Performance Evaluation of Hybrid Evolutionary Algorithms in Minimizing Localization Error for Wireless Sensor Networks

S Sivakumar\(^1\)* and R Venkatesan\(^2\)

\(^1\)Department of Information Technology, PSG College of Technology, Coimbatore-641004, India.
\(^2\)Department of Computer Science and Engineering, PSG College of Technology, Coimbatore-641004, India.

Received 10 November 2014; revised 16 September 2015; accepted 19 December 2015

Localization is considered as one of the most significant research issues in Wireless Sensor Network (WSN). The objective of localization is to determine the physical coordinates of sensor nodes distributed over the sensing field. Location information plays a vital role for coverage, deployment of sensor nodes, routing and target tracking applications. Initially, the localization of sensor nodes can be performed by Mobile Anchor Positioning (MAP), a range-free localization method. To further enhance the location accuracy obtained by MAP, we propose three algorithms, viz. Differential Evolution with MAP (DE-MAP), Ant Colony Optimization with MAP (ACO-MAP) and Simulated Annealing-Differential Evolution with MAP (SA-DE-MAP). The scope of this work is to compare the performance of these three algorithms. Root Mean Square Error (RMSE) has been used as the metrics for comparing the performance. Simulation result demonstrates that out of the proposed algorithms, SA-DE-MAP algorithm achieves better performance in minimizing the localization error when compared to DE-MAP and ACO-MAP algorithms.

Keywords: Localization, Mobile Anchor, Ant Colony Optimization, Differential Evolution, Simulated Annealing.

Introduction

Wireless Sensor Network (WSN) is a kind of ad hoc network that consists of autonomous sensors with low cost, low energy sensing devices, which are connected by wireless communication links. These sensor nodes are tiny in size and possess limited resources namely processing, storage, sensing and communication\(^1\). Localization is especially important\(^2\) when there is an uncertainty on the exact location of fixed or mobile devices. Localization is the process of making every sensor node in the sensor network to be aware of its geographic position\(^3\). As WSNs normally consist of a large number of sensors, the use of GPS is not a cost-effective solution and also makes the sensor node bulkier\(^4\). Currently the existing non-GPS based sensor localization algorithms\(^5\) are classified as range-based or range-free. Typical range-based localization techniques used are Received Signal Strength Indicator (RSSI)\(^6\), Time Difference of Arrival (TDoA)\(^7\), Time of Arrival (ToA)\(^8\), and Angle of Arrival (AoA)\(^9\). Range-based methods give fine-grained accuracy but the hardware used for such methods are expensive. Range-free or proximity based localization schemes rely on the topological information, e.g., hop count and the connectivity information, rather than range information but they provide coarse-grained accuracy. A Range-free Energy Efficient Localization Technique using Mobile Anchor (RELMA)\(^10\) especially for large scale WSNs is suggested to improve both accuracy and energy efficiency by minimizing the number of anchor nodes used. Weighted Centroid localization method\(^11\) using three mobile beacons was found superior compared to Weighted Centroid localization method with a single mobile beacon. Genetic algorithm\(^12\) proposed for localization of the sensor nodes provided good localization accuracy but the solution space was very huge and the algorithm has to search a large number of solutions in each of the iterations. A PSO based localization algorithm (PLA)\(^13\) is recently suggested for WSNs with one or more mobile anchors. PLA can achieve superior performance in various scenarios compared to Centroid localization method but provides only coarse-positioned location accuracy. Localization in Wireless Sensor Networks (WSNs) is viewed intrinsically as an unconstrained optimization problem\(^14\). The proposed evolutionary approach namely Hybrid Differential Evolution i.e. Simulated Annealing-Differential Evolution

\(^*\)Author for correspondence
E-mail: sivas_cbe@yahoo.co.in
Evolution with Mobile Anchor Positioning (SA-DE-MAP) algorithm is applied after performing location estimation using mobile anchors. This work uses a range-free approach, where the anchor nodes broadcast their location on the move and the obtained localization result is optimized by means of optimization as stated above.

The rest of the paper illustrates the existing method namely, Mobile Anchor Positioning (MAP)\textsuperscript{15} and elaborates the three proposed algorithms namely, Differential Evolution\textsuperscript{16} with Mobile Anchor Positioning (DE-MAP), Ant Colony Optimization\textsuperscript{17} with Mobile Anchor Positioning (ACO-MAP), and Hybrid DE\textsuperscript{18} approach i.e. Simulated Annealing-Differential Evolution with Mobile Anchor Positioning (SA-DE-MAP) and compares the performance of proposed SA-DE-MAP algorithm with DE-MAP and ACO-MAP algorithms and also with an existing algorithm known as Mobile Anchor Positioning (MAP).

