively simple, it is due to the advantages of limits its use, especially in the premise is not quite sure of using refinement algorithm may not be able to achieve accurate results, many factors will influence the refinement algorithm convergence and accuracy, so in the traditional refinement algorithm introduced error threshold and eliminating bad nodes of the control.

3. Implementation of AMDS-MAP (D) Location Algorithm

Algorithm is realized by node class, node function is mainly receiving messages, local computation and message is issued. Nodes algorithm divided into two kinds: anchor nodes and unknown nodes, to start the algorithm, all nodes will by the interface StartMessage() received MSG_STRAT' message. The message from the network class, it will initialize all anchor nodes, and other variables, namely to the anchor node position which gives value. After that, all subsequent messages through Message () function to receive and process, until it receives a MSG_STOP message. The message by another function method StopMessage () deal with.

(1) Main variable in a class of nodes

This : The ID value of the current node

Position : Node's current location of the valuation, the storage mode for floating point array and array size corresponding to the location of the dimension.

Status : State node status

At present there are four kinds of state : STATUS_ANCHOR: The node knows its location.STATUS_UNKNOWN: The node failed to estimate its location.

STATUS_POSITIONED: The node has already estimated its position.

STATUS_BAD: The node is either invalid or unable to locate the node (the specific definition is different depending on the location algorithm).

Anchors : Anchor node information, storage mode for the linked list. Each table entry is a structure anchor_info, that has an anchor node information, as shown in Figure 5 (a). Its information includes the anchor node ID, anchor node position value Position, the number of hops to the anchor node count hop_, the last hop node ID and the distance from the anchor node to the path of the current node on the last hop lasthop_dis. If the current node is the anchor node, the first item of the list is stored in its own information, in the initialization is inserted in the list.

Neighbors : Neighbor node information storage mode for the linked list. Each table has a structure that has a neighbor node information, as shown in Figure 5 (b). The information includes the neighbor node ID, the neighbor node state Status, the neighbor node error degree Error, to the neighbor node distance, indicates the neighbor node neighbor vector state CNV_status.

CNV_status : The nearest neighbor vector state of the node is a floating point array. Array for two-dimensional, respectively, the nodes of the ID and quantization levels.

Error : Error of node.

Refine count : The current node refinement times.

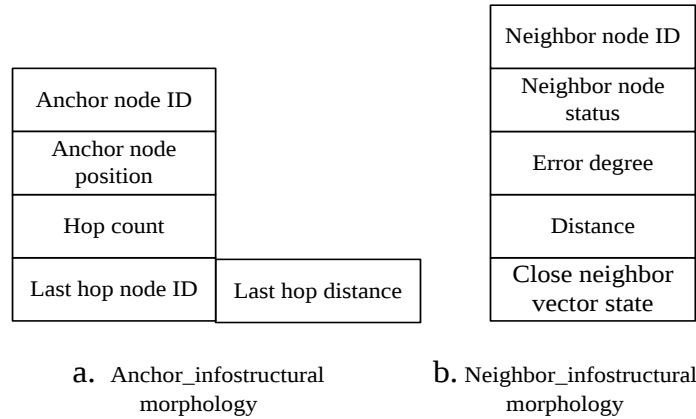


Figure 5. The Corresponding Data Items in the Location Algorithm

(2)Main message

In the algorithm, there are two kinds of message MSG_ANCHOR and MSG_POSITION, as shown in Figure 6 . MSG_ANCHOR is a message with anchor node location information, which includes all the anchor node information related to the original node. And MSG position with news of the position information of nodes, it is equipped with the degree of the node position estimation and error, and incidentally its neighbor table (CNV).Each anchor node and neighbor node in the two messages are represented by the structure Anchor_info and Neighbor_info.

