An Efficient Path Planning Algorithm For Networked Robots

MIRIAM CATHY JOY[1], R.JAYAPARVATHY[2]
Department of Electronics and Communication Engineering
SSN College of Engineering,Chennai,India
miricjoy@gmail.com [1], jayaparvathyr@ssn.edu.in [2]

Abstract

Networked robots refer to multiple robots operating together in coordination using wired or wireless networks for communication to share and distribute tasks. We focus on the problem of path planning of these robots in an unknown environment with obstacles. In the existing techniques, the robots collaboratively find the location of obstacles and available path, and exchange the entire map of the environment among themselves without information from the static sensor network. This exchange of map information creates computational overhead and communication overhead in complex environments. We propose the Directed Ant Colony Optimization Algorithm in which the environment detection task is shared between the robot and the static sensor network, thereby reducing the computational overhead. The D-ACO algorithm is analyzed for both grid and random deployment of sensors. On comparison with the existing ACO techniques, simulation results show that the rate of convergence of the D-ACO algorithm is increased by two times in grid deployment, and by four times in random deployment of static sensors. Also, due to the use of the target-directed pseudogradient ,the D-ACO algorithm finds the shorter path with less convergence time compared to conventional ACO algorithm.

Keywords

Networked robots, path planning, multi-hop communication, directed ant colony optimization algorithm, RSSI

1. Introduction

Networked robots are a group of mobile robotic devices operating together in coordination. The individual robots are less intelligent but highly communicative which enables them to solve a complex task together. Interesting applications of networked robots can be found in disaster management, emergency rescue, military, communication, transportation, and factory automation. For example, in road traffic control, with the support of intelligent road information, based on wireless sensor networks, autonomous vehicles can be induced to choose suitable pathways. The major challenges faced in the area of networked robotics can be categorized into design of the system, simultaneous localization and mapping(SLAM), cooperation between the robots, path planning, and communications . We focus on the problem of path planning which aims at constructing collision-free trajectories from a given initial location to a target location .

Many techniques for path planning of networked robots have been discussed in literature. Jung et al presented a multi-robot path planning algorithm in [1] in which the individual robots detect the paths and obstacles in the environment and exchange the entire map of the environment with the other robots in the team. Zhang ,Dawei and Chen [2] describe a navigation algorithm named DRAPP(Distance and Robustness Aware Path Planning) which uses the RSSI- distance characteristics and odometry to make the robot to travel in the shortest geometrical path to reach the target node. A layered dual-swarm framework with three communication channels was proposed by Shen et al in [3],to provide an efficient interaction channel for both WSN network and mobile multi-robot swarm to cooperate. Zhou and Tan present a feasible scheme of WSN-aided mobile robot navigation in [4] which includes initial localization of mobile robot, orientation adjustment, path planning, and position correction based on RSSI in Grid-pattern WSN. Jehn-Ruey et al proposed three schemes in [6] to coordinate and navigate mobile robots with directional antennas in a positionless wireless sensor network for the purpose of emergency rescue, namely, *k*-farthest-node forwarding scheme, Mobile Robot Coordination (MRC)scheme and Tree Assisted Navigation (TAN) scheme .

We propose a technique to reduce the memory overhead of the robots by storing the information about the environment (i..e location of obstacles and target) in the static sensor nodes. We propose the Directed Ant Colony Optimization algorithm in which each robot is considered as an artificial ant, which searches for the optimal path. In nature, the ants communicate with each other through the environment, by leaving a pheromone trail. The following ants trace this trail to find the path to the food source. To mimic this behavior, the artificial ants communicate with each other through a wireless sensor network (WSN). We develop a feedback mechanism to update the probability of selection of the nodes corresponding to the shortest path found during navigation .The proposed scheme uses the distributed nature of the wireless sensor network to assist the robot in path planning. The WSN nodes detect the target and obstacle location. Based on this information, the robot has to build a feasible path to the target. The task of path planning is shared between the robot and static sensor nodes.

The rest of the article is organized as follows**:**

Section 2 explains the proposed system model which is used for simulating the path planning algorithm.Section 3 discusses the WSN-based implementation of the proposed navigation algorithm called Directed Ant Colony Optimization (D-ACO). In section 4,the hardware implementation of the proposed scheme is explained .In section 5,the results obtained by implementing the D-ACO algorithm in grid and random deployment of sensor nodes is presented. Section 6 describes the conclusions drawn from the work and the future extensions.

2. System Model

A Wireless Sensor Network environment which has both static nodes and mobile nodes is considered. The static nodes have information about the target which is calculated based on a pseudogradient as discussed in section 3. The mobile node is the robot which communicates with the static sensor nodes to get the information about the environment. The static sensor nodes act as signposts to guide the robot towards the target. The complexity of motion planning is shared by the mobile robot and the distributed intelligence inherent in the WSN.

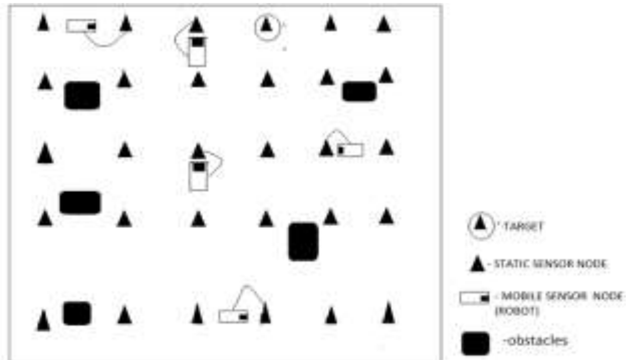


Figure 1. System Model of the Proposed Scheme

The following assumptions are made in the proposed scheme:

1. The nodes are distributed in grid pattern and random pattern in a square field of 600 X 600 mts.

2. All nodes in the WSN are reachable by multihop communication.

3. A swarm of 5 mobile robots is considered to solve the path planning problem in the environment with obstacles.

4.The optimal radio range for a node that ensures full connectivity of the network is given by $R = \varphi\sqrt{\frac{\log N}{N}}$

The φ parameter stands for a 2D plane diameter directly proportional to the number of nodes N [10].

