Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **5**, September **2018**

A Hybrid Localization Algorithm Based on Clustering for WSNs

Xiao-Gang Qi
School of Mathematics and Statistics
Xidian University
xgqi@xidian.edu.cn

Xiao-Ke Liu
School of Mathematics and Statistics
Xidian University
xkliu_2@stu.xidian.edu.cn

Li-Fang Liu
School of Computer Science and Technology
Xidian University
lfliu@xidian.edu.cn

Received February, 2018; revised May, 2018

Abstract. *This paper presents a hybrid localization algorithm based on clustering for wireless sensor networks (WSNs) called HLBC, which is suitable for dynamic proactive wireless sensor networks. A novel clustering algorithm is used to generate the specified number of cluster nodes in WSNs, considering connectivity and remaining energy of nodes. An original multilateral correction algorithm is introduced to filter and correct them which have large error calculated by MDS-MAP algorithm, after finding the bad node of the network by matching degree function. The running time of HLBC is obviously lower and more stable than that of the centralized MDS-MAP, especially when the number of nodes increases. Results in extensive simulation experiments of square and C-type networks show that the proposed algorithm is more accurate and more efficient than the MDS-MAP algorithm.*
Keywords: Clustering algorithm, Multilateral correction algorithm, Matching degree function, MDS-MAP algorithm, Wireless Sensor Networks.

---

1. **Introduction.** Wireless sensor networks are widely used in military reconnaissance, environmental monitoring, smart home and other fields due to its perceived, computing, and self-adaptive abilities. Interestingly, the location service of WSN is a guarantee of important services such as information collection, target tracking, information management and so on. With the development of WSNs, dynamic sensor networks are getting more and more attention. It is unwise to deploy tens of thousands of sensors in large areas, such as forest, to achieve the monitoring goal. Moving a reasonable number of sensors to monitor the wide region can effectively improve the disadvantages caused by above mentioned, which is generally achieved by attaching them to vehicles or animals.

WSNs can be divided into proactive networks and reactive networks. The sensor nodes in the proactive sensor network sense the detection area in the cycle time and periodically send collected information to the base station. For a dynamic proactive network, we only need to calculate the position of each node periodically rather than momently which means the positioning of it can be regarded as the positioning of a static network at a particular moment in time. However, there exits two positioning error of dynamic proactive WSNs, which are different from static WSNs, namely, the error of algorithm itself, and the error derives from the movement of sensor nodes when the algorithm is running.

Currently, according to the problem whether the distance between nodes needs to be measured, existing positioning algorithms can be classified into the Range-based algorithm and the Range-free algorithm. Range-based algorithms include trilateral algorithm [1], multilateral algorithm [2], DEP algorithm [3], etc. Range-free algorithms are mainly comprised of centroid localization algorithm [4], APIT (Approximate Point-in-Triangulation Test) algorithm [5] and DV-Hop(Distance Vector-Hop) algorithm [6, 7]. The MDS-MAP algorithm [8] can be considered as the Range-based algorithm [9], or the Range-free algorithm [10]. The location accuracy of the Range-based algorithm is much higher than the Range-free algorithm under the same consumption. Nowadays, many researchers prefer to use heuristic algorithms to solve the localization problems of wireless sensor networks, such as Bat Algorithm (BA) [11], Artificial Fish Swarm Algorithm (AFSA) [12], Flower Pollination Algorithm (FPA) [13], Particle Swarm Optimization (PSO) [14] and Multi-Objective Firefly Algorithm (MFA) [15], but the time complexity of these is relatively high. For example, the running time of the algorithm described in reference [16] is up to 500 seconds. The characteristics of error of dynamic proactive wireless sensor networks require that the running time of the algorithm must be reduced as much as possible, in order to cut down the positioning error. Therefore, heuristic algorithms do not apply to this type of network.

Since the multilateral algorithm and MDS-MAP algorithm have advantages in positioning accuracy, this paper proposes a hybrid localization algorithm based on clustering. Considering the input matrix of centralized MDS-MAP algorithm has a high time complexity which is caused by performing $n(n+1)/2$ single-source shortest path algorithms, we introduced the idea of dividing the wireless sensor network into several clusters to reduce the time complexity of it. Besides, this approach can decrease the error on the irregular area such as C-type region, compared with the centralized MDS-MAP algorithm. when the sensor network is attacked, there is a great possibility that some nodes will be invalid and the information in the link will be maliciously altered. In this case, the estimated positions of all nodes in WSNs obtained by the centralized MDA-MAP algorithm are no longer reliable. But for the clustering algorithm, only clusters, where attacked nodes or links are located in, are affected. Shang Y proposes an improved MDS-MAP algorithm [17]. The algorithm has to cluster and execute the shortest path algorithm for each node such that it has high computational complexity and node energy consumption. Tian H. L proposes MDS-MAP (EP) algorithm [18] which can balance the energy consumption of nodes, regardless of the position of cluster heads. Both of these algorithms have a common disadvantage: clusters need to be merged by the fusion algorithm after running the MDS-MAP algorithm in each cluster. The premise of fusion is that there are more or less common nodes between clusters. Of course, the more public nodes, the more accurate the fusion will be. But this process can not only reduce the lifetime of the common node but also increase the time cost because they need to send or receive messages to multiple cluster heads.