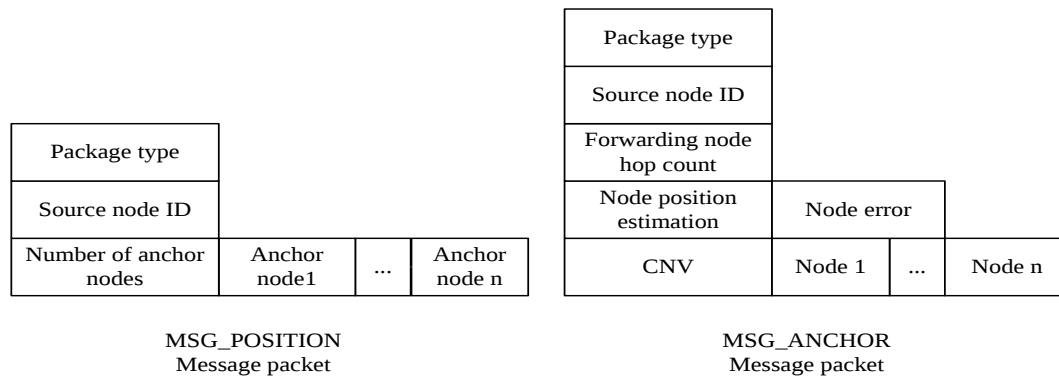


Figure 6. Corresponding Message Packets in Localization Algorithm

(3)Realization method

Message() : Receive messages and make corresponding processing. If the node is an anchor node, call the anchor () function, and call the unknown () function on the contrary.

Anchor() function to deal with all the information related to the anchor node.

```

        // function Anchor()
        Switch Message type
        Case MSG_ANCHOR
        If new message is new anchor()
        update() ; // Update the new anchor node information
        Case MSG_POSITION
        Break ;

New_anchor() Function to determine whether or not the anchor node in the package
is a new anchor node that is not used or not.
        //function new anchor()
        For ( i=0 ; i< Number of anchor nodes in message ; i++)
        Take out an anchor node from the message ;
        Anchor's hop++ ;
        If anchor->ID==Anchor node linked list ID
        End loop ;
        If the node was not found in the anchor node list. (anchors)
        //update anchor's message
        Update anchor-> The last hop node ID ;
        Update anchor-> Last hop distance ;
        Insert the anchor into the list ;
        //Set broadcasting anchor message timer
        reset Timer (anchor) ;
        Else
        If anchor hop< Number of hops in a table
        Operation as above ;

unknown() Function to deal with all the information related to the unknown node.
        //unknown()function
        Switch Message type
        Case MSG_ANCHOR
        If new_anchor()
        caculate();//According to the new anchor node positioning calculation
        // Set the timer for the broadcast node to estimate the location;
        resetTimer(position) ;
        break ;
        Case MSG_POSITION
        Update neighbor information new_neighbor() ;
        If neighbors.length> dimension and neighbor-> Hop < Forwarding node hop count
        do_caculate() ; // Positioning calculation for unknown nodesbreak ;

        New_neighbor() function will receive the information of neighbor nodes added to the
        neighbor node information table and update the corresponding information, updated
        node information is stored in the array update[], and will update the information in
        MSG_POSITION are broadcast out.
        //new_neighbor()function
        If the new node ID is not in the neighbor node information table
        The node is inserted into the neighbor node information table;
        Update related information in information table;
        Collection of new node CNV update new node CNVs ;
        To update update[] ;
        update->update=true ;
        radio broadcast MSG_POSITION ;
        Else

```

```

Update neighbor node information table ;
    To update update[] ;
    update->update=true ;
    radio broadcast MSG_POSITION ;

```

Caculate() and do_caculate() function is the function of node localization, The caculate () function is a function which is used to locate the new anchor node. However, do_caculate () function is to add the new unknown node to the whole network to locate the function.

All node information is finally broadcast through the sendanchor () and sendposition () function, these two functions are triggered by the timer. In the function sendposition (), in addition to the broadcasting node in the new location of the information later, but also incidentally the neighbor vector table sent out, and decide whether it is necessary to enter the second phase of the refinement algorithm.

```

//sendposition()function
Waiting for a period of time waiting for the algorithm to stabilize
    If Error threshold error=1
        Return ;
If The number of refinement > The maximum number of refinement
    Return ;
    Package message with error and position values ;
        If (update->update)
            The nearest neighbor vector table is filled in the message. ;
            Update->update=false ;
            send message ;

```

4. Algorithm Complexity Analysis

4.1. Computational Cost

In the calculation of the distance measurement, the main calculation is focused on the generation of the distance quantization and the connected relation matrix. For a nodes, each node will perform the a times quantization, and note that in last section all the anchor nodes assumed can reach all the nodes, which is still set up here. After a times quantization, the a CNVs are calculated, and the connected relation matrix is generated.