3.Proposed Navigation Algorithm

We propose a Directed Ant Colony Optimization(D-ACO) algorithm which enables the robot to navigate in the WSN environment. It differs from the conventional ACO algorithm in that ,the direction in which the target is located is specified by the static sensor nodes. The coordinates of the deployed sensor nodes are known in advance. The robots search for all possible solution in that particular direction instead of searching the entire solution space. This reduces the computational overhead for the robots. The overall flow chart of the proposed navigation algorithm is given below

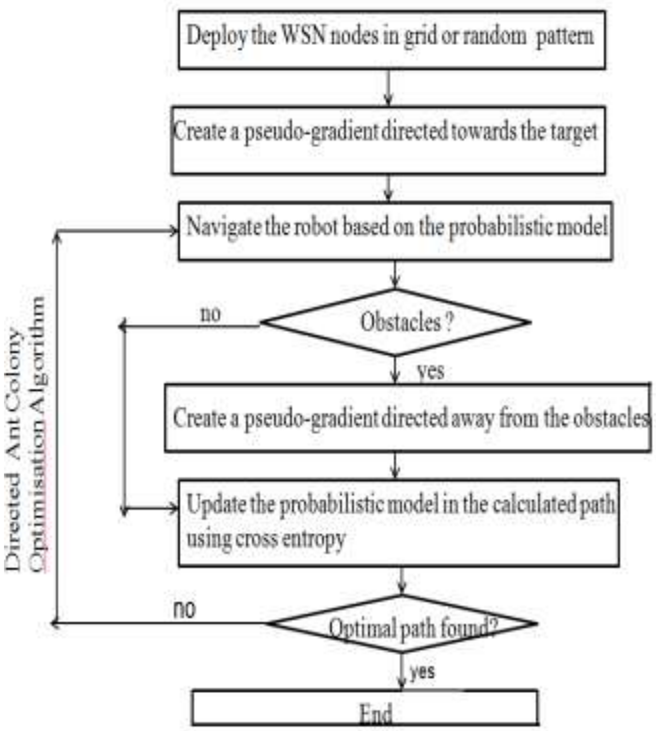


Figure 2. Overall flowchart of the proposed scheme

Each step of the algorithm is explained in detail in the subsequent sections.

3.1 Creation Of Pseudo-gradient

The static sensors provide the environmental information to the robots using two pseudo-gradients.(i) target-directed pseudo-gradient (ii) obstacle-directed pseudo-gradient .

The target-directed pseudo-gradient is created based on the idea given by Deshpande et al [3].When any static node detects a target, it communicates it to all nodes in the network via multihop communication. Each sensor node is assigned a weight which is a function of the node's hop distance from an identified target location. As a consequence, the target node has the lowest weight assigned to it(indicated bylight color in Fig 5). Each subsequent sensor node increments the hop-count value that it receives by one, before broadcasting the message to its neighbors. We use the hop-count as weights for the sensor nodes. In this way, the weights of the subsequent sensor node increases in ascending order, with the target having the minimum weight. The node farthest from the target has the highest weight assigned to it(indicated by dark colour in the Fig 4). Since a constant communication radius of r is assumed, a sensor node with a hop count of h would be at most a distance of h * r from the target node.

For the target node, pseudo-gradient weight = 0

(since hop count of the target is 0)

For all other nodes,

pseudo-gradient weight = Neighbor node's weight +1

The mobile robot placed into this environment can follow the decreasing order of the weights on the sensor nodes to reach the target from any location within the region.

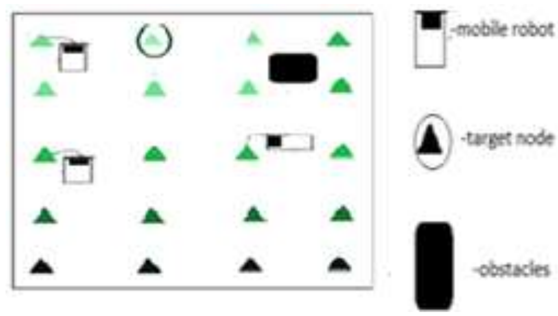


Figure 3.Target-directed Pseudo-gradient

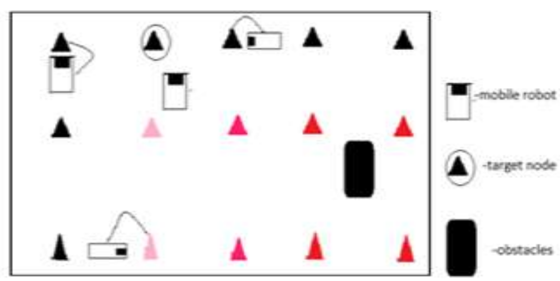


Figure 4.Obstacle-directed Pseudo-gradient

In addition to this we create an obstacle-directed pseudo-gradient to guide the robots away from the obstacles. When a robot detects a obstacle, the static nodes near the obstacles are given a high weight (indicated by dark colour in Fig 6) and its neighboring nodes are given a decreasing order of weights, creating a pseudo-gradient. In this way,the leading robot conveys the information about the environment to the following robots using the pseudo-gradients. The following robots chooses the minimum weight path, avoiding the obstacles. Based on the feasible direction determined using the pseudo-gradients, the robots navigate using the directed Ant Colony Optimization algorithm .

