A Hybrid Framework for Robot Path Planning and Navigation using ACO and Dijkstra's Algorithm

Rebika Rai
Department of Computer Science & Engineering
Sikkim Manipal Institute of Technology, Majitar, Sikkim

Tejbanta Singh Chinghtam
Department of Computer Science & Engineering
Sikkim Manipal Institute of Technology, Majitar, Sikkim

ABSTRACT

The social insect metaphor for solving problems has become an emerging topic in the recent years. This approach emphasizes on direct or indirect interactions among simple agents. Swarm Intelligence offers an alternative way of designing intelligent systems. This paper explores the behavior of a group of mobile agents or robots to find the shortest path between the food (destination) and nest (source), without any visible, central, active coordination mechanism. Feedback by the agent during traversal of the path causes more agent concentration on the path, thereby influencing the behavior of the other agents and the indirect communication allows the agent to modify their environment to influence the behavior of other agents. Several obstacles are likely to be encountered in the course of this traversal. The objective of the agent is to find an appropriate and an optimize solution to bring itself closer to the goal considering the cost, time and path availability. A typical case of Traveling Salesman Problem (TSP) is incorporated to achieve this navigation problem wherein, an agent plans a route through a number of nodes and each node or location is only visited once with the agent returning back to city of origin. The Ant Colony Optimization (ACO) is a popular approach that searches for an optimal solution in a given set of solutions. Dijkstra's Algorithm is an approach to find the shortest route between two locations. This paper addresses method of path finding problem using this two different approaches.

General Terms

Ant Colony Optimization, Dijkstra's Algorithm, Swarm Intelligence, Traveling Salesman Problem.

Keywords

Pheromone, Optimized path, Navigation, Path planning, TSP, ACO

1. INTRODUCTION

The Traveling Salesman problem is a real life business problem. It is usually described as planning a route through a number of cities such that each city is only visited once and with the salesman returning to his city of origin. The problem gets more complicated when the number of cities grows and a visual inspection no longer provides an obvious solution, which have historically required intensive amount of time, traveling cost and labor. Some improvements have been made recently due to introduction of new algorithms to solve it, which on the other hand is still costly for larger areas (as the number of cities that is to be traveled increases). In this regard the need of reducing the overall cost effectiveness is of paramount importance. Swarm intelligence (SI) is based on the collective behavior in decentralized [12], self-organized systems. It's typically made up of a population of simple agents interacting locally with one another and with their environment. Though there is no centralized control structure dictating how individual agents should behave, local interactions between such agents lead to the emergence of global behavior. Natural examples of SI include ant colonies, bird flocking [4], animal herding, bacterial growth, and fish schooling.

Figure 1: Ants moving from nest (source) towards its food (Destination) [16].

Ants communicate with each other using pheromones. In species that forage in groups, a forager that finds food marks a trail on the way back to the colony; this trail is followed by other ants, these ants then reinforce the trail when they head back with food to the colony. When the food source is exhausted, no new trails are marked by returning ants and the scent slowly dissipates. This behavior helps ants deal with changes in their environment. For instance, when an established path to a food source is blocked by an obstacle, the foragers leave the path to explore new routes. If an ant [8] is successful, it leaves a new trail marking the shortest route on its return. Successful trails are followed by more ants, reinforcing better routes and gradually finding the best path. The Ant Colony Optimization algorithm is one among the few other approaches to solve this problem.

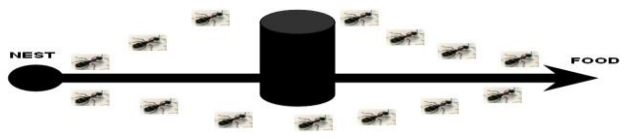

Figure 2: An obstacle placed on the way between nest and food[16].

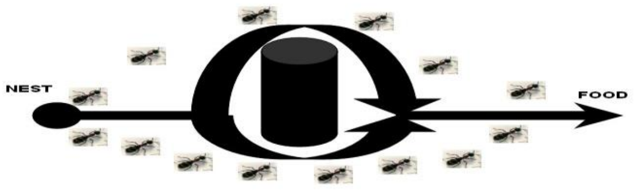

Figure 3: Ants randomly choosing the path[16].