**Proposed Localization Approach**

The localization strategy used in this work can be visualized to work in two phases. In the first phase, a range-free algorithm namely Mobile Anchor Positioning (MAP) is used for determining the location of the unknown sensor nodes. Since a range-free algorithm is used, (which offers coarse-grained accuracy) the obtained location will be just as an estimate. In the second phase (post optimization phase), an evolutionary strategy namely Simulated Annealing-Differential Evolution with MAP (SA-DE-MAP) algorithm is applied over MAP for fine-tuning the results of the sensor nodes obtained using MAP and thereby improving the localization accuracy.

**Mobile Anchor Positioning (MAP) Algorithm**

The simulation environment is set-up as follows: The sensor nodes are randomly deployed in the sensing field. The assumption made during simulation is that the Mobile anchors, which are location aware nodes move throughout the sensing field according to the positional data specified in a movement file, which is given as input to the Network Simulator (NS-2)\textsuperscript{19}. As they move around the sensing field, they periodically broadcast messages containing their current location at fixed time interval to all the nodes, which are at a hearing distance from it. Such messages are known as beacons. The mobile anchors traverse around the sensing field with a specific speed and their directions are set to change for every 10 seconds. All the nodes in the communication range of the mobile anchor will receive the beacons. A sensor node will collect all the beacons in its range and store it as a list. Communication range of the sensor node and the mobile anchor node are assumed as same. Once enough beacons are received and if a sensor node does not receive a beacon, which is at a distance greater than the already received ones, the localization begins at that particular node. Assume that the sensor node has received and stored four beacons (locations of the mobile anchor) in its list \{T\(_1\), T\(_2\), T\(_3\), and T\(_4\)\} as shown in Fig. 1. From the list, two beacons, which are farthest from each other, are chosen (T\(_1\), T\(_4\))\textsuperscript{29}. These points are known as Beacon points. These two points are marked as the end of the sensor node’s communication range, since the sensor node has not received a beacon farther from this point. Hence T\(_1\) and T\(_4\) (Beacon points) represent either two positions of the same mobile anchor or positions of two different mobile anchors when they were at the end of the sensor node’s communication range. With these two Beacon points as centers and the communication range of a sensor node as radius, two circles are constructed. Each circle represents the communication range of the mobile anchor, which has sent the beacon. The sensor node has to fall inside this communication range, as it has received the beacon. Since the sensor node has received packets either from both anchors or from the two positions of the same anchor, the node has to fall inside both the circles. Hence, it can be concluded that circles will intersect each other. The intersection points of both circles are determined (S\(_1\), S\(_2\)). The intersection points are the possible locations of the sensor node. The reason is as follows: The two farthest points (Beacon points) are the end points of a sensor node’s communication range. The sensor node lies on the circumference of the other circle since it is the same with the other mobile anchor position. Therefore, the sensor node lies on the circumference of both the circles. The only points satisfying the above condition are the two intersection points. Hence, by means of Mobile Anchor Positioning, the location of the sensor node has been approximated to two locations.

**Identifying the Sensor Locations using MAP with Mobile Anchor (MAP-M)**

The visitor list is searched after identifying the two possible positions i.e. the intersection points. If a node could hear around its range, there is a possibility of a beacon point which can be situated at a distance
r from one of the two possible locations. Thus, there is one point in the list, whose distance from one possible location is less than r, and the distance from other possible location is greater than r, then the first possible location is chosen as the location of the sensor node. It is assumed that the communication range of a mobile anchor is R. The MAP-M maintains the visitors list after receiving the beacon packets from the mobile anchor. The information from the visitor list is used to approximate the location of the sensor node. Let the visitor list of a sensor node S consists of various location information represented as \{T_1, T_2, \ldots, T_n\}. The beacon points are the two extreme points i.e., T_1 and T_n. Two circles with radius R and center T_1 and T_n are constructed and their intersection points of two circles are found to be S’ and S’’.

