An Efficient Approach for Multi-Target Tracking in Sensor Networks using Ant Colony Optimization

Fathima Shareena K. P.1
1M. Tech, Communication Engineering
1KMEA Engineering college, Aluva

Abstract— Multi-Target Tracking for Sensor Networks using Ant Colony Optimization is proposed in this paper. The proposed approach uses ant colony optimization technique that composed of both dynamic and mobile nodes. While mobile nodes are used for optimizing the target tracking, dynamic nodes ensure the total coverage of the network. As a result, the performance of the Wireless Sensor Networks with multi-target tracking on mobility nodes will improve. While improving the performance of the wireless sensor networks, automatically minimum distance will be travelled by the nodes, thereby decreasing the energy consumption of the mobility nodes. This is achieved by selecting the optimal path by searching each path of the existing network. The software should check whether any chance of collision and if it is happening, then how to avoid it by providing a minimum delay for finding the optimal path. For finding the prediction of each node to reach the optimal path different methods should be applied, like k-means and random selection. The experimental results show that Object tracking with prediction mechanism is much better than without prediction mechanism.

Keywords: Wireless Sensor Networks, Mobile Nodes, K-Means, Ant Colony Algorithm, Multi-Target Tracking.

I. INTRODUCTION

Mobile Sensor Networks (MSN) are networks composed of a large number of wireless devices having sensing, processing, communication, and movement capabilities [1]. The main constraint of sensor nodes is their limited energy resources since their batteries are nonrenewable. One important factor is thus to reduce the energy consumption of the sensors in order to increase the lifetime of the network. One can distinguish between two types of mobility in MSN: the uncontrolled (also called passive) mobility, where sensors are moved in an uncontrollable manner, and the controlled mobility, where sensors are moved in response to internal or external commands. Passive mobility makes the use of MSN more challenging since sensor nodes need to be relocated continuously; whereas in controlled mobility, one could take advantage of the mobility of the nodes to improve the accuracy of the sensed data in the network.

MSN have a variety of applications in different fields, such as military and environment monitoring [2]. One interesting application of MSN is target tracking. It consists of estimating instantly the position of a moving target. It is of great importance in surveillance and security especially in military applications. This problem has been mainly considered for networks having static nodes [3]. However, when sensors are able to move, it is important to take advantage of their mobility in order to improve the position estimation. This contribution focuses on target tracking in MSN where nodes have a controlled mobility. Different techniques have been proposed to manage the mobility of the nodes [4]. These techniques have mainly focused on upgrading the topology of the network, improving the area coverage or increasing the lifetime of the network, etc. A few methods have been developed for target tracking in MSN [5]. Nodes are thus able to move to a new position chosen within a set of candidate locations, being at one step away from the current location. The movement decision is made upon whether the new location will improve the tracking quality or not. The number of candidate locations is limited, as well, in order to reduce the complexity of the method. In a different researchers in [6] have considered the problem of a mobile target, called mouse, trying to avoid detection by mobile sensor nodes, called cats. In this paper, we propose a novel strategy for managing sensors mobility, aiming at improving the tracking of a single target. The method consists of four consecutive phases that iterate at each time step as follows:

1. Estimating the current position of the target,
2. Predicting the next-step position of the target using current and previous position estimates,
3. Computing a set of new locations to be taken by the mobile nodes in the way to improve the estimation process,
4. Assigning each mobile node one new location within the computed set using the ant colony optimization algorithm (ACO).

The estimation phase is performed using random selection. The target positions are performed with prediction mechanism and without mechanism. Predicting the next position leads to a region of interest to be covered. The relocating phase consists of moving the nodes in order to improve the assessment phase while minimizing the energy consumption. The optimization phase is completed using the Ant Colony Optimization algorithm [7]. Using ant colony method, the ACO is able to solve efficiently complex operational research problems. Due to its stochastic nature, it affords the combinatorial explosion in the number of possible solutions. Besides the energy limitation, one important constraint is to maintain the total coverage of the network while moving the sensor nodes. The whole monitoring area should be covered by sensors to be robust to any other intruders. For this reason, two types of sensors are used: static and mobile nodes. While mobile sensors are moved to improve the quality of target tracking, static nodes are uniformly distributed in order to ensure a continuous coverage of the network independently of the movement of the mobile ones.

