IJMTST is UGC Approved Journal with Journal Id: 43137

International Journal for Modern Trends in Science and Technology, 8(S05):95-100, 2022

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST08S0516
Available online at: https://www.ijmtst.com/vol8si05.html

Improved Algorithm for Locating Nodes with Minimum Error in Wireless Sensor Networks

M.Saritha Devi | N.V.D.S.S.Praneeth | D.N.S.Pavan Kalyan | D.Rohit Venkatesh | L.Kiran Kumar

Department of Electronics and Communication Engineering, Godavari Institute of Engineering and Technology(A), JNTUK, Kakinada.

To Cite this Article
M.Saritha Devi, N.V.D.S.S.Praneeth, D.N.S.Pavan Kalyan, D.Rohit Venkatesh and L.Kiran Kumar. Improved Algorithm for Locating Nodes with Minimum Error in Wireless Sensor Networks. International Journal for Modern Trends in Science and Technology 2022, 8(S05), pp. 95-100. https://doi.org/10.46501/IJMTST08S0516

Article Info
Received: 26 April 2022; Accepted: 24 May 2022; Published: 30 May 2022.

ABSTRACT

Recent advances in developing sensor nodes have made the use of Wireless Sensor Networks (WSN) possible in many different scenarios, most of them require each node to have some information about its geographical location. There are many other optimization techniques which are left with the drawbacks of mean localization error and consumption of higher computational time. This project presents a two-step distance-based algorithm to the sensor network localization carried out in a centralized architecture. The first phase of the algorithm utilizes an improved version of the DV Hop distance algorithm to provide coarse position estimates for all the nodes. During the second phase, Particle Swarm Optimization (PSO) is performed to further refine the output of the first phase. In the second phase, several techniques are also used to address the problems of localization such as flip ambiguity, collective translation and also overcomes the drawbacks of the other algorithms. This proposed method uses less computational time than the existing methods. MATLAB tool is used for analyzing the algorithm.

KEYWORDS: Wireless Sensor Networks (WSN), Optimization, Localization, DV Hop distance algorithm, Particle Swarm Optimization (PSO)

1.INTRODUCTION

The Wireless Sensor Networks generally consists of number of sensor nodes which can sense, compute and communicate with each other through a wireless link. The Wireless Sensor Network domain is applicable in many inspecting fields and offers wide range of applications like in environmental monitoring, disaster management, military applications and others. Simply gathering information from the sensors would not be useful without knowing the exact geographical location of the sensor node from which the data is sent. For example, a network is established to detect the forest fires in a forest [1]. Whenever the fire is detected, a sensor detects the abnormal increase in temperature and sends an alarm message to the base station to inform about the fire by passing through other nodes in a multi-hop manner. In this case, the message sent must include both the data and the location information of that particular node. The Localization techniques can provide an infrastructure to perform the algorithms like location-aware routing [2]. Therefore, self-localization capability has become one of the most important requirements in the self-organized networks.

The simplest way for localization is to go for Global Positioning System (GPS) adapter equipped with each node. But, the use of GPS technology is an expensive way not only in cost but also in size and power consumption. In practice, we consider only small number of anchor nodes whose locations are known by either GPS or some other means and then we try to find out the locations of the remaining nodes. Based on hardware installed, localization algorithms are classified into two categories [3]. The first is Range-free localization algorithms which uses the proximity information such as energy consumption information, area information and others to derive the node locations. The other is Range-based localization algorithms which uses the information such as time of arrival (ToA), received signal strength (RSS) and others to estimate the node locations.

The limitations of Wireless Sensor Networks fueled the necessity for searching simple yet low-cost techniques to derive location information of the unknown nodes. The gradient search methods measure distance between nodes and use them to modify the estimated coordinates in the manner to match with the distance coordinates. These techniques probably may result in the solution which gets trapped in local minima. In contrast to these techniques, the stochastic optimization techniques have the ability to come out of the local minima problem. Thus, these can be used in the accurate location estimation of sensor nodes by optimizing the solutions. Concentrating on these stochastic optimizations, this paper proposes a two-phase algorithm based on the Particle Swarm Optimization (PSO) and the concept of node selection priority. Several fundamental problems in the distance-based algorithms such as flip ambiguity and error propagation are controlled further using the proposed algorithm. This paper is further divided into following sections: Section II reviews the existing method. The concept of PSO is discussed in section III. The details of proposed algorithm are in section IV. Section V deals with the results and section VI concludes the topic.