This paper presents a hybrid localization algorithm based on clustering (HLBC) inspired by reference [19, 20, 21, 22]. HLBC algorithm consists of cluster head selection algorithm and multilateral correction algorithm. In the cluster head selection algorithm, a specified number of cluster heads can be selected by an interest function combining the node connectivity and the residual energy. The multilateral correction algorithm can filter and correct bad nodes which have large error calculated by MDS-MAP algorithm.

2. Related work.

2.1. Problem description. For proactive wireless sensor networks, dynamic localization is equivalent to static localization in one cycle. So, we consider a static WSN with n wireless nodes labeled $1, 2, ..., n$ in 2-dimensional space. The number of anchor nodes whose locations are known already is $m(m < n)$, so there are $n - m$ unknown nodes should be localized in this problem. By using RSSI signal propagation model, we can estimate the distance from one node to its neighbors.

2.2. Error in localization problem. In this paper, the positioning error consists of two parts, one is the error of the algorithm itself and the other is the error caused by the algorithm running time, denoted as Err_A and Err_T respectively.

Definition 2.1. *Total Localization Error,*

$$Total\_Err = Err_A + Err_T \tag{1}$$

$$Err_A = \sum_{i=1}^{n-m} \sqrt{(x_i - \tilde{x}_i)^2 + (y_i - \tilde{y}_i)^2} \Big/ n - m \tag{2}$$

$$Err_T = time * v \tag{3}$$

where $X = \{x_i\}_{i=1}^{n-m}, x_i = [x_i, y_i] \in \mathbb{R}$ represents the real coordinates of unknown nodes, and $\tilde{X} = \{\tilde{x}_i\}_{i=1}^{n-m}$ represents the estimated coordinates. $time$ is the execution time of the positioning algorithm. The parameter v refers to the average moving speed of sensor nodes.

2.3. RSSI signal propagation model. An important feature of wireless signal transmission is that the strength of signal attenuates with the increase of distance. The most widely used simulation model to generate RSSI samples as a function of distance in Radio Frequency (RF) channels is the log-normal shadowing model. A detailed description of the model is given in reference [23], which not be repeated here.

2.4. Some rational assumptions.

- All nodes in the WSNs have the same computing power and storage capacity, and have a unique ID identifier;
- The communication range of nodes can be dynamically changed by changing the size of the wireless transmission power;
- The energy, required for radio signals to transmit information, is the same in all directions.

3. Clustering algorithm.

3.1. Cluster head selection phase. Suppose n nodes are randomly distributed in a square region with area S, and the optimal percentage of cluster head nodes is P. Thus the entire sensor network will be divided into $n * P$ clusters. In other words, we need to select $n * P$ cluster heads in it. In order to achieve this target, we take into account the number of neighbors $N(i)$ and the remaining energy $E(i)$. Details about the calculation method of $E(i)$ see reference [18]. We construct an interest function $W(i)$ to measure whether node i should be selected as the cluster head. The details of the function are as follows:

$$W(i) = \alpha \frac{N(i)}{N_{\max}} + \beta \frac{E(i)}{E_0} \tag{4}$$

The number of neighbor nodes and the remaining energy of node i are $N(i)$ and $E(i)$ respectively. N_{max} represents the maximum number of neighbor nodes in all nodes, and E_0 represents the initial energy of the node. Where α and β are weight coefficients and satisfy the relation $\alpha+\beta=1$. In this article, we set $\alpha=\beta= 0.5$. In addition, we must control the distance between cluster head nodes so that the cluster head nodes are evenly distributed in the WSNs. Approximately, we assume that $n * P$ clusters just divide the square area with area S into $n * P$ equal region, so the shortest path distance between the selected cluster head nodes must be not less than D.

$$D = \sqrt{\frac{S}{n * p * \pi}} \tag{5}$$

Let $W= \{W_1, W_2, W_3, \ldots, W_n\}$ denote the set of function values sorted by node sequence $\mathrm{I} = \{1, 2, 3, \ldots, n\}$. We sort W in descending order to get $W'= \{W'_1, W'_2, W'_3, \ldots, W'_n\}$, and the node sequence becomes $\mathrm{I}' = \{1', 2', 3', \ldots, n'\}$ correspondingly. Due to large energy consumption, anchor nodes do not participate in the selection of cluster head nodes. Thus, after removing the function value of the anchor nodes in W', W' changes to $W'= \{W'_1, W'_2, W'_3, \ldots, W'_{n-m}\}$ and I' changes to $\mathrm{I}' = \left\{1', 2', 3', \ldots, (n-m)'\right\}$.