This paper is organized as follows: Section II includes the clustering using K-means algorithm. Section III describes the Ant Colony Optimization Algorithm. Section IV
presents the system model for the proposed work. Section V gives the experimental results and discussions. The paper is concluded in Section VI.

II. CLUSTERING USING K-MEANS ALGORITHM

K-Means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori [8]. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point it is necessary to re-calculate k new centroids as bar centers of the clusters resulting from the previous step. After obtaining these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop, one may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function.

\[
 J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 
\]

Where \( x_i^{(j)} \) is a chosen distance measure between a data point \( x_i \) and the cluster centre \( c_j \), is an indicator of the distance of the \( n \) data points from their respective cluster centers. The algorithm is composed of the following steps:

Step 1: Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.

Step 2: Assign each object to the group that has the closest centroid.

Step 3: When all objects have been assigned, recalculate the positions of the K centroids.

Step 4: Repeat Steps 2 and 3 until the centroids no longer move.

This produces a separation of the objects into groups from which the metric to be minimized can be calculated. Although it can be proved that the procedure will always terminate, the K-Means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. K-Means is a simple algorithm that has been adapted to many problem domains and it is a good candidate to work for a randomly generated data points. One of the most popular heuristics for solving the K-Means problem is based on a simple iterative scheme for finding a locally minimal solution. This algorithm is often called the K-Means algorithm.

![Flowchart of K-Means Method](image)

Fig. 1: Flowchart of K-Means Method

III. POSITIONING OF NODES USING ANT COLONY OPTIMIZATION

Having the set of positions that must be taken by the sensors, one should assign each sensor one position within the set while minimizing the traveled distance of the nodes. The problem is thus defined as an optimization algorithm that is solved using the ACO. In the following, first introduce the ACO. We then apply it to the relocation problem.

A. Ant Colony Optimization Algorithm

In the early 90s, M. Dorigo discovered Ant-Colony Optimization Algorithm (ACO) [9]. The inspiring source of ACO algorithms is the foraging behavior of ants in the real world. The ACO is a probabilistic method for solving complex computational problems. This algorithm was first developed to solve the Travelling Salesman Problem. It has been applied efficiently afterwards in different fields such as quadratic approximation problems, vehicle routing [7]. The main idea of ACO consists of imitating the behavior of real ants in their way to find the shortest path to get food sources. A path is thus generated according to two elements: a chemical substance called pheromone and the visibility of the ant which in turn determines the path to find the Target [10]. Let f(x₁, . . . , xₙ) be a function of n variables whose values are taken from a specific set S. Optimizing f consists of finding the n-permutation of (x₁, . . . , xₙ) over all possible permutations that optimizes the function f. In such problems, the function f is called objective function or fitness function, whereas x₁, . . . , xₙ are called the decision variables. Let m be the cardinal of S, then the number of all possible n-permutations is equal to m! / (m-n)!

With m! Being the factorial of m. evaluating all solutions requires too much computational time especially for large-size problems. In such cases, using optimization algorithms such as ACO becomes crucial to reduce the time of computation.

Starting with an initial solution, ACO moves toward optimal solutions using an efficient memory-based search technique.
The generation of solutions basically employs two parameters: visibility and pheromone. These parameters correspond to a priori and a posteriori information about the solutions, respectively. While visibility remains unchanged, pheromone is modified during the optimization process according to solutions evaluation. Technically, the ACO algorithm considers a fixed number of ants $K$, each of which generating one solution at every iteration. Solutions are thus encoded by assigning each decision variable, one after the other, a specific value of $S$.

B. Sensor Relocation

The relocation problem consists of minimizing the total distance traveled by the nodes while moving to their new positions. The fitness function is thus equal to the sum of the distances traveled by the moving nodes, whereas the decision variables are the sensors coordinates. These variables take their values within the set of the positions. The relocation method for a given time step $t$ is illustrated.