In the refinement algorithm, each node performs the least square method and generates a new error matrix in each iteration. Assuming the algorithm after n circulation, the calculation of Least square method is $D*(c-1)+(c-1)*(2*D+1)+(c-1)*2*(D+1)+D^2*(c-1)+D^3/3=(c-1)(5D+3)+(c-1)*D^2+D^3/3$, where C is all nodes in one hop neighborhood, the computation of each error matrix requires C additions and 1 Division. The total number of calculations required for each node in the refinement algorithm is $n[(c-1)(5D+3)+(c-1)*D^2+D^3/3+c+1]$.

4.2. Energy Cost

BCt, BCr and F are used to represent the transmission energy consumption, primary energy consumption and single calculation (flop) energy consumption. (related to a specific system).

(1)Measurement algorithm

Communication energy cost $a*BCt+ac*BCr$

Calculation energy cost $a[1+(a-1)(D^2+5D+3)+D^3/3]*F$

(2)Refinement algorithm

Communication energy cost $n*BCt+nc*BCr$

Calculation energy cost $n[(c-1)(D^2+5D+3)+D^3/3+c+1]*F$
 (3) Total cost
 Communication energy cost $(a+n)*BC_t+(ac+nc)*BC_r$
 Calculation energy cost $\{(a+n)(1+D^3/3)+[a(a-1)+n(c-1)](D^2+5D+3)\}*F$

4.3. Algorithm Complexity

Assuming the total number of nodes is n , the number of anchor nodes is m , for a $n*n$ matrix, the time complexity of measurement algorithm is $O(e*n*\log n)$, where e is the number of edges in the graph calculated according to the matrix. Each node in the distributed algorithm uses n auxiliary nodes and 3-dimension embedding dimension at most, so the computational complexity of each node is $O(e*n*\log n+n^2)$. Dijkstra algorithm is to calculate the shortest path distance matrix, the time complexity is $O(n^2)$. The time complexity of multidimensional scaling algorithm is $O(dn^2)$, where d is the number of embedding dimension. It uses m anchor nodes to calculate the relative coordinates to the absolute coordinates of the conversion parameters for the complexity of $O(m^3)$, completion the relative coordinates to the absolute coordinates for the complexity of $O(n)$.

5. Simulation Experiment

Design of AMDS-MAP (D) algorithm using MATLAB simulation, set the environment for in the region of 100×100 randomly generated 200 nodes, the nodes in a wireless communication radius of 10. In a network of 200 nodes, all data points is contained in more than 100 times the experimental results of the average value. In order to reduce the error of the distance between the simulation nodes, the distance value is subject to the standard distribution of the mean of the real distance.

In this paper, the simulation experiment is done under the condition of the distance measuring error 2.5%. If the range error is increased, the node localization accuracy will decrease, but the node localization accuracy will be improved if compared with the original algorithm. Due to the limitation of experimental conditions, to change the location error need to build the experimental platform, this did not elaborate. This is the next step to study the focus.

In the simulation experiment, the "anchor node 5%" is the 5% of the total number of anchor nodes. The percentage of "unknown node localization" refers to the location accuracy of unknown nodes;

Defined mean error is defined as the formula (4-1).

$$error(d) = \frac{\sum_{i=a+1}^n ((x_{est_i} - x_{real_i})^2 + (y_{est_i} - y_{real_i})^2)}{(n-a) \times R(d)} \quad (4-1)$$

Among them, n and a respectively represent the total number of sensors and the total number of anchor nodes, $R(d)$ for the node's wireless communication radius. (x_{real}, y_{real}) is the real position and (x_{est}, y_{est}) is the estimated position. The lower the average error is, the better the method is.

(1) Measure algorithm error experiment

This experiments are in the range error of 5%, using the average error formula (4-1) calculation, select the anchor node ratio is 5%, 10% and 20%. As shown in Figure 7.