2. EXISTING METHOD

The DV-Hop algorithm is a range free algorithm that uses hop distances in its calculations. The process of transmitting the data signal from the sender node to the receiver node is called a Hop. This algorithm, as in [4], comprises of three different phases. The result of the first phase comprises of assigning minimum hop count values to all the anchor nodes. In the second phase, the average single hop distance is calculated to convert hop count value into the physical distance. For the reference node (x_i, y_i), the average single hop distance is given by

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

Where (x_j, y_j) is the location of the node j (j≠i), and h_j is the hop count value from node j to node i. The $HopSize_i$ estimated gets broadcasted throughout the network. Each receiving node keeps hold of the minimum hop-count value per anchor node.

In the third phase of the algorithm, with the distance estimations to at least three reference nodes, the unknown node locations are estimated using the multilateration method. Distance of the unknown nodes to the anchor nodes is given by the formula

$$d_{ij} = HopSize * h_j$$

Assume beacons a, b, c with the locations (x_a, y_a), (x_b, y_b) and (x_c, y_c) and d_a, d_b, d_c are distances to an unknown node D with coordinates (x, y).

$$d_a = \sqrt{(x - x_a)^2 + (y - y_a)^2}$$
$$d_b = \sqrt{(x - x_b)^2 + (y - y_b)^2}$$
$$d_c = \sqrt{(x - x_c)^2 + (y - y_c)^2}$$

The coordinates of D can be calculated and the error is given by $d_{calculated} - d_{real}$. In this algorithm, the maximum value of the hop sizes received by the anchor nodes in the second phase is the hop-size which will be considered in the third phase of the algorithm. The average one-hop distance among anchors and the average one-hop distance used by each unknown node for its location estimation are modified. Then, the distance of each unknown node, d_u is given by:

$$d_u = Max\{hop\text{-}size\}_i * \text{minimum number of hop counts to each node}$$

where, $Max\{hop\text{-}size\}_i$ is the maximum hop distance received by the anchor i.

This algorithm may not always produce unique localization result. When a node's neighbors are nearly collinear, the position estimation can be reflected across the line best fits its neighbors, in addition to satisfying distance coordinates. This phenomenon is called as Flip Ambiguity, it is depicted in the figure 1. Here, node A is situated in the surroundings of almost collinear nodes B, C, D and E. In this type of situations, there is a possibility for the algorithm to estimate the location of node A in the flipped location A' without disturbing the calculated distance estimates. When this phenomenon occurs, the output generated seems to be an accurate one. But the location coordinates may be the flipped version of the original coordinates. Figure 1 shows the clear representation of the flip ambiguity problem in the sensor node localization.

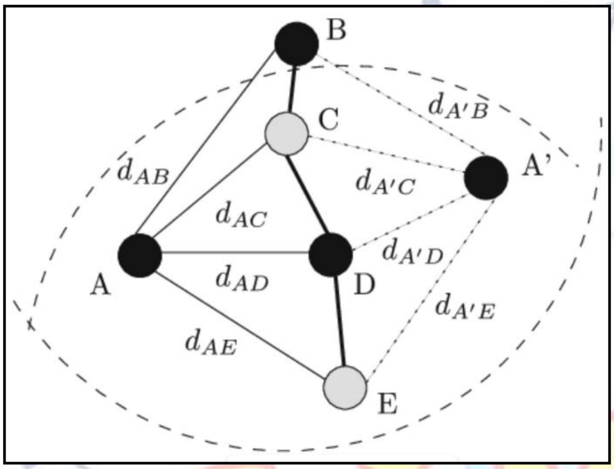

Figure 1: Flip ambiguity problem

3.A REVIEW ON PSO TECHNIQUE

Particle Swarm Optimization (PSO) is a stochastic algorithm developed by Kennedy and Eberhart. It is based on collective intelligence, inspired by the social behavior of bird flocking. Particle Swarm Optimization optimizes the problem by considering a swarm of solutions, called as particles, and moving them systematically around the search-space. The particle movements are dependent on the cognitive behavior of the whole particle population. Each particle keeps track of its best solution coordinates achieved in the search-space, here called as Personal Best (pbest). Other best values that are obtained so far by the particle swarm optimizer are called local best (lbest) and the best solution among all the population values is called Global Best (gbest). Each particle velocity and position in the search-process can be obtained by:

$$
\begin{aligned}
v_{i,j+1} = w.v_{i,j} &+ c_1.rand().(\text{pbest}_i - x_{i,j}) \\
&+ c_2.rand().(\text{lbest}_i - x_{i,j}) \\
&+ c_3.rand().(\text{gbest}_j - x_{i,j}) \\
x_{i,j+1} = v_{i,j} &+ x_{i,j}
\end{aligned}
$$

Where j is index of the iterative steps. $x_{i,j}$and $v_{i,j}$ describe position and velocity of particle i respectively, in step j. *rand()* is a random number distributed uniformly in the interval (0,1). c_1, c_2 and c_3 are positive constants that provide weight of each term. w represents the inertia weight. Depending on the variations in the parameter w, Particle Swarm Optimization is of different variants. In the proposed algorithm, linear inertia reduction strategy is used, in which, the value of w is reduced from 0.9 to 0.4. In order to achieve best solution faster, the effects of the neighboring particles are ignored.