1) Select the first element $1'$ in set $\mathrm{I}' = \left\{1', 2', 3', \ldots, (n-m)'\right\}$ as the starting cluster head node, and the set of cluster head nodes is $CH = \{1'\}$.
2) Starting from the second element in the set, we compute the shortest path distance vector d_i between this node (assumed to be the ith element in I') and all cluster nodes in CH. When satisfied $d_i \geq D$, node i is elected as cluster head and added to cluster head node set. $CH = CH \cup \{i'\}$.
3) Judge the number of elements in CH, the algorithm doesn't end up until the number is equal to $n * P$.

3.2. **Node into cluster stage.** After selecting all cluster head nodes in the network, cluster head nodes broadcast messages to other nodes that themselves was elected as cluster head. Receiving this message, non-cluster head nodes decide which cluster to join. The clustering process is divided into two steps, the anchor node clustering and the ordinary node clustering. For the anchor node clustering, the first cluster head node finds two anchor nodes which are closest to it and add them to its cluster. The second cluster head seeks out two anchor nodes from remaining anchor nodes in the same manner. In that way, each cluster ought to include two anchor nodes. Thus, at least $2 * n * P$ anchor nodes are deployed in the WSNs. If the number of anchor nodes is more than $2 * n * P$, the remaining anchor nodes calculate the distance to each cluster head node and join the nearest cluster. For the ordinary node clustering, each ordinary node calculates the shortest path distance between cluster heads, then joins the nearest cluster.

4. **Multilateral correction algorithm.** After running classical MDS-MAP algorithm which is detailed describe in reference [24], some nodes have a great error due to uneven distribution of nodes in the WSNs. Due to the existence of these nodes with great error, the average positioning error of the whole network is increased. A sensible idea is that these nodes should be identified and calibrated to improve positioning accuracy. The multilateral correction algorithm involves the following two procedures. Firstly, filtration

Algorithm 1 Cluster head Selection algorithm

Input: n,m,S,P,α,β;
Output: CH;

Calculate D according to Eq.(5)
for $I = 1$ to n **do**
  Calculate $W(i)$ according to Eq.(4)
  $w(i,1) = W(i)$,$w(i,2) = i$
end for
$w' = sortrows(w(m+1 : end, :), -1)$
%Descending order the elements in which not include anchor nodes.
for $j = 1$ to $size(w,1)$ **do**
  if $size(CH,1) == 0$ **then**
    $CH = [CH, \mathrm{w}'(j,2)]$
  else
    if $d_j \geq D$ **then**
      $CH = [CH, \mathrm{w}'(j,2)]$
    end if
    if $size(CH,2) == round(n * P)$ **then**
      break;
    end if
  end if
end for

Algorithm 2 Cluster head Selection algorithm

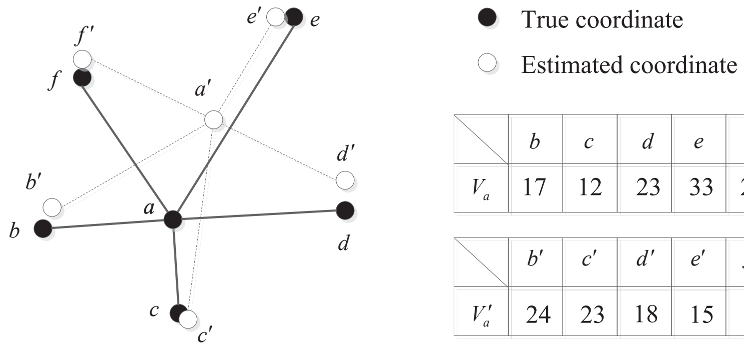
Input: CH,n,m,S,P,$Anchors$;%Anchors is the ID vector of anchor nodes.
Output: CH_Anchor;

for $I = 1$ to $size(CH,1)$ **do**
  $anchors = setdiff(Anchors, CH\_1)$%Anchors\CH_1,CH_1 is initialized to empty.
  for $j = 1$ to $size(Anchors,1)$ **do**
    $d(i,j) = Floyd(i, Anchors(j))$
  end for
  $CH\_1 = SelectDist(d(i,:))$
  % Seeks out two anchor nodes closest to node i to join the cluster of i.
  $CH\_2(1:2,i) = CH\_1$
end for
for $k = 1$ to $m - 2 * n * P$ **do**
  $CH\_3 = Getthenearestclusterheadnode$
end for
Combine CH_2 and CH_3 to get the final result CH_Anchor.

must identify a certain percentage of *bad nodes* with low localization accuracy. Secondly, calibration must utilize *good nodes* with high localization accuracy to correct the coordinate of *bad nodes*. To avoid confusion, hereafter we use "estimated coordinates" to denote the node coordinates before filtration.