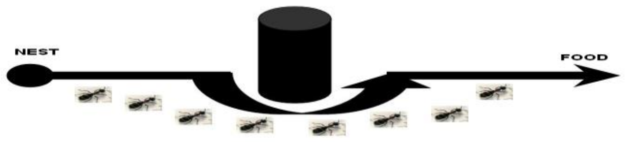

Figure 4: Shortest path choosen by maximum ants based on pheromone deposits[16].

Each ant drops a quantity of artificial pheromone [1] on every point that the ant passes through. This pheromone simply changes the probability that the next ant becomes attracted to a particular path. Dijkstra's Algorithm solves the single-source shortest path problem for a directed graph with a non-negative edge weights. For instance, if the vertices of the graph represent cities and the edge weights represent driving distance between pairs of cities connect by a direct road; Dijkstra's Algorithm can be used to find the shortest route between two cities.

In the TSP a set of locations (cities) and the distances between them are given. The problem consists of finding a path of minimal length that visits each city once and only once. To apply ACO to the TSP, we consider the graph defined by connecting the set of cities with the set of vertices of the graph. This graph is known as *construction graph* [2]. Since in the TSP it is possible to move from any given city to any other city, the graph is fully connected and the number of vertices is equal to the number of cities. The lengths of the edges between the vertices is set to be proportional to the distances between the cities represented by these vertices and pheromone values are associated and heuristic values with the edges of the graph. Pheromone values can be modified at runtime and represent the cumulated experience of the ant colony, while heuristic values are problem dependent values that, in the case of the TSP, are set to be the inverse of the lengths of the edges. The solution is constructed as follows. Each ant starts from a randomly selected city. Then, at each step it moves along the edges of the graph. Each ant keeps a memory [3] of its path, and in subsequent steps it chooses among the edges that do not lead to vertices that it has already visited (TSP). A solution is said to be constructed once it has visited all the vertices of the graph. At each construction step, an ant probabilistically chooses the edge to follow among those that lead to yet unvisited vertices. The probabilistic rule is biased by pheromone values and heuristic information: the higher the pheromone and the heuristic value associated to an edge, the higher the probability an ant will choose that particular edge. Once the tour is completed by all the ants or agents, the pheromone on the edges is updated. Each of the pheromone values is initially decreased by a certain percentage as the pheromone is a chemical that dissipates with time when the food source is exhausted and no new trails are marked by returning ants. Each edge then receives an amount of additional pheromone proportional to the quality of the solutions to which it belongs. This procedure is repeatedly applied until a termination criterion is satisfied.

2. PROBLEM DEFINITION

The Traveling Salesman problem is a popular illustration of the limits of normal computational techniques and equally a popular subject for demonstrations of alternate approaches to solving such challenges. TSP can be represented by a complete weighted graph G = (N, A) with N being the set of n = |N| nodes (cities), A being the sets of arc fully connecting the nodes. Each arc (x, y) Є A is assigned a weight d_{xy} which represents the distance between cities x and y. The graph that may be considered can be symmetric or Asymmetric. Asymmetric TSP is considered where the distances between the cities are independent of the direction of traversing the arc where $d_{xy}=d_{yx}$ for every pair of nodes. Group of mobile agents or robots begins its journey from a source and heads towards its destination. On its traversal several obstacles are likely to be encountered. On encountering of obstacles, the agent needs to make a decision related to the path selection i.e. agent might try overcoming the obstacles by following the same path or diverts itself by back-tracking and searches for another path taking into account that every nodes have been visited at least once and the robot does not traverse the already visited nodes and finally comes back to its origin from where it had begun its journey. Traveling Salesman Problem is incorporated to achieve this navigation problem.

3. PROPOSED SOLUTION

The behavior of the mobile agents or robots may vary drastically depending on the algorithm used. The robots when placed at a particular node/location may not be able to determine the solution path that may bring them close to the goal considering the cost time and path availability. Two algorithms are used for the implementation on the robotic test bed i.e. Dijkstra's algorithm to find the shortest path and Ant Colony Optimization to find the optimal solution from given set of solutions to solve the problems faced by Travelling Salesman.