If there is any Ti (2 ≤ i ≤ n-1), such that the distance between Ti and S’ is less than R and that between Ti and S’’ is greater than R, then we can conclude the location of the sensor node is S’. This is because of the fact that the sensor node should lie inside the communication range of mobile anchor to receive the beacon packets. Consequently, the distance between the sensor node S and beacon packet Ti should be less than R. There is an area named as the shadow region, as shown in Fig. 1. If all the Beacon points lie inside this region, it is not possible to determine the location of the sensor as the shadow region comes under the range of both the intersection points. This could be explained by drawing two circles with S’ and S’’ as centre and the shadow region is the intersection of the two circles. Hence, in order to estimate the location of the sensor node there is a need that at least one of the beacon packets in the visitor list must lie outside the shadow region, as shown in Fig. 1. Therefore, it is not possible to determine the location of the sensor node S using the available beacon packets, thus the node is made to wait until it gets further beacon packets. If no further beacons are obtained, then a single position of sensor node S cannot be obtained. The node will have two positions S’ and S’’. To overcome this problem, the method of Mobile Anchor Positioning-Mobile Anchor & Neighbor (MAP-M&N) is being adopted.

**Localization Steps used in MAP-M&N Method**

The steps followed in finding the location of the sensors in the field using MAP-M&N method are listed below:

1. Deploy 100 sensor nodes randomly in the 1000 m × 1000 m area of the sensing field in the simulation environment and deploy 3 location aware nodes (anchor nodes) i.e sensor nodes with GPS.

**Forming Additional Anchors & Identifying Sensor Locations using MAP with Mobile Anchor & Neighbor (MAP-M&N)**

The location estimation done for sensors using MAP-M method gives positions for few sensors and for the others, it gives two positions and so it is the responsibility of MAP-M&N method to produce outputs with a single position for each sensor. It is possible for the sensor nodes that have already determined their location to assist other nodes in determining their locations. As soon as the location is identified, the localized nodes start acting like anchors. They embed their calculated location inside the packet and then broadcast the beacons. Nodes, which are at its hearing range and waiting for additional beacons to finalize their location, can make use of these beacons. However, if the sensor node has determined its location, it simply discards the beacon packet. As a consequence, by using MAP-M&N method, the cost of movement of the mobile anchor can be reduced.
The Mobile Anchor nodes move throughout the sensing field according to the positional data specified in the movement file which is given as input to the NS-2 simulator. The anchor nodes periodically broadcast their location packets, which are known as beacon packets, while on the move through the sensing field.

Every sensor node maintains a visitor list containing beacon packets based on the information obtained from anchors. The sensor nodes can identify the farthest beacon packets and chooses those beacon packets as beacon points.

With those two beacon points as the centers and the communication range of a sensor node as radius, two circles are constructed and the intersection points are found.

Sensor nodes try to identify its position out of the two intersection points. Here, at least one of the beacon points in the visitor list must lie outside the shadow region or based on the beacon points obtained from neighbor nodes.

The approximate location for each of the sensor nodes is estimated using MAP-M&N method.

**Proposed Differential Evolution with Mobile Anchor Positioning (DE-MAP) Algorithm**

The localization steps used in Differential Evolution with MAP–M&N are the following:

1. The algorithm takes the results of MAP–M&N as its input. The results of MAP-M&N, giving the approximate solution of the location of each sensor at each specified time instance is given as the input to the post optimization method.

2. Each node will separately undergo the process of differential evolution to produce a fine grained accurate location.

3. The initial population P of random individuals is the output produced from MAP-M&N method for each node.

4. Check for the stopping criteria and continue if the stopping criterion is not met.

5. For each individual $N_i$ ($i = 1, 2, ..., \text{pop Size}$) from $N$, repeat the following:-
   a) Create the new location from the chosen locations which acts as the parents. Choose three parents from the population.
   b) The calculation of the new location (child) is as stated by Equation (1) below:-

   $$(x,y) = \text{Parent1} + F \cdot (\text{Parent2} – \text{Parent3}) \ ... \ (1)$$

   Where F is a scaling factor, assumed as 1.

6. Calculate the fitness for the new location (child). If its fitness is better than that of the Parent’s, then the new the location replaces parent. Otherwise, the new location is discarded.

7. Repeat the procedure until the stopping criterion is met.

**Proposed Ant Colony Optimization with Mobile Anchor Positioning (ACO-MAP) Algorithm**

The localization steps followed in Ant Colony Optimization (ACO) with MAP-M&N algorithm are listed below:

1. The algorithm takes the results of MAP–M&N as its input. The results of MAP-M&N, giving the approximate solution of the location of each sensor at each specified time instance is given as the input to the post optimization method.

2. Let each node's x and y co-ordinates at different instances of time be $(x_1,y_1), (x_2,y_2), \ldots \ldots, (x_n,y_n)$, where $n$ denotes the number of sensor nodes. Each of these positions is considered as a separate ant. Hence producing as much of ant as that of the approximate positions found at regular intervals.

3. The first ant moves randomly until it finds the food source, then it returns to the nest, laying a pheromone trail. Once the ant's path is complete, by applying the local update rule it updates its pheromone value.