3.2 Directed Ant Colony Optimization Algorithm

The combinatorial optimization problem which we consider in our work is the shortest path problem .We propose the Directed Ant Colony Optimization(D-ACO) Metaheuristic to find a optimal solution to the shortest path problem .We consider the mobile robots as the artificial ants which plan the path between the source and destination. The individual ants find solutions to the shortest path problem by performing a random walk in the environment based on a probabilistic model,leaving behind a pheromone trail. They communicate this solution to the other ants indirectly through the environment by a property called stigmergy. As our system model incorporates a WSN environment, the ants store their solutions in the WSN nodes as pheromone trails . Given a node i , we interpret the pheromone values on all its neighboring nodes j as a probability distribution using which the following ants (mobile robots) make a intelligent choice, to reach the goal. In this way, the other ants need not search the entire solution space for feasible solutions. The search by the future ants for the optimal solution is biased by the pheromone values. As the algorithm iterates , the overall process converges toward having the majority of the robots following a single trail,which tends to be a near optimal path from the source to the destination. The optimal solution is found due to the collective interaction among the ants.

The objective function f, for the shortest path problem is given as

$$f = \min \left(\sum_{n=1 \ldots p,\ k \in \beta_n}^{\infty} L_k \right) \qquad \text{----(1)}$$

where β_n is the complete path from source to goal

p is the number of complete paths

L_k , k =1......m are the lengths of the edges in a path.We solve the shortest path problem using model-based search approach of D-ACO algorithm.

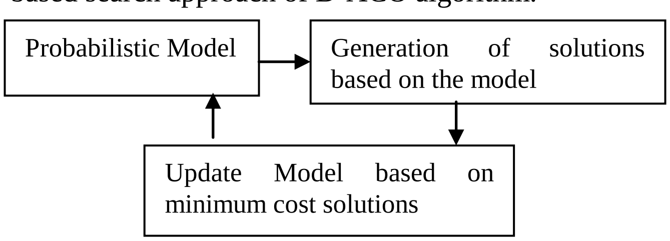


Figure 5. Model based search approach for DACO algorithm

In model-based search algorithms, candidate solutions are generated using a parameterized probabilistic model that is updated using the previous solutions in such a way that the search will concentrate on the regions containing high-quality solutions.

The steps in the model based search approach for DACO algorithm are given as

Step1 : Generate random solutions $S_1, S_2, S_3 \ldots S_n$ according to the probabilistic model . Each solution is a set of nodes and corresponds to a possible path between the source and target nodes.

Step 2 : Calculate the cost function f(S) (i..e path length) for each solution .

Step 3 : The nodes that occur often in the minimum - cost feasible paths are given high priority by increasing their corresponding probabilities in the probabilistic model. In this way, the probabilistic model is updated for the next iteration

Step 4: Iterate steps 1 to 3 till the algorithm converges to a optimal solution. Convergence occurs when the difference between the probability distribution(i.. cross entropy) in the current iteration and next iteration is less than ρ. The value of ρ is set between 0.01 and 0.1

3.3.1 Probabilistic Model

When a robot is at a current node i, it makes a transition to the next node j using the probability values in the transition probability matrix. We define the transition probabilities as a function of the pheromone value, τ_{ij} and heuristic value, η_{ij}

$$P_{ij} = \frac{\tau_{ij}\,\eta_{ij}}{\sum_{j \in N_i} \tau_{ij}\,\eta_{ij}} \qquad \text{----(2)}$$

where j is selected from the set of nodes in the neighborhood of i, N_i .

The heuristic value is defined as $\eta_{ij} = \frac{1}{d_{ij}}$.

The value d_{ij} gives the Euclidean distance between the current node i and possible neighboring nodes j .The nodes i and j are one-hop neighbors in the WSN network under consideration. The nodes j are chosen in such a way that they have a decreasing value of the pseudo-gradient as mentioned in the previous section.

3.3.2 Probabilistic Model Updation

The total collection of pheromone markings in a network at current iteration is modeled by a probability matrix P_m where each element $P_{m,ij}$ (at row i and column j of the matrix) reflects the normalized intensity of pheromones pointing from node i toward node j.

The pheromone value can either increase, as the robots deposit virtual pheromone on the sensor nodes they use, or decrease, due to pheromone evaporation. A increased pheromone value on a sensor nodes indicates that many ants have used that node and produced a very good solution .This increases the probability that this node will be used again by future ants to end up in a optimal solution. To avoid the rapid convergence of the algorithm toward a suboptimal region, we incorporate the pheromone evaporation phenomenon which allows the exploration of new areas of the search space.

In the traditional path planning algorithms, since there is no sensor network, each robot has to store the entire map of the environment(i.e location of the obstacles, target, possible paths etc) in its memory . The entire probability matrix is stored in the robot's memory and updated for each iteration. Since the objective of our method is to reduce the memory overhead of the robots, we store each row of the probability matrix on each static sensor node in a distributed way .For each iteration, each sensor node updates each row of the probability matrix, reducing the overall computational overhead. The probability for choosing a node depends on the pheromone values stored on the individual static sensor nodes and does depend on the robot's memory.