The Dijkstra's Algorithm works by keeping for each vertex ***v*** the cost ***d[v]***, of the shortest path found between source ***s*** and vertex ***v***. Initially, the value of ***d[v]*** is zero for ***s*** (***d[s] =0***) and infinity for all other vertices. This represents the fact that robot is unknown of the path leading to those vertices. The basic operation of this algorithm is edge relaxation which means that if there an edge from vertex u to vertex v, then the shortest known path from ***s*** to ***u***, ***d[u]*** can be extended to a path from ***s*** to ***v*** by adding ***edge(u,v)*** at the end. The resultant path will have the length ***d[u] + w(u,v)***. If the resultant value is less than the current ***d[v]***, current value can be replaced with the resultant value. When algorithm finishes, ***d[v]*** will be the cost of the shortest path from s to v or infinity, if no such path exists. The robots find the appropriate solution path and bring itself closer to the goal where the already visited node is not visited again by the robot.

In ACO Algorithm, each edge is assigned a small random value to indicate the initial pheromone concentration, τ_{ij} (0). An ant randomly selects which edge to follow next. Place *n* number of ants on the origin node. Next node is selected based on the probability equation [4], until destination node is reached and is defined as:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t)}{\sum_{j\in\mathcal{N}_i^k}\tau_{ij}^\alpha(t)} & \text{if } j \in \mathcal{N}_i^k \\ 0 & \text{if } j \notin \mathcal{N}_i^k \end{cases}$$

Where $N^k{}_{i,}$ is the set of feasible nodes connected to node i, with respect to ant k. If, for any node i and ant k, $N^k{}_{I=\Phi,}$ the predecessor to node i is included in $N^k{}_{i.}$ Pheromone amount is deposited to each link, (*i*, *j*), of the corresponding path.

$$\Delta\tau_{ij}^k(t) \propto \frac{1}{L^k(t)}$$

Where, L^k (*t*) is the length of the path [4] constructed by ant *k* at time step *t*. That is,

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k}\Delta\tau_{ij}^k(t)$$

If ant k is currently located at node i, $n_{k,}$ is the number of ants. Gradually, Pheromone evaporation takes place so the pheromone deposited τ_{ij} (*t*) at a particular instant of time at each node has to be reduced as follows [4]:

$$\tau_{ij}(t) \leftarrow (1-\rho)\tau_{ij}(t)$$

The constant, *ρ*, specifies the rate at which pheromones evaporate and also *ρ* controls the influence of search history. For large values of *ρ*, pheromone evaporates rapidly, while small values of *ρ* result in slower evaporation rates. The more pheromones evaporate, the more random the search becomes, facilitating better exploration.

4. PROPOSED THEOREM

The simulation environment in Figure 5, 6, 7 and 8 below comprises of different number of nodes and the obstacles placed randomly in between. Initially, the robot is placed on the source node and when it begins its journey and heads towards its destination, on encountering obstacles [14], robot has to divert its direction and traverse through alternative path. [Here, assumption is made that the cost to overcome the obstacle is very high]. Referring to Figure 5, where there are four nodes and two obstacles, at least one possible solution path can be generated no matter the obstacles is placed in any order. On the other hand, in Figure 6, with four nodes and five obstacles, no solution path can be generated. Similarly, in figure 7 depicts that when there are five nodes and three obstacles, at least one solution path can be generated. Similarly, as depicted Figure 8, with five nodes and six obstacles, no solution path can be generated. This conclusion of the number of possible solution sets that can be generated is proved below:

If N is the total number of nodes, O denotes total number of obstacles .Then, total paths generated between every other nodes where all nodes are connected can be obtained as P = N (N-1) /2. The solution path comprises of N+1 connected node and N paths [16]. Therefore, if (P – O) > = N; at least one solution path is generated. Else, if (P - O) < N; there is no possible solution path.

In Figure 5, N=4, O=2, P= 4(4-1)/2 = 6. Here, P – O = 6 – 2 = 4 which implies P-O = N (4 = 3). Therefore, at least one solution path can be generated.