4. Other ants follow one of the paths at random, also laying pheromone trails. Since the ants on the shortest path lay pheromone trails faster, this path gets reinforced with more pheromone, making it more appealing to future ants.

5. When all ants have completed their path, it modifies the pheromone value by global update rule again.

6. The ants become increasingly likely to follow the shortest path since it is constantly reinforced with a larger amount of pheromones. The pheromone trails of the longer paths evaporate.

7. When the ants select the path again, it will tend to choose the path of higher pheromone, and release the pheromone.

8. After much iteration, there will be a pheromone on a path much more than the other path, this path is the optimal path.

9. Repeat the procedure until the stopping criterion is met.
Stopping criteria = Maximum iterations or Profit value.  
Maximum iterations = arbitrarily chosen as 100.

**Proposed Simulated Annealing-DE with MAP (SA-DE-MAP) Algorithm**

The localization steps followed in Simulated Annealing-Differential Evolution with Mobile Anchor Positioning (SA-DE-MAP) (i.e. Hybrid DE algorithm) are listed below:

1. The algorithm takes the results of MAP–M&N as its input. The results of MAP-M&N, giving the approximate solution of the location of each sensor at each specified time instance is given as the input to the post optimization method.
2. Each node separately undergoes the process of differential evolution to produce a finely accurate location.
3. For each individual \( N_i \) \((i = 1, ... \text{ pop Size})\) from \( N \), Initialize location points and optimal points.
4. Randomly select the local best solution from parent chromosomes.
5. At time step \( t \), the \( i \)-th parent chromosome is \( X_i(t) \) and its offspring, created through the DE-type mutation and crossover operations at the next time step, is \( U_i(t + 1) \).
6. If the offspring is better than the parent with respect to objective function \( f(U_i(t + 1)) < f(X_i(t)) \), then \( X_i(t) \) is surely replaced in the next generation with \( U_i(t + 1) \). Otherwise \( X_i(t + 1) = X_i(t) \).
7. But even if \( f(U_i(t + 1)) > f(X_i(t)) \), \( U_i(t + 1) \) may replace the parent chromosome \( X_i(t) \) as shown in Equation (2) with a probability \( P_i \) if:
   \[
P_i(\text{rand}(0,1)) < \exp \left( -\frac{f(U_i(t + 1)) - f(X_i(t))}{T} \right) \quad \ldots \quad (2)
   \]
8. Check for the stopping criteria and continue if the stopping criterion is not met.

**Simulation Results and Discussion**

The following parameters namely number of mobile anchors, speed of mobile anchors, number of sensor nodes and execution time are varied and the results were analyzed using NS-2 for each of the parameter variation. Random way-point mobility model is considered for our simulation. AODV is the routing protocol used especially for broadcasting messages (Hello Packets) during localization. The simulation results are generated after performing ten trials and the average values have been presented. From the various simulation experiments carried out for Mobile Anchor Positioning (MAP), the simulation settings mentioned in Table 1 is followed when the proposed Simulated Annealing-Differential Evolution (SA-DE) Algorithm was applied over MAP-M&N algorithm and also to compare the performance of Differential Evolution with MAP (DE-MAP) and Ant Colony Optimization with MAP (ACO-MAP) algorithms. With the NS-2 simulation settings as specified in Table 1, the results were analyzed by comparing the performance of the three proposed approaches namely, SA-DE-MAP to that of DE-MAP and ACO-MAP algorithms respectively and the respective graphs were plotted. When the simulation time was considered as 100 sec, the percentage of localized nodes is not appreciable. But, when the simulation time was increased to 500 sec, the percentage of localized nodes is much improved. Thus the simulation time is chosen as 500 sec. Considering 100 nodes scenario, in a trial, when the number of iterations are varied from 25 to 100, the range of error is 8.82 to 8.44 in SA-DE-MAP, 10.44 to 8.95 in DE-MAP and 11.05 to 8.92 in ACO-MAP algorithms respectively. Hence it is observed that the positional error is less in SA-DE-MAP algorithm compared to DE-MAP and ACO-MAP algorithms.