Let the initial probability distribution be P_0. The probability distribution in each iteration is altered in an attempt to increase the probability of generating shortest path solutions after each iteration. If P_m is the probability distribution in the current iteration and P_n is the probability distribution in the next iteration,

$$P_n = P_m \; f \qquad \text{-----(4)}$$

where f is the objective function as given in (2). After k iterations, we obtain , $P_n = P_m\ (f)^k$.As k -> ∞ , P_n would converge to a probability distribution which gives the optimal solution, S*. The algorithm convergences when the difference between the distributions P_m and P_n is minimized using cross entropy given as

$$\min D(P_m \| P_n) = \min \left(P_m(i) \log \frac{P_m(i)}{P_n(i)} \right) \qquad \text{------- (8)}$$

$$= \min(P_m(i) \log P_m(i) - P_m(i) \log P_n(i))$$

The first term gives the entropy for the probability distribution $P_m(i)$ and the second term gives the cross entropy between the probability distributions $P_m(i)$ and $P_n(i)$.The algorithm continues to iterate till the difference between P_m and P_n is very minimum (i..e between 0.1 to 0.01). At this point, the algorithm converges and the probability of obtaining the optimal solution is maximized.

4. Hardware Implementation of the Prototype

In our work ,we use six xbee devices to form a mesh network- four of which is used for the static nodes and two are used for the mobile robot. The robots are configured as the end device and one static node acts as the coordinator(target) to provide network synchronization. The other static nodes act as routers to relay the data packets in the network. Each of these nodes has a microcontroller to get the RSSI value of the target and calculate the corresponding distance value to form the pseudo-gradient. The robot uses these distance values to navigate to the target.

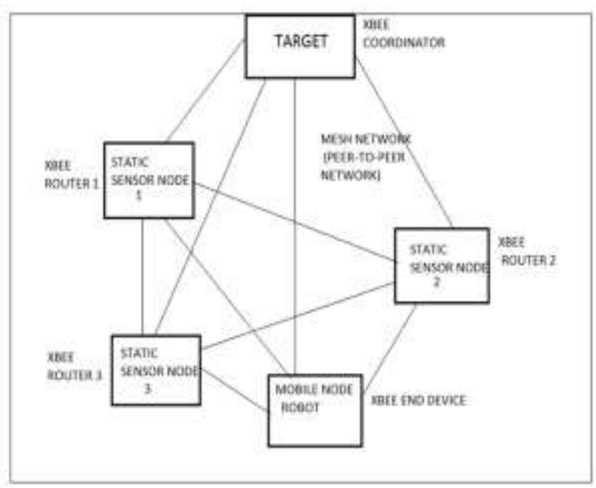


Figure 6 Block diagram for hardware implementation

The XBee uses Zigbee protocol(IEEE802.15.4) as the communication medium between the mobile robots and the static nodes. The XBee 2mW Chip Antenna series1 has been used in this project. It uses 3.3 V DC supply,which can be obtained by feeding the input voltage 5V to the voltage regulator LM317. It has a built in UART interface, which makes it easy to interface with the PIC microcontroller. Transmit power output is rated at 1mW with an operating frequency of 2.5GHz with operating current around 45-50 mA and RF data rate of 250k bps. The XBEE modules are configured using the X-CTU software,which is a Windows-based application provided by Digi. This program was designed to interact with the firmware files found on Digi's RF products and to provide a simple-to-use graphical user interface to them.

The PIC 16F877A microcontroller is configured to operate on 8 Mhz frequency using an external oscillator. It is programmed using MPLAB IDE with HITECH C compiler with the RISC instruction set.

The robot moves using metal gear wheels powered by DC motors.The DC motors operate using a H bridge which is an electronic circuit that enables a voltage to be applied across a load in either direction,to enable forward and backward movement.

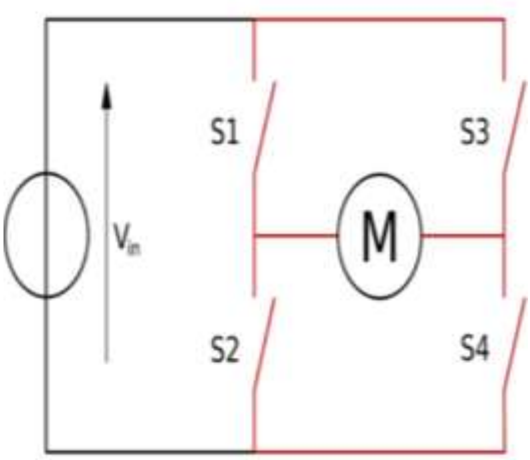


Figure 7 H-Bridge Circuit

The term H bridge is derived from the typical graphical representation of such a circuit. An H bridge is built with four switches . When the switches S1 and S4 (according to the first figure) are closed (and S2 and S3 are open) a positive voltage will be applied across the motor. By opening S1 and S4 switches and closing S2 and S3 switches, this voltage is reversed, allowing reverse operation of the motor.

The H-bridge arrangement is generally used to reverse the polarity/direction of the motor, but can also be used to 'brake' the motor, where the motor comes to a sudden stop, as the motor's terminals are shorted, or to let the motor 'free run' to a stop, as the motor is effectively disconnected from the circuit.The H-Bridge is constructed using 4 relays and its operating sequence is given in the table below.