Similarly, referring Figure 6, where, N=4, O=5, P= 4(4-1) / 2 = 6. In this case P – O = 6 – 5 = 2 which implies P-O < N (2 < 4). Therefore, no solution path can be generated.

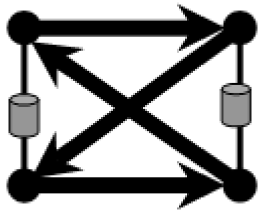
Figure 5: Four nodes with two obstacles.

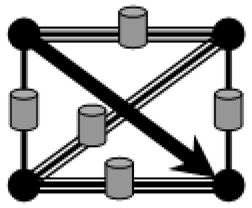
Figure 6: Four nodes with five obstacles.

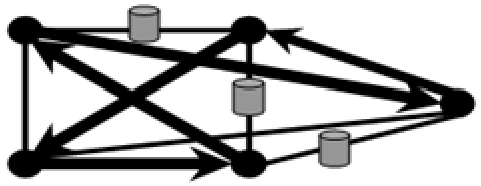
Figure 7: Five nodes with three obstacles.

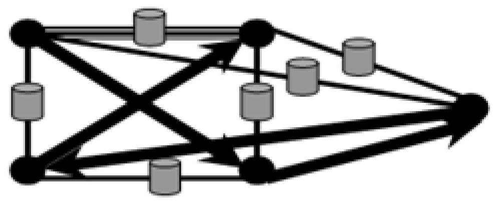

Figure 8: Five nodes with six obstacles.

Legends

Symbols	Meaning
●	Nodes
▮	Obstacles
➡	Direction of the solution path
═	Path that cannot be traversed

In Figure 7, N=5, O=3, P= 5(5-1)/2 = 10. Here, P – O = 10 – 3 = 7 which implies P-O = 7 > N (7 > 5). Therefore, at least one solution path can be generated.

Similarly, referring Figure 8, where, N=5, O=6, P= 5(5-1) / 2 = 10. In this case P – O = 10 – 6 = 4 which implies P-O < N (4 < 5). Therefore, no solution path can be generated.

A general theorem is devised to determine the availability of the solution path irrespective of the number of nodes, with all the nodes interconnected and randomly placed obstacles between the nodes, which prove helpful in determining about the solution path beforehand.

5. SIMULATION TEST BED

The simulation environment in Figure 9 and Figure 10 comprises of a robot placed at source node, in a complete weighted graph to generate a solution path with minimum cost to solve TSP using the Dijkstra’s and Ant Colony Optimization Algorithm. The robot is initially placed at node 1, which is the source node and the robot after traversing every other node has to finally reach the origin i.e. node 1. The minimum cost required to navigate from the source node to every other node such that all the intermediate nodes are visited are determined. Finally, comparison is made on the cost generated and minimum cost and the solution paths are generated.

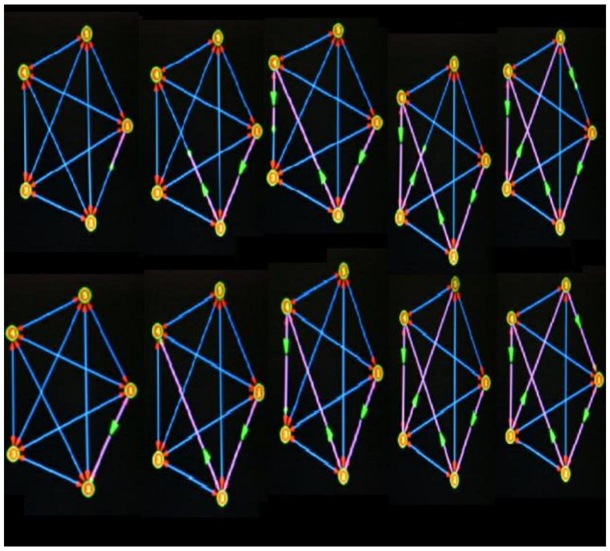

Figure 9: Simulation environment with five nodes highlighting the optimal path.