4.1. **Filtration algorithm.** According to the log-normal shadowing model, the value of RSSI between neighbor nodes decrease as the distance increases. Let V_{a_1} denote the distance vector which is calculated by the log-normal shadowing model between a and its neighbor nodes. Let V'_{a_1} denote the distance vector calculated by estimated coordinate between a and its neighbor nodes. Obviously, the more similar V_{a_1} and V'_{a_1}, the more accurate the estimated coordinates of a. Moreover, with a large number of simulations, we found that classical MDS algorithms may lead to the overall migration of several nodes. Based on those observations and analyses, we propose a filtration algorithm.

First, we can easily get V_{a_1} and V'_{a_1} by calculation, as illustrated in Fig.1.



	b	c	d	e	f
V_a	17	12	23	33	22

	b'	c'	d'	e'	f'
V'_a	24	23	18	15	19

FIGURE 1. Neighborhood node matching

In an environment without noises, V_{a_1} and V'_{a_1} should be extremely similar. If there is significant mismatch between them, it indicates a large error in the node's estimated coordinate.

Second, an interesting question here is how to measure the match degree between V_{a_1} and V'_{a_1}. In this paper, we solved this question with the help of Manhattan distance. The Manhattan distance is the distance between two points measured along axes at right angles. There are distinct advantages from a computational perspective as the Manhattan distance costs less resources.

Definition 4.1. *Given two vectors $X = (X_1, X_2, \ldots, X_N)$ and $Y = (Y_1, Y_2, \ldots, Y_N)$ of the same dimension, the Manhattan distance for two vectors is:*

$$MD = \sum_{k=1}^{N} |X_k - Y_k| \tag{6}$$

By the definition of the Manhattan distance, the Manhattan distance value MD_1 between V_{a_1} and V'_{a_1} is $|17-24| + |12-23| + |23-18| + |33-15| + |22-19|$, namely, it is 44.

It is not enough that only consider the Manhattan distance of the vector in which elements are neighbor nodes, because the MDS-MAP algorithm may cause the overall migration of several nodes. Thus, we get V_{a_2} which denotes distance vector between and its 2 hop nodes calculated by Floyd algorithm. We get V'_{a_2} which denotes the distance vector calculated by estimated coordinate between a and its 2 hop nodes. Analogously, we can get V_{a_3} and V'_{a_3}. Furthermore, we can obtain MD_2 and MD_3.

Definition 4.2. *We define the matching degree M_a of node a as follows:*

$$M_a = \gamma * \left(1 - \frac{MD\_1}{N_1 * Ad\left(V_{a\_1}\right)}\right) + \eta * \left(1 - \frac{MD\_2}{N_2 * Ad\left(V_{a\_2}\right)}\right) + \sigma * \left(1 - \frac{MD\_3}{N_3 * Ad\left(V_{a\_3}\right)}\right) \tag{7}$$

$$Ad\left(X\right) = \frac{\sum\limits_{K=1}^{N} X_k}{N} \tag{8}$$

Where γ,η and σ are weight coefficients and satisfy the relation $\gamma+\eta+\sigma=1$. In this paper, we set $\gamma=0.4, \eta = 0.3, \sigma = 0.3$. In theory, the greater the value of M_a, the more accurate the estimate coordinate of node a is. We do some experiments to verify the relationship between *matching degree* and positioning error. The results are plotted in Fig.2. It is worth noting that, as the topology of the network changes due to the movement of nodes, the range of matching degrees will fluctuate. In other words, *bad nodes* can't be filtered out well by setting a fixed threshold of *matching degree*. Our strategy is to set the proportion of bad nodes to P_b. Generally, we set P_b=20% $\sim$ 40%. If P_b is taken too large means that the number of *good nodes* for calibration is too small, which affects the accuracy of the calibration. Let $M = \{M_1, M_2, M_3, \ldots, M_n\}$ denote the matching degree vector sorted by node sequence $\mathrm{I} = \{1, 2, 3, \ldots, n\}$. We sort M in ascending order to get $M' = \{M'_1, M'_2, M'_3, \ldots, M'_n\}$, and the label sequence of the corresponding node becomes $\mathrm{I}' = \{1', 2', 3', \ldots, n'\}$. The first $n * P_b$ elements of M are *bad nodes*, and the remaining nodes are *good nodes*.