4.PROPOSED METHOD

Like other optimization algorithms, the Particle Swarm Optimization algorithm also needs to be initialized with some starting points in order to begin the search. One general way is to choose random solutions in the search-space. But, assigning random coordinates initially to the sensor nodes is always not a good option as it can produce large error and the optimizer may not produce desired result due to the large number of variables and local minima in the cost function. So, some additional information is needed to be provided before running the optimization phase. Some rough estimation of the nodes can improve the algorithm efficiency. Hence, a slightly modified version of DV-Hop algorithm is used in the first phase of our algorithm to provide initial coarse location estimates and then use them as a starting point in the optimization phase.

FIRST PHASE OF PROPOSED ALGORITHM:

In general, unknown sensor nodes do not lie in the surroundings of three or more anchor nodes in order use basic multilateration, which requires a node to have range estimations to at least three nodes. The usage of the average distances instead of Euclidean distance may create a huge error. So, the first problem is to find the accurate distance estimations from anchor nodes to the unknown nodes. The distance estimates to distant anchor nodes are provided by the DV-distance propagation method [5] by using multi-hop collaboration between the nodes. After gathering the distance information from all the nodes, Dijkstra's algorithm is applied to simulate the propagation task and the shortest-path between arbitrary pair of nodes is calculated. Hence, with the knowledge of anchor nodes coordinates and the shortest-path length, each anchor node calculates vector of correction coefficients as:

$$c_i = \left\{ \underset{\substack{j=i,..,m \\ j\neq i}}{c_j} \;\middle|\; c_j = \frac{\|a_i - a_j\|}{p_{ij}} \right\}$$

Where a_i represents anchor i, p_{ij} represents shortest-path length between anchor i and anchor j. The Euclidean distance between the anchors i and j can be calculated by multiplying the shortest-path length between them with their corresponding correction coefficient. The reason for not using the average distances is to reduce the sensitivity of the distance estimations. In some cases of irregular network topologies, the shortest-path length between anchor and unknown nodes differs from the length of straight line between them while this difference is very small in some other cases. Therefore, considering the average distances can cause the deviation in the final estimation of the positions. The algorithm's resistance against the irregularity can be increased by considering limited number of anchor nodes. On the other hand, if the number of anchor nodes are limited to three or four, it can result in the co-linearity. It is found that the usage of 7 nearest anchor nodes can produce the best results. The multilateration technique is performed after obtaining the distance estimates to all the 7 anchors for each node and the initial coarse estimates are obtained as the result of the first phase.

SECOND PHASE OF PROPOSED ALGORITHM:

The second phase uses the Particle Swarm Optimization (PSO) algorithm as its core. This phase utilizes the initial coarse estimates produced by the first phase as inputs and accurately refines the coordinates of the unknowns. The Error propagation is one of the major problems, where the new estimate of selected node has large error because of the errors present in the position estimates of the selected neighbor nodes. To avoid these error propagations, the concept of node selection priority is used in the algorithm. Here, the optimizer chooses the node which has better neighborhood in order to localize first. A parameter called Reliance Coefficient (RC) is assigned to every node in the network, that reflects how reliable the current position estimate of a node is. The RC values of anchor nodes always equals to 1 and the minimum value of zero is chosen for the unknown nodes at the beginning of the second phase. All the available nodes are divided into two separate sets: LG and UG, which consists of nodes that are permanently localized and nodes that are yet to be localized respectively.

$$LG_{initial} = \{n_1, \dots, n_m\}$$
$$UG_{initial} = \{n_1, \dots, n_n\} - LG_{initial}$$

The average of RC's of all the neighbor nodes for every node in set UG is called as their Selection Probability (SP) value. Based on their SP values, all the nodes in the set UG are sorted. The node with highest SP value, n_B,is then chosen as the most prone node of set UG. After all the optimization process is done for n_B, the RC gets updated with its SP value and this new value is used for the next node selection process.