Table 1 Operation of H-Bridge

S1	S2	S3	S4	Result
1	0	0	1	Motor moves right
0	1	1	0	Motor moves left
0	0	0	0	Motor free runs
1	0	1	0	Motor brakes

5. Results And Discussion

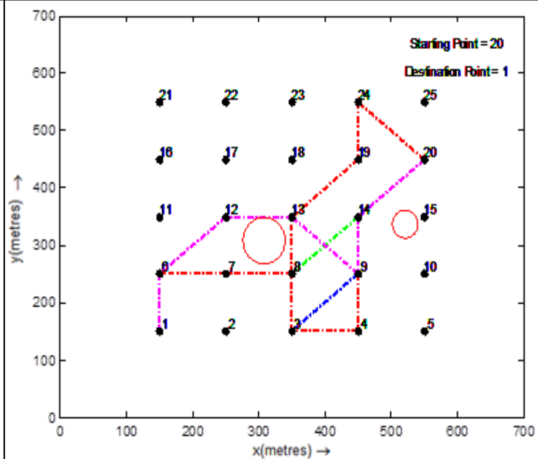
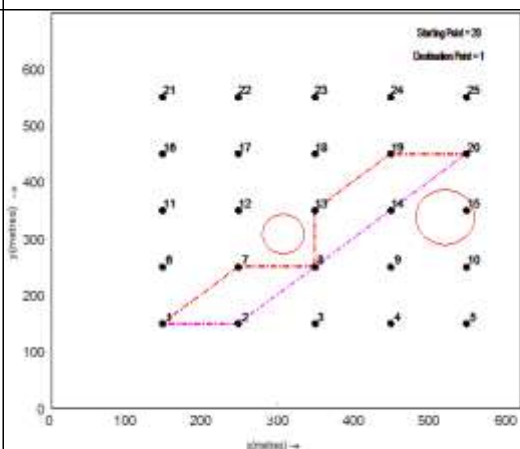
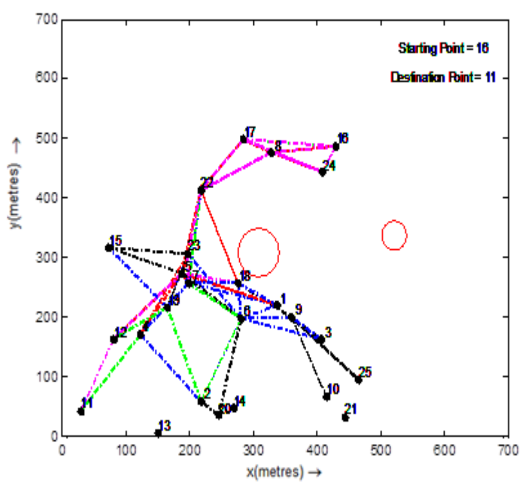
We consider the grid and random deployment of 25 sensors in the square field of 600 x 600 meters.

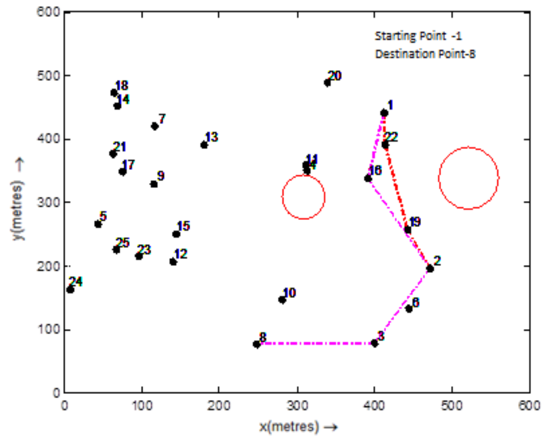
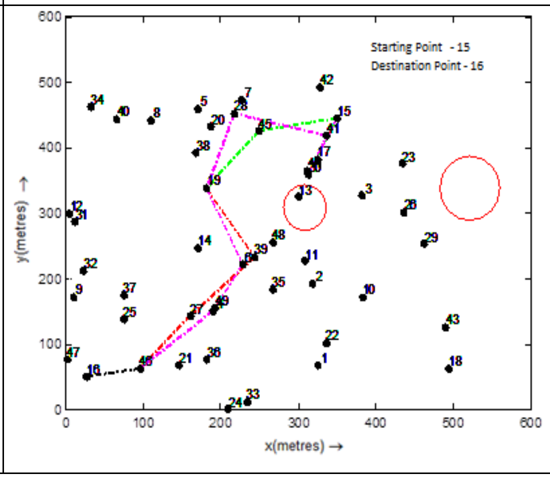
The following parameters are analyzed :

1. Travel-distance ratio, which is the ratio of the average distance traveled by the robot, to the Euclidean distance (theoretical shortest path) between its start node and the target node.
2. Probability of generating optimal solution
3. Convergence of the algorithm to optimal solution

To analyze the travel-distance ratio for the following cases given in Table 1, we specify the start point and target for the robots in a WSN environment with grid-based and random deployment of 25 sensors. We simulate the D-ACO algorithm and the conventional ACO to find the shortest path and compare the performance of the algorithms in the same scenario.

Table 2 Performance comparison of the travel-distance ratio between conventional ACO and D-ACO

S.No	Algorithm	Final trajectories generated using the algorithm	Average distance travelled	Euclidean distance	Travel-distance ratio
1.	Conventional ACO Algorithm in grid pattern (25 nodes)		800m	500 m	1.6
2.	Directed- ACO Algorithm in grid pattern (25 nodes)		572m	500m	1.14
3.	Conventional ACO Algorithm in random deployment (25 nodes)		1130m	630 m	1.8

4.	Directed ACO Algorithm in random deployment (25 nodes)		562m	394m	1.5
5.	Directed ACO Algorithm in random deployment (50 nodes)		800m	500m	1.6

Since there is no target-directed pseudo-gradient in the conventional ACO algorithm(case 1 of Table 1), it is observed that the robots search in all directions to find the target. This increases the average distance travelled by the robots .But in D-ACO algorithm(case 2 of Table 1), we use the static sensors which directs the robots towards the target using the pseudo-gradient. The robots find a shorter path because of the directed search which restricts the wandering of ants in all directions. In this way, the search area is minimized and hence the computational overhead in the robots is reduced. The travel-distance ratio is minimized using the D-ACO algorithm The circles in the figure denote the obstacles.It is observed that the robots plan a path to the destination avoiding the obstacles.