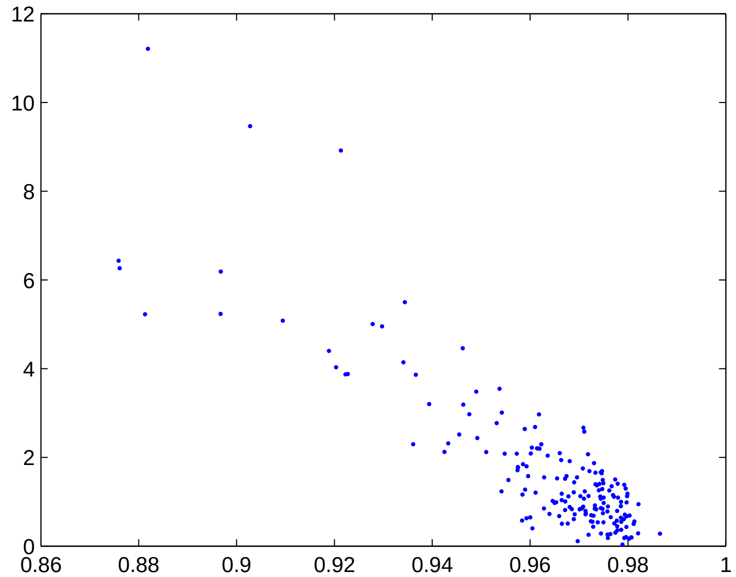


FIGURE 2. The relationship between matching degree and estimated location error

4.2. **Correction stage.** After filtering out *bad nodes* in WSNs, we regard *good nodes* as new anchor nodes. At the moment, the number of anchor nodes in WSNs is equal to the number of original anchor nodes plus the number of *good nodes*. Then, we recalculate the coordinates of bad nodes according to the multilateral algorithm [2].

5. **HLBC.** In this section, we summarize the main steps of the hybrid localization algorithm based on clustering (HLBC).

1) Use the Cluster head Selection Algorithm to find all cluster heads in WSNs.
2) According to the clustering rules of nodes mentioned in Section 3.2, all nodes are clustered.
3) Calculate the location of cluster head nodes by the multilateral algorithm. Cluster head nodes which obtain location information become new anchor nodes.
4) Each cluster runs MDS-MAP algorithm to calculate the coordinate of nodes in it.

5) Multilateral correction algorithm is used to calibrate the coordinates of *bad nodes* in each cluster.

6. Simulation and result analysis.

6.1. The result of clustering algorithm. In order to verify the performance of the clustering algorithm, we perform this algorithm in the square area and the C-type area respectively. In experiments, 200 nodes are placed randomly in WSNs. Let $P = 3\%$, which means we need to select 6 cluster head nodes. The simulation results are shown in Fig.3 and Fig.4. From those figures we can draw conclusions that the performance of the algorithm is excellent and cluster head nodes are evenly distributed in the network.

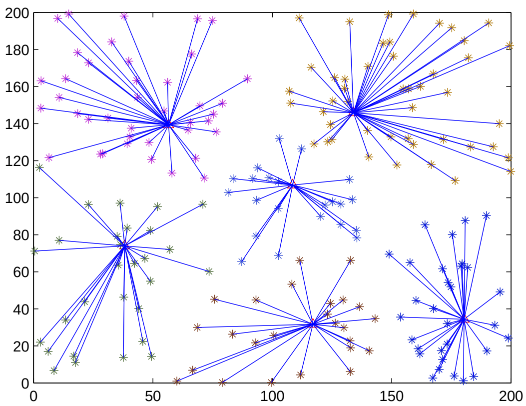


FIGURE 3. Clustering results in square area

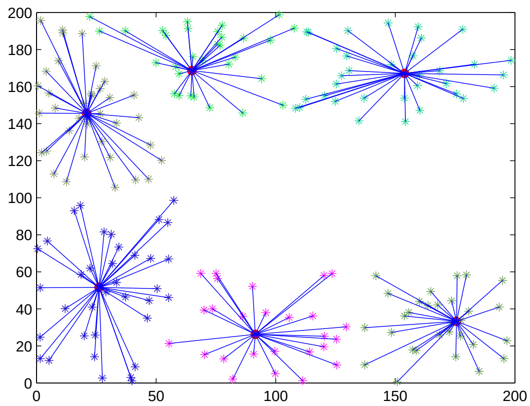


FIGURE 4. Clustering results in C-type area

6.2. The result of multilateral correction algorithm. Due to random movement of nodes in a dynamic proactive wireless sensor network, it is meaningless to assume that nodes are evenly deployed. We compared the location performance before and after applying the correction algorithm in the same simulating configuration. The experimental parameters are shown in Table 1, and the experimental results are shown in Fig.5. Fig.6 shows the variation of the positioning error caused by the changes of the two algorithms with the communication radius.Through the above simulation experiments, we are able to conclude that the correction algorithm can effectively calibrate some bad nodes in WSNs, and improve the positioning accuracy.