In a nutshell, the second phase of the algorithm chooses the most appropriate node from the set UG by using the node selection process. Then another two sets of nodes N^1_B and N^2_B are indicated, where N^1_B represents the set of immediate neighbors in addition to $n_B$and N^2_B represents the second neighbor set. The sub-network which contains all the nodes in N^1_B should be isolated from the rest of the network. After isolating common nodes with the set LG, the elements of N^1_B will be simultaneously localized and a vector of their coordinates is then obtained to feed the optimizer. The optimization algorithm should also have the knowledge about how distant the node n_B is from its second neighbors. Finally, the corresponding shortest-path length will be calculated by using Dijkstra's algorithm. The PSO optimizer solves the problem of localization by using:

$$\min_{\substack{(\hat{x}_i,\hat{y}_i) \\ n_i\in N_B^1 \\ n_i\notin LG}} \left\{ CF = w_1.\sum_{\substack{j\in N_B^1 \\ j\neq B}} \left(\hat{d}_{Bj} - \tilde{d}_{Bj}\right)^2 + w_2.\sum_{\substack{i\in N_B^1 \\ i\neq B}} \left(\sum_{\substack{j\in N_B^1 \\ j\in N_i \\ j\neq i}} \left(\hat{d}_{ij} - \tilde{d}_{ij}\right)^2 + \sum_{\substack{j\in N_B^1 \\ j\notin N_i \\ j\neq i}} U\left(R - \hat{d}_{ij}\right).\left(R - \hat{d}_{ij}\right)^2 \right) + w_3.\sum_{k\in N_B^2} \left(\hat{d}_{Bk} - \tilde{d}_{Bk}\right)^2 \right\}$$

$$w_1, w_2 = \begin{cases} 1, & j \in UG \\ 2, & j \in LG \end{cases}, \qquad w_3 = \{0.9, 0.4, 0.1\}$$

Where, w_1, w_2 and w_3 are weighting coefficients and U represents the unit step function. Both w_1 and w_2 are used to magnify the terms that are associated with anchor nodes or the members of LG, that are previously localized nodes. When two non-neighbor members of N^1_B are at a distance more than R from each other, the function U is used to eliminate the extra terms influence. Step by step, the accuracy of the estimations of the algorithm gets improved. Initially, the usage of second neighbors has striking effects on the localization. But later, the approximation of shortest path becomes the error source. To overcome this problem, the effect of second neighbors must be gradually decreased during the optimization. After the predetermined iterations of the Particle Swarm Optimization algorithm, the position coordinates of all the nodes in the set N^1_B gets updated with new estimated values. The node n_B will be moved into LG. Finally, new course begins again with node selection procedure.

5.RESULTS

In order to study the performance of the proposed algorithm using the Particle Swarm Optimization, this paper simulates the algorithm by using the MATLAB as the platform and obtained the results under different sensor node proportion.

In theresults, the blue dots represent the unknown nodes whose location is unknown. Whereas, the blue dots with red diamond shape boundary represents the anchor beacons which acts as the reference nodes to find the location of the unknown nodes.