Similarly we analyze the travel-distance ratio for robots in random deployment of 25 sensor nodes. We specify the start point and target for the robots and simulate the conventional ACO (case 3 of Table 1) and D-ACO algorithm (case 4 of Table 1) to find the shortest path and compare the results .We infer that D-ACO algorithm gives a near-optimal solution to the shortest path problem by reducing the travel-distance ratio. This is because of the target-directed pseudo-gradient which directs the robot to the target.

On comparing the performance of the D-ACO algorithm in grid (case 2 of Table 1) and random(case 4) topology, we find that the travel-distance ratio is lesser using grid deployment of sensors. This is because the grid topology has a regular, well-defined pattern for the deployment of sensors. However, when the number of nodes is increased in random pattern ,we observe that the travel- distance ratio is decreased. It is observed that the D-ACO algorithm performs better with 50 randomly deployed nodes (case 5) than with 25 nodes(case 4),by minimizing the travel-distance ratio. This because as the number of nodes increases ,the number of waypoints to guide the robot to the target increases. The robots can make a intelligent decision to find the shortest path to the target.

To analyze the probability of generating the optimal solution in D-ACO algorithm, the visual representation of the probability matrix is shown in the Fig 6 . It is based on the probabilistic model in Section 3. We represent the pheromone levels on the nodes as the probability distribution values .Initially ,since the probability is equally distributed among all nodes in the neighborhood, the ants perform a unbiased search in the solution space by searching for all possible paths to the target.

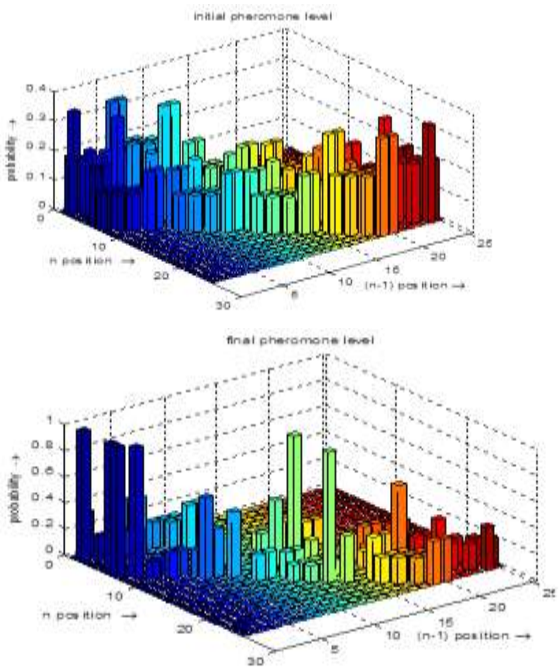


Figure 6. Visual representation of the initial and final probability distribution

As the algorithm iterates we observe that the values of the initial probability distribution are updated as shown in Fig 7 based on the solutions generated by the ants, as explained in MBS (model-based search approach) in Section 3. For the path- finding problem given in case 2 of Table 1, the visual representation of the final probability distribution is given in Fig 6.The algorithm converges to an optimal solution (i..e) shortest path given by the nodes 20 --> 14 --> 8-->2-->1 and 20--> 19 --> 13--> 8 -->7-->1 These nodes get high pheromone level and most of the ants follow the shortest path. The other longer paths found in the initial stages of the algorithm are discarded. This biases the search space of the future ants to search only in the regions of high pheromone levels (high probability distribution). From the final probability distribution ,we observe that the probability to choose the optimal solution (i..e nodes 20, 14, 19,8,2,1) increases to values between 0.8 and 1. Several nodes have pheromone values close to zero, making a selection of these nodes very unlikely. This explains the reinforcement of pheromones on the elite path and evaporation of pheromones on the longer paths, as explained in the previous section. Thus we maximize the probability of generating good solutions

We analyze the rate of convergence of the D-ACO algorithm by comparing it with conventional algorithm.

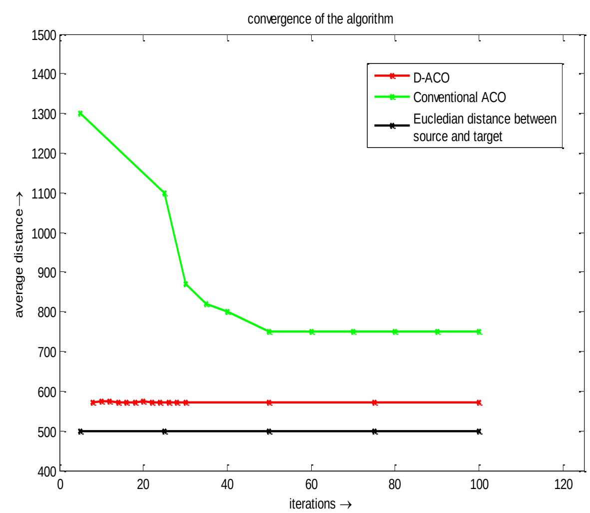


Figure 7. Convergence of the algorithms in grid-based deployment of 25 sensor nodes

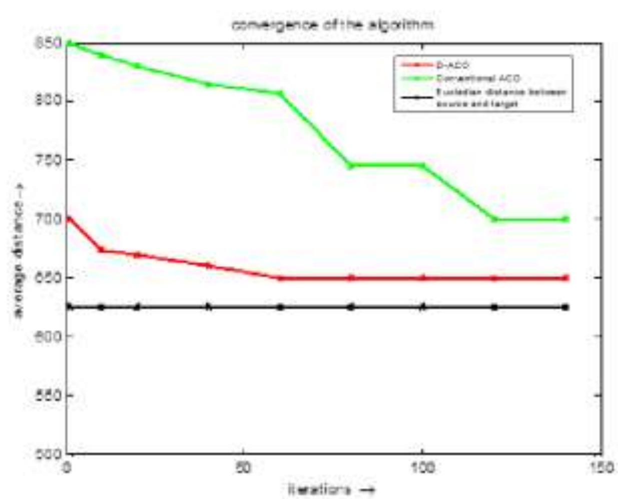


Figure 8 Convergence of the algorithms in grid-based deployment of 36 sensor nodes