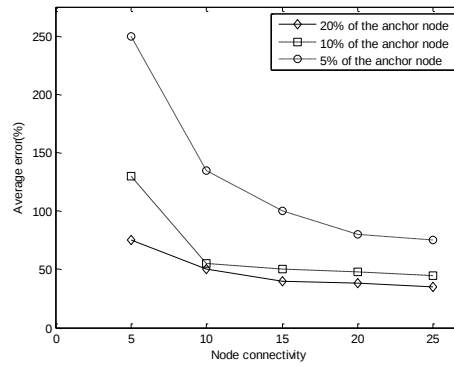


Figure 7. Average Error of Nodes Indistributed Algorithm (ranging error 5%)

Figure 7 shows in the distance measurement error of 5% perform distributed algorithm node average error, from Figure 7, we can see that when the network connectivity is less than 10, less than 10% anchor node localization error amounted to more than 100%, in extreme cases, when the network is almost no anchor nodes are also connected degree is very low, the algorithm error rate even reached 250%.

(2) Refinement algorithm error experiment

This experiments are in the range error of 5%, using the average error formula (4-1) calculation, select the anchor node ratio is 5%, 10% and 20%. As shown in Figure 8.

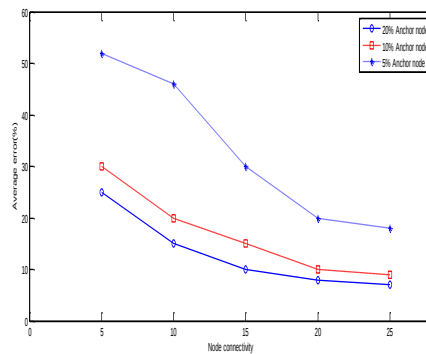


Figure 8. Node Average Error Map after Refinement Algorithm (ranging error 5%)

Figure 8 is the distributed algorithm refinement algorithm by experiment. As shown in Figure 8, curve shape and Figure 7 is similar, but it is evident that error degree decreases greatly, the in connectivity is 5, all three anchor node proportion of error are not more than 55%, and the connectivity degree and the proportion of anchor nodes is large, the node localization accuracy can be reached more than 90%.

The experimental results of analysis show, improved algorithm for sparse anchor node and connectivity degree is low, the performance is not very good, but refinement algorithm can greatly improve the localization accuracy.

(3) Location contrast experiment

The two sets of experiments are in the range of 2.5% error, the anchor node ratio is 5%.

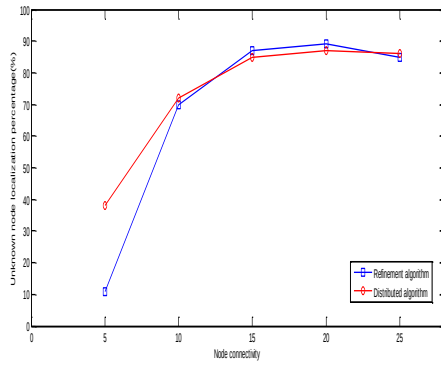


Figure 9. Distributed Algorithms and Refinement Algorithm Location Percentage Map (anchor node 5%, ranging error 2.5%)

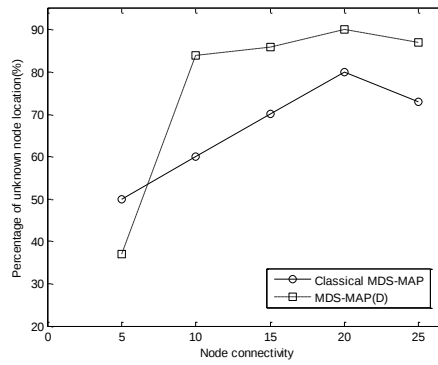


Figure 10. Classic MDS-MAP Algorithm and MDS-MAP (D) Algorithm Location percentage map (anchor node 5%, ranging error 2.5%)

Figure 9 shows the percentage of MDS-MAP (D) algorithm in the distributed algorithm and refinement algorithm of unknown node positioning. Experimental results further show that in the low anchor node probability and sparse networks, there will be a large localization error. At the same time, we can see clearly that 5% anchor node, when connected degree is large enough (at least more than 10), positioning of unknown nodes percentage continued in a relatively stable range.