TABLE 1. Parameters used for square area

Variable	Value
Deployment method	random
Map size	$200m \times 200m$
Sensor nodes	200
Anchor nodes	20
Radio range	$30 \sim 50m$

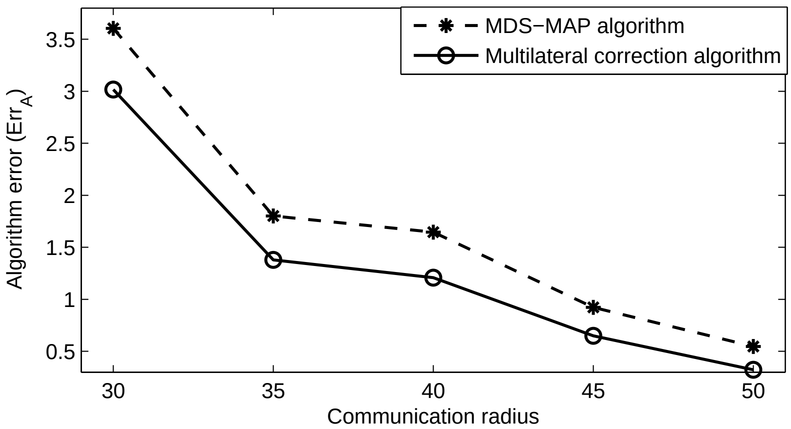


FIGURE 5. Relationship between communication radius and localization error(Err_A)

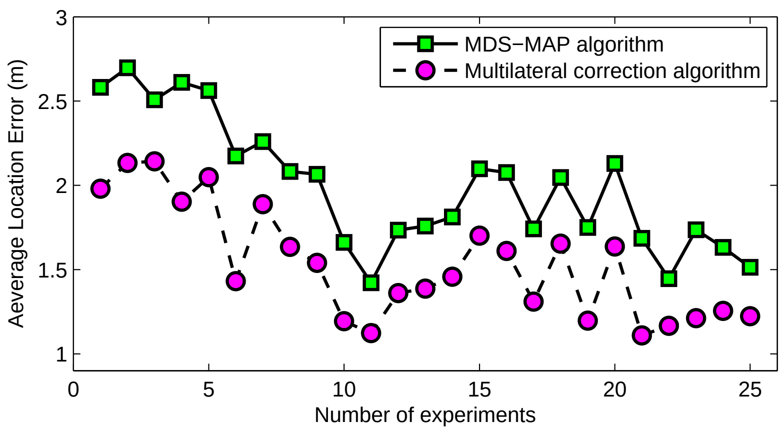


FIGURE 6. Comparison of localization errors(Err_A)

6.3. **The result of HLBC.** As described in Section 2.2, the error in a dynamic proactive wireless sensor network derived from two parts, Err_A and Err_T. First of all, we carried out 30 simulation experiments to compare the Err_A value of the MDS-MAP algorithm with the HLBC algorithm. The results are plotted in Fig.6. Overall, the localization accuracy of the HLBC algorithm is better than that of the MDS-MAP algorithm. Note that the value of the error Err_A of the HLBC algorithm is slightly larger than that of the MDS-MAP algorithm in the 25th experiment. The reason is caused by the fact that the node with larger connectivity consumes more energy and the cluster head selection algorithm tends to choose the node with more remaining energy as the elected cluster head as time goes by. In this case, the value of $Total\_Err$ in the HLBC algorithm is larger than that of the MDS-MAP algorithm absolutely, and the lifetime of the WSNs becomes longer. Secondly, we implemented 30 simulation experiments to compare the $Total\_Err$ value of the MDS-MAP algorithm with the HLBC algorithm. We set the node's average moving speed v $1.4m/s$ references to the human walking speed. The result is shown in Fig.7. Furthermore, the running time of these two algorithms is compared. Fig.8 shows the variation in runtime of those two algorithms as the number of nodes in the network changes. There are two facts: the running time of the MDS-MAP algorithm is obviously higher than that of the HLBC algorithm, and the former gradually increases whereas the later remains stable as the number of nodes increases.