**Metrics used to determine the Localization Accuracy**

Localization problem can be mathematically stated as follows: Consider a network formed by \( L = m + n \)

<table>
<thead>
<tr>
<th>Parameter Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area of the sensing field</td>
<td>1000 m × 1000 m</td>
</tr>
<tr>
<td>Number of sensor nodes</td>
<td>100</td>
</tr>
<tr>
<td>Number of mobile anchors</td>
<td>3</td>
</tr>
<tr>
<td>Speed of mobile anchors</td>
<td>100 m/s</td>
</tr>
<tr>
<td>Time interval between successive anchors</td>
<td>1 s</td>
</tr>
<tr>
<td>Execution time</td>
<td>500 s</td>
</tr>
<tr>
<td>Transmission range</td>
<td>250 m</td>
</tr>
<tr>
<td>Routing protocol</td>
<td>AODV</td>
</tr>
<tr>
<td>MAC protocol</td>
<td>IEEE 802.11</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random way-point</td>
</tr>
<tr>
<td>Number of generations</td>
<td>10-100</td>
</tr>
</tbody>
</table>
sensor nodes, where $m$ represents the anchor nodes and $n$ represents the non-anchor nodes. The goal of a localization system is to estimate the coordinate vectors of all $n$ non-anchor nodes. Let the actual $x$ and $y$ coordinates of sensor nodes generically be $(x_{\text{act}(i)}, y_{\text{act}(i)})$, and the obtained values of $x$ and $y$ coordinates of sensor nodes be $(x_{\text{obt}(i)}, y_{\text{obt}(i)})$, where $i$ ranges from 1 to $n$. The metrics that is used to evaluate the accuracy in localization process is Root Mean Square Error (RMSE)\(^2\). The RMSE formula shown in Equation (3) is used for calculating the average error obtained by MAP-M&N and by the proposed approaches stated as,

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} [x_{\text{act}(i)} - x_{\text{obt}(i)}]^2 + [y_{\text{act}(i)} - y_{\text{obt}(i)}]^2} \quad \text{... (3)}$$

Where, $x_{\text{act}(i)}$, $y_{\text{act}(i)}$ – denote the actual $x$ and $y$ coordinates of sensor nodes and $x_{\text{obt}(i)}$, $y_{\text{obt}(i)}$ denote the obtained values of $x$ and $y$ coordinates of sensor nodes.

$N$ – represents the total number of localized sensor nodes.

**Comparison of RMSE for SA-DE-MAP, DE-MAP and ACO-MAP algorithms**

The accuracy in localization can be evaluated based on minimization in positional error. RMSE value is evaluated for SA-DE-MAP, DE-MAP and ACO-MAP algorithms by varying the number of nodes. The results of simulation are plotted graphically as shown in Fig. 2. Consider for instance, the actual position of one particular node which is placed in the simulation environment is (218.78, 684.98). The calculated position of this node is found to be (215.78, 682.98) in SA-DE-MAP, (210.88, 682.73) in DE-MAP and (209.84, 681.01) in ACO-MAP algorithms respectively. This example is taken to illustrate that deviation is less in SA-DE-MAP compared to DE-MAP and ACO-MAP algorithms. In a similar manner, the estimated positions of all the remaining sensor nodes placed in the simulation environment are observed. The RMSE analysis of the three proposed algorithms namely, DE-MAP, ACO-MAP and SA-DE-MAP corresponding to 10, 20, 30 etc. up to 100 nodes scenario illustrates that on an average, the RMSE value gets reduced in SA-DE-MAP algorithm compared to ACO-MAP and DE-MAP algorithms.

**Conclusions**

Mobile Anchor Positioning is a range-free localization method that does not involve usage of any hardware. MAP-M&N method is considered appropriate for localization as the percentage of localized nodes is high. Since this method does not provide fine-grained accuracy, optimization techniques are applied over the results of MAP-M&N. In this paper, DE-MAP, ACO-MAP and SA-DE-MAP algorithms have been proposed over MAP-M&N to minimize the localization error. Then Hybrid DE algorithm namely, SA-DE-MAP algorithm has been applied over MAP–M&N in order to reduce the localization error further. The input set considered are 100 sensor nodes randomly placed over a 1000 m × 1000 m area of the sensing field and 3 mobile anchor nodes as reference. The mobile anchor nodes move throughout the sensing field according to the positional data specified in the movement file which is given as input to NS-2 simulator. Initially, the positions of 100 sensor nodes are estimated using MAP-M&N method. The results obtained using MAP-M&N method becomes the initial population for the three proposed evolutionary algorithms. From the RMSE comparison graph given in Fig. 2, pertaining to 100 nodes scenario, Simulated Annealing-Differential Evolution with Mobile Anchor Positioning (SA-DE-MAP) algorithm minimizes the localization error marginally by 1.54 % (calculated by percentage error formula) when compared to DE-MAP algorithm and by 4.82 % (calculated by percentage error formula) when compared to ACO-MAP algorithm. Other evolutionary approaches available in the literature can be applied for minimizing the localization error further.
References

19. Network Simulator ns-2; http://www.isi.edu/nsnam/ns