Figure 10 shows the classic algorithm and MDS-MAP (D) algorithm for the location of the unknown node percentage. The experimental results show that the difference between the improved algorithm and the classical algorithm is obvious. In sparse networks, the localization performance of the improved algorithm is not as good as the classical algorithm, but when the node connectivity is greater than 10, the localization rate of the improved algorithm will be significantly higher than the classical algorithm.

6. Conclusion

In this paper, we design and implement a advanced distributed localization algorithm based on MDS-MAP algorithm named AMDS-MAP (D). Experimental results show that AMDS-MAP (d) algorithm in high connectivity degree and a higher proportion of the anchor node, the node localization accuracy higher than classical algorithm, the introduction of the refinement algorithm can further improve the accuracy of the node localization; due to the algorithm will be calculated steps are dispersed in each node, thus reducing the computational complexity of the algorithm, thereby reducing the node battery operation power consumption and prolong the lifetime of the nodes. AMDS-MAP(D) is suitable for the location of complex environment, such as indoor, urban canyon, forest land and so on, and has a relatively broad application prospects.

Acknowledgements

This work was partially supported by National Nature Science Fund of China (NSFC) (Grant No.61403229, 61503213), Quzhou Science and Technology Project(2014Y006) and Quzhou University middle-aged and young backbone teachers funded project.

References

- [1] I. F. Akyildiz, W. Su and Y. Sankarasubramaniam, "A Survey on Sensor Networks", IEEE Communications Magazine, vol. 40, no. 8, (2002), pp. 102-114.
- [2] F. L. Lewis, "Wireless sensor networks", In D. J. Cook and S. K. Das, editors, Smart Environments: Technologies, Protocols, and Applications, New York, (2004), pp. 35-64 .
- [3] Z. Lu, "Research on Distributed Localization Algorithm Based on wireless sensor networks", master thesis. Hubei, College of computer science and technology, Wuhan University of Technology, china, (2008).
- [4] H. Shuangxia, Z. Lu, F. Yiming and C. Jiangfu, "WSN distributed multidimensional scaling localization algorithm of improved", sensing technology, vol. 5, no. 22, (2009), pp. 728-733.
- [5] Z. Lu, H. Shuangxia, F. Wei and Y. Mingxia, "A Distributed MDS-MAP Algorithm for Wireless Sensor Networks", Applied Mechanics and Materials, vol. 52-54, (2011), pp. 1626-1631.
- [6] Y. Shang, J. Meng and H. Shi, "A New Algorithm for Relative Localization in Wireless Sensor Networks", Proceedings of the 18th International Parallel and Distributed Processing Symposium(IPDPS'04), vol. 24, (2004), pp. 26-30.
- [7] L. Yang and X. Jianping, "Three dimensional node localization algorithm in wireless sensor networks with random communication radius", Journal of sensing technology, vol. 1, no. 24, (2011), pp. 88-92.
- [8] W. L. Xiong, T. Mengna and X. Baoguo, "A for wireless sensor network node positioning new method", Journal of sensors and increasing, vol. 4, no. 24, (2011), pp. 576-580.
- [9] Initial y, "Distributed localization algorithm in wireless sensor network research and implementation of [Ph.D. Thesis]", Beijing: Institute of computer science and technology, Beijing University of Posts and telecommunications, (2007).
- [10] L. Huabo, C. Jianming and D. Hongjun, "Wireless sensor networks based on multi classification algorithm for wireless sensor networks", sensing technology, vol. 5, no. 24, (2011), pp. 771-777.
- [11] L. Min, L. Ting and X. Hua, "A wireless sensor network localization algorithm based on reference node selection model", vol. 2, no. 24, (2011), pp. 264-268.
- [12] Z. Shaoping, "Research on node localization algorithm in wireless sensor networks [doctoral dissertation]", Wuhan: Huazhong University of Science and Technology computer software and theory, (2010).

Authors



Zhang Lu, she is in School of Electrical and Information Engineering, Quzhou University, Quzhou, Zhejiang, 324000, China.
E-mail: zhanglu90573@163.com