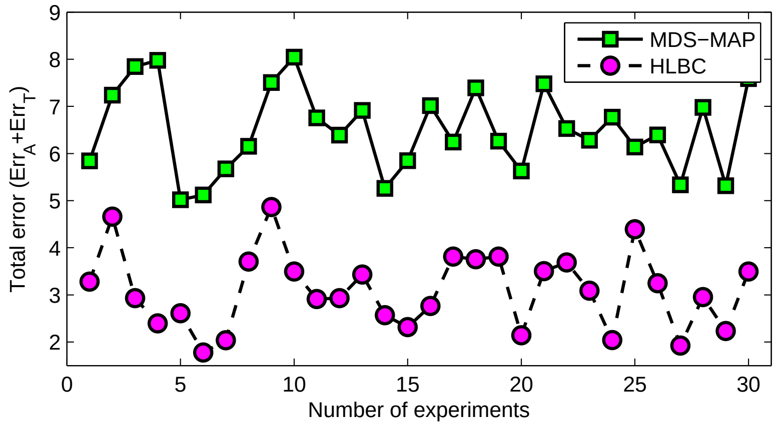


FIGURE 7. Comparison of localization errors (Err_A) in square area

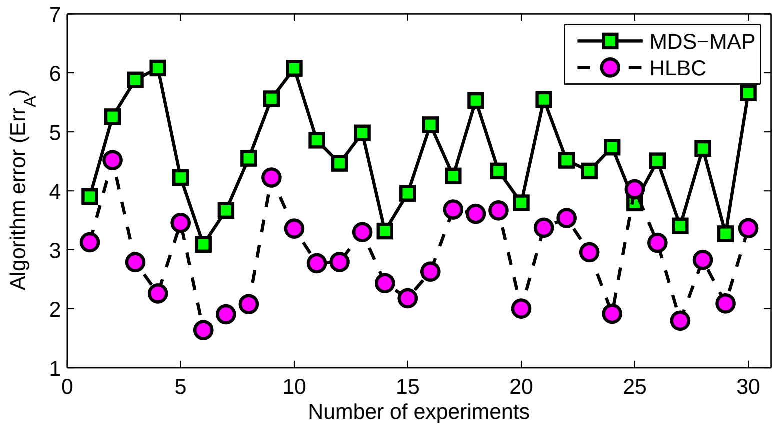


FIGURE 8. Comparison of Total localization errors ($Total\_Err$) in square area

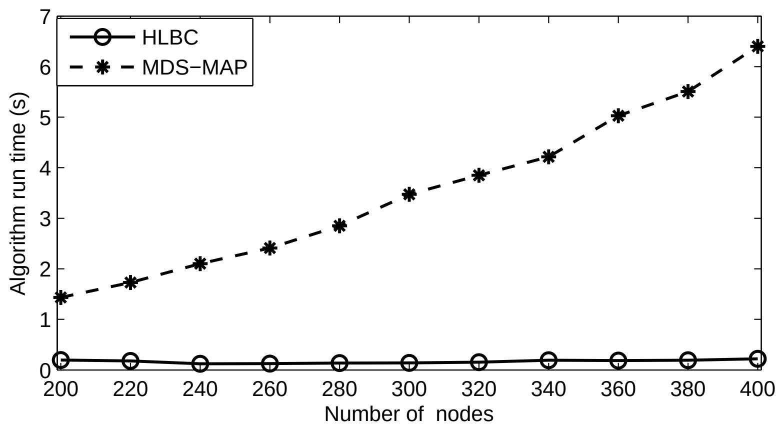


FIGURE 9. Comparison of run time

Nodes movement is bound to change the shape of the area in which nodes located, and this change is unpredictable. To further demonstrate the performance of the HLBC algorithm, we conducted experiments on this algorithm in a typical irregular region (C-type region). Fig.9 shows the simulation results. Compared with the MDS-MAP algorithm, the

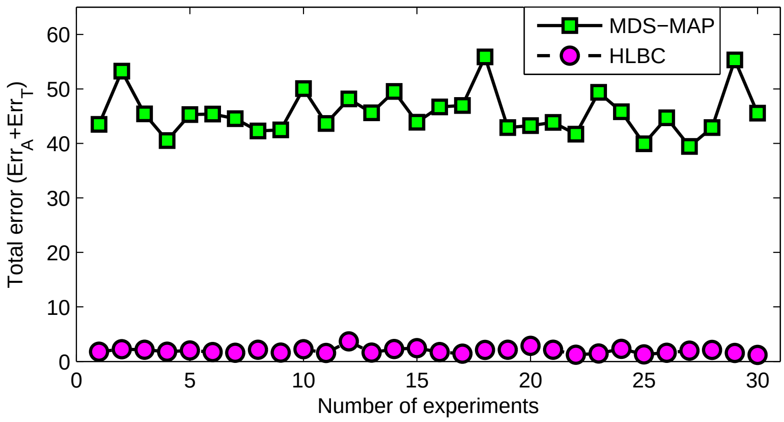


FIGURE 10. Comparison of Total localization errors ($Total\_Err$)in C-type area

HLBC algorithm has overwhelming superiority. The reason is that the HLBC algorithm successfully avoids the influence of the "hole" in WSNs by clustering.

7. **Conclusions.** In this paper, we propose a hybrid localization algorithm based on clustering for WSNs. This algorithm, whether in the square area or C-type area, can not only improve the positioning accuracy, but also increase the lifetime of wireless sensor network. In addition, the running time of this algorithm has a stable property which means it does not grow with the number of nodes increases.

Acknowledgment. This Project supported by the National Natural Science Foundation of China (Grants Nos. 61572435, 61877067), Joint Foundation High-Tech LSNSET (No. KX172600039), Ningbo Natural Science Foundation (Grant Nos. 2016A610035, 2017A610119).

REFERENCES

[1] W. T. Lan, E. Q. Zhang and J. Y. Luo, Design and Implementation of Adaptive Intelligent Trilateral Localization Algorithm,*Chinese journal of sensors and actuators*, vol. 30, no. 7, pp. 1089-1094, 2017.

[2] R. X. Gong, J. W. Zhao, and X. B. Meng, WSN Multilateral Localization Algorithm Based on MLE, *International Conference on Network Computing and Information Security*, pp.426-429, 2011.

[3] H. A. B. De Oliveira, E. F. Nakamura, A. A. F. Loureiro and A. Boukerche, Directed Position Estimation:A recursive localization approach for wireless sensor networks, *INFORMATICAProceedings of the 7th International Conference on Computer Communication and Networks*, pp.557-562, 2005.