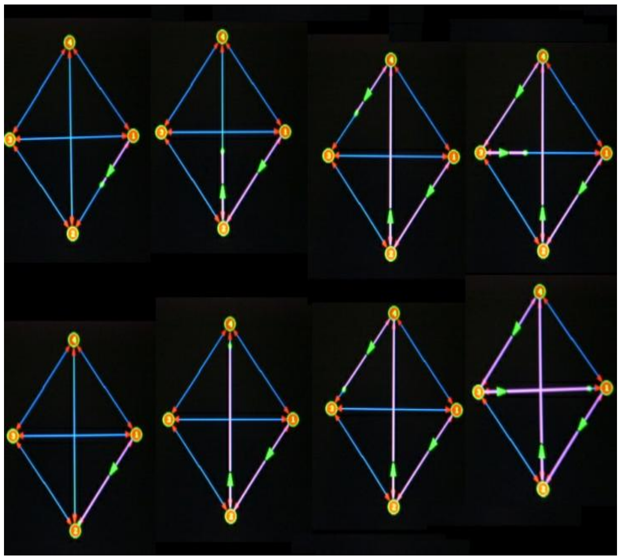

Figure 10: Simulation environment with four nodes highlighting the optimal path.

Legends

Symbols	Meaning and Equivalent colour in Simulation Environment
●	Nodes (yellow)
↔	Edges between nodes (Blue)
↔▶	Direction of the solution path (green)
↔▶	Solution path (Pink)

6. RESULTS

The main goal of navigation problem [10] is to generate a collision-free path from a starting position to an end or destination position. Figure 11 represents a Lego Mindstroms NXT robot deployed in the physical environment to traverse from source node towards its destination avoiding all the obstacles encountered on its way and finally reach the origin from where the robot had begun its journey (TSP). On the other hand, Figure 12 demonstrates two Lego Mindstroms NXT robots traversing from the source node and heading towards the destination where the second robot follows the first one.

However, navigation problem includes several difficult phases that need to be overcome, such as obstacle avoidance [9], position identification, and so forth. A navigation algorithm must be able to identify the current location of the robot, determine a path to the destination and avoid any collisions due to obstacles [13][15].

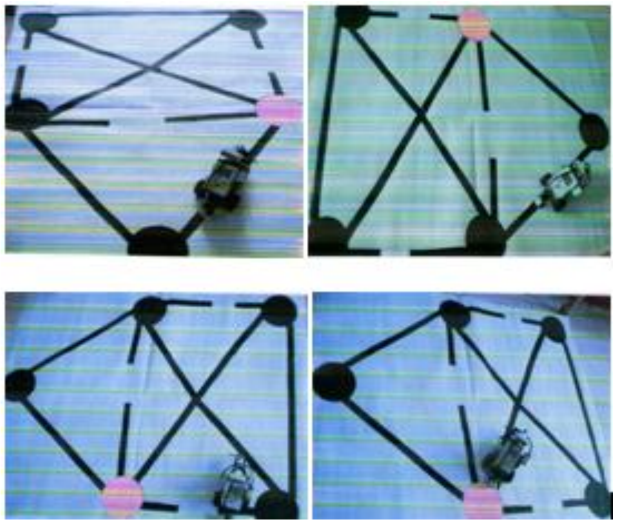

Figure 11: A Lego Mindstroms NXT robot deployed in the physical environment to traverse from source node towards its destination.

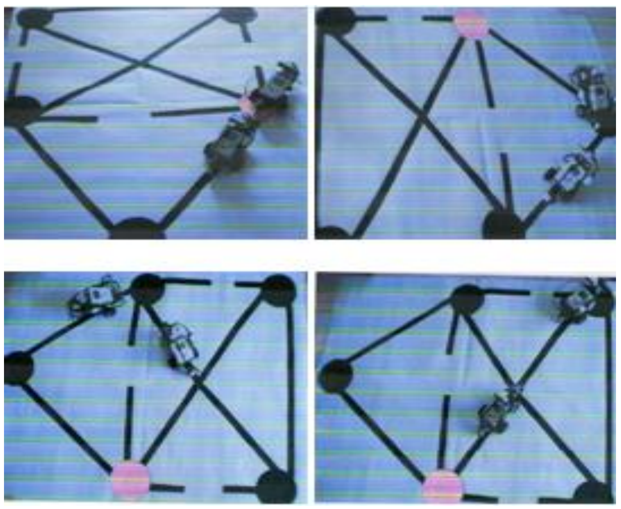

Figure 12: Two Lego Mindstroms NXT robots deployed in the physical environment to traverse from source node towards its destination with second robot following the first one.