IV. SYSTEM MODEL

The proposed architecture consists of mainly two sections. One is the tracking section and the other one is the visualization part. The main part of the project deal with the tracking methods. For each and every new step formation, there will be adding a separate technique or a suitable algorithm for tracking the target. The upper part of the tracking section and the lower part show the data analysing as well as the visualization of both the target and optimal path formation.

A. Overall System Architecture

First of all, the generation of number of nodes is defined in the network and the range of the network is determined. The mobile nodes should set up according to the range of the network in addition to the easier communication with the neighbouring nodes. Then the wireless sensor network set up should be built in order to connect each and every node with the adjacent nodes. After that, according to adjoining nodes the clustering should be done for the easier path detection. The clustering is done with the k-means technique. The next step is the target detection. By using the ant colony optimization method, several optimal paths can be found out. The next predicate position can be easily found out after finding each hope between the nodes in the network. For fixing this node selection, prediction mechanism is to be used. For selecting the optimized path, ant colony optimization technique is used. Thus, the battery consumption can be reduced.

The second section consists of finding the target. For that, tracking the moving object should be found in each and every moment. And the details should be passed to the neighbouring nodes as well. Depending on the information received by the nodes, the optimal path will be opting for. At last it should be visualized in the software. Figure below shows the overall system architecture.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

The experimental results show that Object Tracking with prediction mechanism is much better than without prediction mechanism. Mode selection is included for selecting stored values and random values. The number of nodes can be changed at any time. Firstly deploy the network depending on number of nodes. Then next step is to model the cluster head and clustering division. After that, set the source node and destination node manually. Later find the minimum distance depending on the iterations for two paths. This is done with prediction mechanism and without prediction mechanism.

![Table 1: Result of Object Tracking Without Prediction Mechanism](image)

<table>
<thead>
<tr>
<th>No. of Items</th>
<th>No. of Iterations(x)</th>
<th>Minimum Distance y (metre)</th>
<th>Minimum Distance y1 (metre)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>498</td>
<td>468</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>367</td>
<td>350</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>288</td>
<td>280</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>156</td>
<td>148</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>120</td>
<td>112</td>
</tr>
</tbody>
</table>

Table 1: Result of Object Tracking Without Prediction Mechanism

While considering the measurement values from the table 1 and table II, it is obviously clear that that Object Tracking with prediction mechanism is much better than without prediction mechanism in terms of Minimum Distance. Since Object Tracking with prediction mechanism has low Minimum Distance and showing high performance.

![Table 2: Result of Object Tracking With Prediction Mechanism](image)

<table>
<thead>
<tr>
<th>No. of Items</th>
<th>No. of Iterations(x)</th>
<th>Minimum Distance y (metre)</th>
<th>Minimum Distance y1 (metre)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>450</td>
<td>435</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>300</td>
<td>95</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>210</td>
<td>185</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>110</td>
<td>107</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>108</td>
<td>105</td>
</tr>
</tbody>
</table>

Table 2: Result of Object Tracking With Prediction Mechanism

A. Comparison of two paths with and without Prediction mechanism

The graph explains with given values for source and destination for object tracking without and with prediction mechanism. The more the iterations, the more absolute solution can be achieved. Depending on the generated hop or weight of the edge or the distance between neighbouring nodes will influence the minimum distance from one particular source to destination. The graph shows the minimum distance in the decrease level from top to bottom. The x-axis shows the number of iterations as well as y-axis shows the minimum distance of two paths acquired by two source and destination.
VI. CONCLUSION

In this paper, while comparing the two graphs, the minimum distance acquired with prediction mechanism that shows more accurate value than without prediction mechanism in the object tracking method. Using multi-target tracking in the wireless sensor node thus gives faster response using ant colony optimization. The energy consumption is automatically achieved by using clustering and cluster head formation method. Depending on the iterations, the comparison can be easily done and good result is achieved. From the graph itself, it is known that object tracking using prediction mechanism gives better performance than without prediction mechanism in tracking. Simulation results show the efficiency of the proposed system as well as the performance increase in finding the minimum distance between the two nodes.

REFERENCES