[4] Y. Bai, C. M. Li, and Y. Xue, A Centroid Localization Algorithm for Wireless Sensor Networks based on RSSI, *Applied Mechanics and Materials*, vol.303-306, no.7, pp.197–200, 2013.

[5] H. B. Tao and F. Liu, Research and Implementation of APIT Positioning Algorithm in WSN, *9th International Conference on Fuzzy Systems and Knowledge Discovery*, pp.2212-2215, 2012.

[6] D. Niculescu and B. Nath, DV based positioning in ad hoc networks, *Telecommunication Systems*,vol.22, no.1, pp.267-280, 2017.

[7] J. Mass-Sanchez, E. Ruiz-Ibarra, J. Cortez-Gonzlez, Weighted Hyperbolic DV-Hop Positioning Node Localization Algorithm in WSNs, *Wireless Personal Communications*, pp.1-23, 2016.

[8] A. Karbasi and S. Oh, Robust localization from incomplete local information, *IEEE/ACM Transactions on Networking*, vol.21, no, 4, pp.1131-1144, 2013.

[9] F. K. W. Chan and H. C. So, Efficient Weighted Multidimensional Scaling for Wireless Sensor Network Localization, *IEEE Transactions on Signal Processing*, vol. 57, no.11, PP, 4548-4553, 2009.

[10] X. Zhang, Y. Wu and X. Wei, Localization algorithms in wireless sensor networks using nonmetric multidimensional scaling with RSSI for precision agriculture, *International Conference on Computer and Automation Engineering*, pp.556-559, 2010.

[11] D. Simian, Range Based Wireless Sensor Node Localization Using Bat Algorithm, *ACM Symposium on PERFORMANCE Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, pp.41-44, 2016.

[12] X. Y. Yang, W. Zhang, and Q. Song, A Novel WSNs Localization Algorithm Based on Artificial Fish Swarm Algorithm, *International Journal of Online Engineering*, pp.64-68, 2016.

[13] J. S. Pan, T. K. Dao, T. S. Pan, T. T. Nguyen and S. C. Chu, An improvement of flower pollination algorithm for node localization optimization in WSN, *Journal of Information Hiding & Multimedia Signal Processing*vol.8, no.2, pp.486-499, 2017.

[14] I. F. M. Zain, S. Y. Shin, Distributed Localization for Wireless Sensor Networks Using Binary Particle Swarm Optimization (BPSO), *Vehicular Technology Conference*, pp.1-5, 2015.

[15] T. T. Nguyen, J. S. Pan, S. C. Chu, J. F. Roddick, and T. K. Dao, Optimization Localization in Wireless Sensor Network Based on Multi-Objective Firefly Algorithm, *Journal of Network Intelligence* , vol. 1, no.4, pp.130-138, 2016.

[16] M. V. Ramesh, P. L. Divya and R. V. Kulkarni, A swarm intelligence based distributed localization technique for wireless sensor network, *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, pp.367-373, 2012.

[17] Y. Shang and W. Rum, Improved MDS-based localization, *Joint Conference of the IEEE Computer and Communications Societies*, vol.4, pp.2640-2651, 2004.

[18] H. L. Tian, Z. H. Qian, Y. J. Wang and X. Liang, Modified MDS-MAP Localization Algorithm with Distsnce Error Correction in Energy Clustering Wireless Sensor Networks, *Journal of Electronics & Information Technology*, vol.39, no.7, 2017.

[19] P. Chen, H. Ma and S. Gao, SSL: Signal Similarity-Based Localization for Ocean Sensor Networks, *Sensors*, vol.15, no.11, pp.29702-29720, 2015.

[20] J. Zhao, W. Xi, Y. He, Localization of Wireless Sensor Networks in the Wild: Pursuit of Ranging Quality, *IEEE/ACM Transactions on Networking*, vol.21, no.1, pp.311-323, 2013.

[21] T. T. Nguyen, T. K. Dao, M. F. Horng, and C. S. Shieh, An Energy-based Cluster Head Selection Algorithm to Support Long-lifetime in Wireless Sensor Networks, *Journal of Network Intelligence* , vol. 1, no.1, pp.23-37, 2016.

[22] J. S. Pan, T. T. Nguyen, T. K. Dao, T. S. Pan and S. C. Chu, Clustering Formation in Wireless Sensor Networks: A Survey, *Journal of Network Intelligence* , vol. 2, no.4, pp.287-309, 2017.

[23] W. Lan, W. Zhang and J. Luo, Design and Implementation of Adaptive Intelligent Trilateral Localization Algorithm, *Chinese Journal of Sensors & Actuators*, vol. 30, no.7, pp.1089-1094, 2017.

[24] Y. Shang, W. Ruml, Y. Zhan and M. P. J. Fromherz, Localization from mere connectivity, *ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pp. 201-212, 2003.