Legends

Symbols	Meaning
●	Source / Destination node
●	Intermediate nodes
▬	Edges
	Lego Mindstroms NXT robot

7. CONCLUSION

The robots when placed at a particular node may not be able to determine the solution path that may bring them close to the goal considering the cost time and path availability. In this paper, Dijkstra’s algorithm is proposed to find shortest path and ACO algorithm in turn is used to find the optimal path from among the set of solution path to solve the Traveling Salesman problem and a theorem that helps to determine whether a solution path can be generated or not when there are several interconnected nodes and obstacles placed randomly in between these nodes. It is observed that instead of using individual computation technique such as Dijkstra’s algorithm or Ant Colony optimization algorithm, or may be other algorithms, hybrid model such as Dijkstra’s algorithm with ACO algorithm to solve some problem like TSP yields better results.

8. REFERENCES

[1]Ryan Ward," Optimizing Pheromone Modification for Dynamic Ant Algorithms", TJHSST Senior Research Project (June 12, 2007)

[2]L.M. Gambardella and M. Dorigo,"Ant-Q: A Reinforcement Learning Approach to the TSP", In Proceedings of Twelfth International Conference on Machine Learning, (1995).

[3]]L.M. Gambardella and M. Dorigo," Solving Symmetric and Asymmetric TSPs by Ant Colonies", In Proceedings of IEEE International Conference on Evolutionary Computation, pages 622–627, (1996).

[4]Marco Dorigo, Vittorio Maniezzo and Alberto Colorni,"The Ant System: Optimization by a colony of cooperating agents", In Proceedings of IEEE International Conference on Evolutionary Computation.

[5]A.P.Engelbrecht,"Computational Intelligence: An introduction", Second Edn John Wiley & Sons, Ltd, (2007)

[6]Roland Siegwart and Illah R. Nourbakhsh," Introduction to Autonomous Mobile Robots", First Edn, MIT press.

[7]Angeniol, B., Vaubois, G de L C. and Texier JYL.(1988) ,"Self-Organizing Feature Maps and the Traveling Salesman Problem , Neural Networks", Vol 1, pages289-293.

[8]Denis Darquennes, "Implementation and Applications of Ant Colony Algorithms", Faculties Universitaires Notre-Dame de la Paix, Namur, Institute of Informatique Annee acad_emique (2004-2005)

[9]Gianni Di Caro, "Ant Colony Optimization and its Application to Adaptive Routing in Telecommunication Networks", Bruxelles, September (2004)

[10]Beni, G., Wang, J.,"Swarm Intelligence in Cellular Robotic Systems", Proceed. NATOAdvanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26–30 (1989).

[11]Ethan J. Tira-Thompson, Neil S. Halelamien, Jordan J. Wales, and David S. Touretzky," Cognitive Primitives for Mobile Robots", Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15123-3891

[12]Yang Liu and Kavin M. Passino,"Swarm Intelligence: Literature review", Deptt. Of Electrical Engineering, The Ohio state University, Columbus, March 30 (2000).

[13]Kyung min Han and Dr. Robert W. McLaren," Collision Free Path Planning Algorithms for Robot Navigation Problem", University of Missouri-Columbia, August (2007).

[14]Luo Xiong, Fan Xiao-ping, Yi Sheng, Zhang Heng," A novel Genetic Algorithm for Robot Path Planning in Environment Containing Large Numbers of Irregular Obstacles",Vol.26, No.1, January(2004).

[15] M. Youself Ibrahim and Allwyn Fernandes, "Study on Mobile Robot Navigation Techniques," IEEE ICIT, Tunisia, December 8-10, (2004).

[16]Rebika Rai, Tejbanta Singh Chingtham, M.K.Ghose, " Optimization of Autonomous Multi-Robot Path Planning & Navigation using Swarm Intelligence", In National Conference on LEAN Manufacturing Implementations : The future of Process Industries (LEMAN '2009)" , (11-12, September 2009).