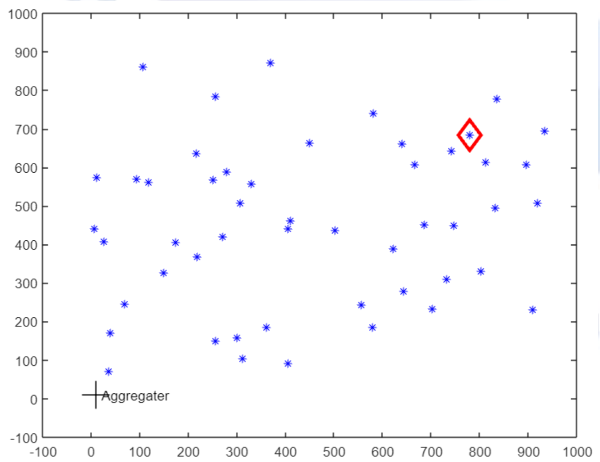

Figure 2: Simulated result when node amount = 50

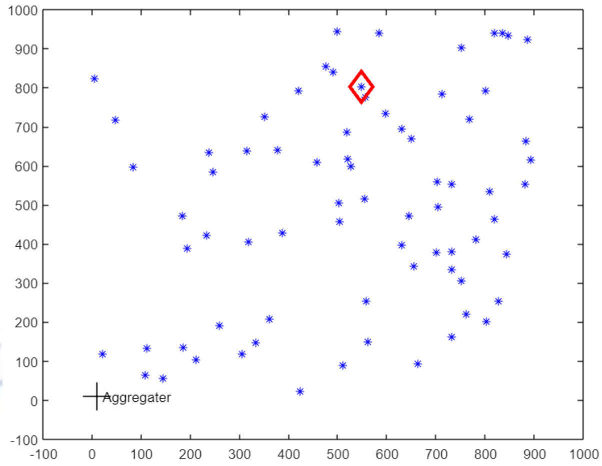

Figure 3: Simulated result when node amount = 75

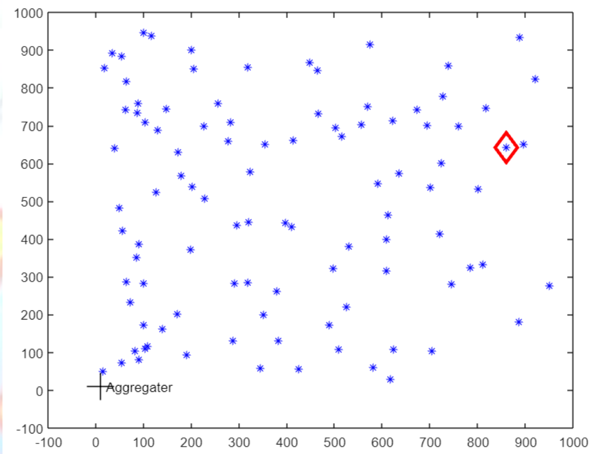

Figure 4: Simulated result when node amount = 100

6. CONCLUSION

In this paper, we discussed about our algorithm which is an application of the Particle Swarm Optimization (PSO) technique. The proposed algorithm is divided into two phases. In the first phase, a modified version of DV-Hop algorithm is used along with the multilateration method to provide the initial coarse location estimates. In the second phase, by using the particle swarm optimizer, we tried to improve the estimation accuracy of the unknown nodes. With the concept of node selection priority, the algorithm became more resistive to the error propagation. It is demonstrated that performing two-hop neighbors in the optimization phase can outflow the flip ambiguity problem especially in the low connectivity rate networks. The results shown that our proposed algorithm outperforms the existing method.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] P. H. Namin, M. A. Tinati, "Node localization using particle swarm optimization", Proceedings of Intelligent Sensors, Sensor Networks and Information Processing, pp. 288-293, 2011.

[2] M. Mauve, J. Widmer, and H. Hartenstein, "A Survey on Position-Based Routing in Mobile Ad-Hoc Networks," IEEE Network, vol. 15, pp. 30-39, Nov. 2001.

[3] G. Han, H. Xu, Q. Trung, J. Jiang, and T. Hara, "Localization Algorithms of the Wireless Sensor Networks: A Survey," Telecommunication Systems, vol. 52, no. 4, pp. 2419-2436, 2013.

[4] Salama, May & Kandil, Magy. (2016). An Improved DV-Hop Localization Algorithm Based on Modified Hop-Size. 10.1109/WSCAR.2016.11.

[5] D. Niculescu and B. Nath, "DV Based Positioning in the AdHocNetworks," Journal of the Tele-communication Systems, vol. 22, pp. 267-280, 2003.

[6] Maurice Clerc, "Particle swarm optimization," Chippenham: Antony Rowe Ltd, 2006.

[7] D. Venkateswara rao, B. Sahith, N. S. S. Kiran, K. D. Prasad, Dr. S. V. R. K. Rao, "Improved Distance vector-hop algorithm in sensor networks using least squares method", The Internation Journal of Analytical and Experimental Modal Analysis, vol. 15, pp. 554-564, 2022.

[8] A. Paul and T. Sato, "Localization in wireless sensor networks: a surveyon algorithms, measurement techniques, applications and challenges," Journal of Sensor and Actuator Networks, vol.6, no. 4, p. 24, 2017.

[9] J. S. Tashnim, C. Elkin, V. Devabhaktuni, D. B. Rawat, and J. Oluoch, "Advances on localization techniques for wireless sensor networks: a survey," Computer Networks, vol. 110, pp. 284-305, 2016.

[10] X. Hu, Y. Shi, and R. Eberhart, "Recent advances in particle swarm," in Evolutionary Computations, 2004. CEC2004. Congress on, vol. 1. IEEE, 2004, pp. 90-97.

[11] A. Gopakumar and L. Jacob, "Localization in wireless sensor networks using particle swarm optimization," in Wireless, Mobile and Multimedia Networks, 2008. IET International Conference on. IET, 2008, pp. 227-230.

[12] C. Rabaey and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in USENIX technical annual conference, 2002.

[13] G. Mao and B. Fidan, "Introduction to wireless senosr network local-ization," 2009.

[14] A. Pal, "Localization algorithms in wireless sensor networks: Current approaches and future challenges," vol. 2, no, 1, 2010, pp. 45-73.