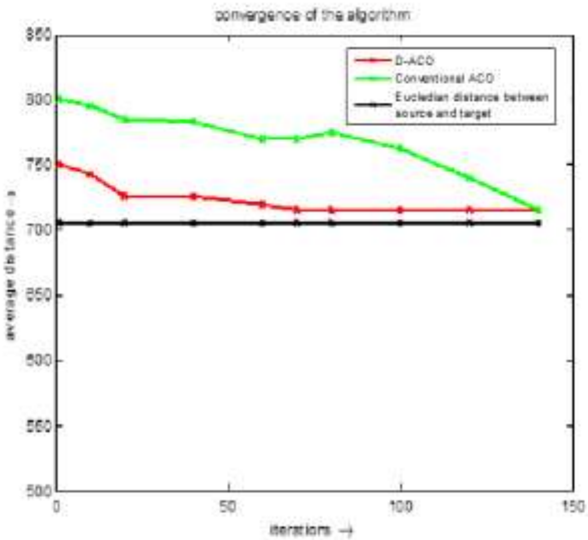


Figure 9 Convergence of the algorithms in grid-based deployment of 50 sensor nodes

In a grid topology with 25 nodes(Fig 7), the conventional ACO algorithm converges in 50 iterations to the optimal solution. But the D-ACO algorithm converges to the optimal solution in 20 iterations. It takes nearly half the time to converge when compared with conventional algorithm. From Fig 7 ,it is observed that the initial average distance does not vary greatly from the final average distance. Since grid layout allows uniform and regular coverage of the region, the robots easily find the shortest path from source to target in the first few iterations, which in turn reduces the computational overhead for the robots. As we increase the number of nodes to 36 nodes (Fig 8), it is observed that the D-ACO algorithm converges a near optimal solution than with 25 nodes. The difference between the theoretical shortest path and simulation results is about 25 meters. As we further increase the number of nodes to 50 nodes (Fig 9), the simulation results almost match the theoretical results with a difference of less than 10 meters. With 36 nodes and 50 nodes, the convergence time increases to 60 iterations and 70 iterations respectively. In both cases (Fig 8 and Fig 9), it is seen that the D-ACO algorithm out-performs the conventional ACO algorithm by decreasing the average distance travelled by the robots .

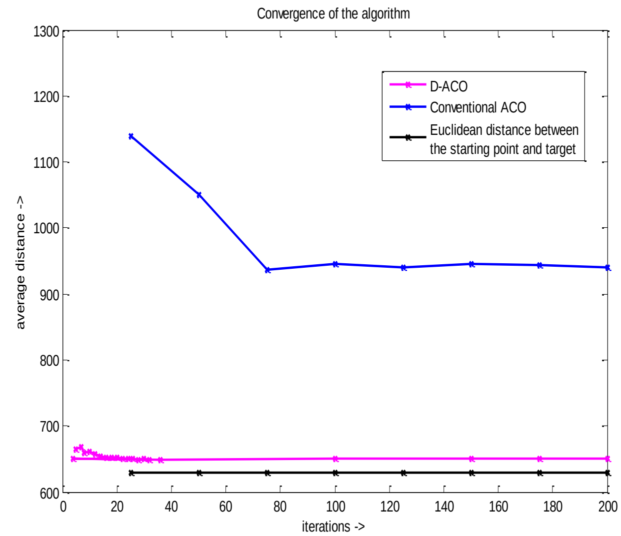


Figure 10. Convergence of the algorithms in random deployment of 25 sensor nodes

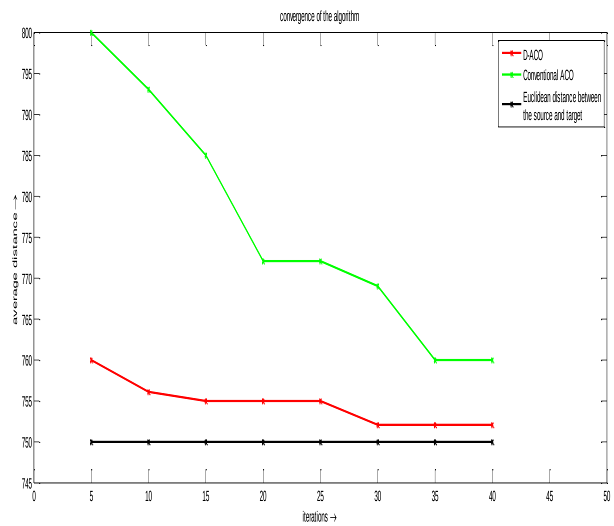


Figure 11 Convergence of the algorithms in random deployment of 36 sensor nodes

In random topology with 25 static sensor nodes (Fig 10), the conventional ACO algorithm converges in 80 iterations to the optimal solution .But ,the D-ACO algorithm converges to the optimal solution in 20 iterations , increasing the rate of convergence by 4 times. As we increase the number of nodes to 36 nodes (Fig 11) ,the D-ACO algorithm finds a near optimal solution in the initial iterations due to the target-directed pseudo-gradient.

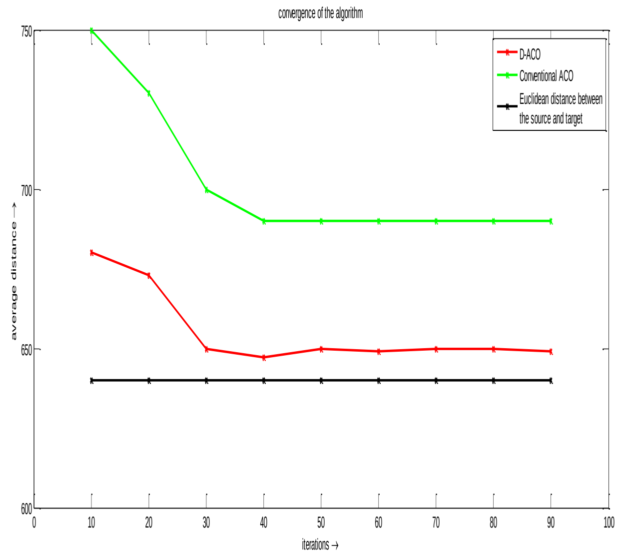


Figure 12 Convergence of the algorithms in random deployment of 50 sensor nodes

As the number of nodes is increased to 50 nodes, it is observed that the D-ACO algorithm converges faster to a near optimal solution than the conventional ACO algorithm with minimum distance travelled.

6.Conclusion

The WSN-aided robot navigation scheme has been implemented using Directed Ant Colony Optimization Algorithm. By comparing the conventional ACO with the D-ACO algorithm ,we observe that the rate of convergence in D-ACO algorithm is increased by two times in grid deployment, and by five times in random deployment. By restricting the direction of the search in D-ACO Algorithm, we obtain the shorter path than conventional ACO algorithm. By implementing the D-ACO algorithm in grid and random deployment of sensor nodes, we observe that the algorithm finds the shortest path in grid deployment due to its regular and uniform pattern.

The future work involves clustering of the static sensor nodes. The cluster head has the position information about all the nodes in that cluster. The mobile robot communicates with the cluster head to find the path to the target .This reduces the energy consumption as only the cluster heads are active all the time to provide information to the robot .

References

[1] Jung J.H., Park S. and Kim S.L. ‘ Multi-Robot Path Finding with Wireless Multihop Communications’, IEEE Communication Magazine,Vol.13 No.6 ,2010 ,pp. 421-425

[2] Zhang Z., Li Z. Dawei and Chen J. ‘Path Planning and Navigation for Mobile Robots in a Hybrid Sensor Network Without Prior Location Information’, International Journal of Advanced Robotic Systems, Vol. 10. No.3 ,2013 , pp. 44-48

[3] Shen W. and Li W. ‘Swarm Behavior Control Of Mobile Multi-Robots With Wireless Sensor Networks’ Journal of network and computer applications, Vol 14. No. 1,2011, pp. 314-318

[4] Nan Z., Zhao X. and Min T. ‘RSSI Based Mobile Robot Navigation In Grid Pattern Wireless Sensor Network’, IEEE Communication Magazine, Vol 2. No.10, 2010 ,pp.213-216

[5] Qun L., Rosa M. and Daniela R., ‘Distributed Algorithms for Guiding Navigation across a Sensor Network’, International Journal of Networking and communication, Vol .8 No.2,2013, pp 86-90.

[6] Jiang J.R., Lai Y.L. and Deng F.C. ‘Mobile Robot Coordination And Navigation With Directional Antennas In Positionless Wireless Sensor Networks’, International Journal of Ad Hoc and Ubiquitous Computing, Vol.12 . No.7,2011 , pp 67-70.

[7] Nicholas C., Manning M. and Nokleby S. , ‘Dynamic Robot Path Planning Using Improved Max-Min Ant Colony Optimisation’, Proceedings of the International Conference of Control, Dynamic Systems, and Robotics, Paper No. 49,2014, pp 15-19.

[8] Chaari I., Koubaa A., Bennaceur H., Trigui S. and Al-Shalfan. K, ‘Smart Path: A Hybrid ACO-GA Algorithm For Robot Path Planning’,Proceedings of the Evolutionary Computation IEEE Congress,2012, pp.1-8.

[9] Rai R. and Singh T. ‘A Hybrid Framework for Robot Path Planning and Navigation using ACO and Dijkstra’s Algorithm’,Proceedings of the International Symposium on Devices MEMS, Intelligent Systems & Communication (ISDMISC),2011, pp 13-16.

[10] Simek M. and Moravek P. ‘Modeling of Energy Consumption of Zigbee Devices in Matlab Tool’ ,Electrorevenu Vol. 2, No. 3,2011, pp 122-126

[11] Zar S. Chi, Hlaing Z.C. and Khine M. 'Solving Traveling Salesman Problem by Using Improved Ant Colony Optimization Algorithm', International Journal of Information and Education Technology, Vol. 1, No.5,2011, pp 56-62

[12] Shweta K.,Fatemeh P. Khosravi , Fardad F. and Reza S. Nadooshan 'An Experimental Study of Ant System for Solving Travelling Salesman Problem', International Journal of Emerging Technology and Advanced Engineering, Vol. 3, No. 7,2013, pp 34-39.

[13] Kuo-Sheng H., Shun F. and Jung Z. Lee 'Improving Ant Colony Optimization Algorithms for Solving Traveling Salesman Problems', Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol.11 No.4,2007pp 127- 130.

[14] Englot H., Brendan J., and Franz S. Hover (2011) 'Multi-Goal Feasible Path Planning Using Ant Colony Optimization.”Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) pp. 2255 - 2260.

[15] Mei H., Tian Y. and Linan Z.(2006) 'A Hybrid Ant Colony Optimization Algorithm for Path Planning of Robot in Dynamic Environment' International Journal of Information Technology, Vol.12, No.3, pp 4-9.