An Ant Colony System: Algorithm for Multi-Robot Path Planning and Optimization

Savita Yadav[1], Deependra Rastogi[2]

[1] *Student, B.Sc (CS), CCSIT, TMU, Moradabad*

[2] *Assistant Professor, CCSIT, TMU, Moradabad*

[1]saveetayadav671996@gmail.com

[2]deependra.computers@tmu.ac.in

Abstract— A distinction of Ant Colony System (ACS) is signified in this paper and pragmatic for the Robot track arrangement drive. The procedure demonstrations a new technique to catch the simplest path from source to endpoint in disconnected mode with the submission of in-build path drawing by succeeding the Robot Path Algorithm (RPA), presented in this paper. Robot continuously follows the path plan delivered to it. To find the direct path as well as it can attain the information that in which technique, i.e. from one bulge to the next bulge, it will have to travel. The drive of the robots is grounded on the movent practice of the ants in the ant cluster. Among all the procedures for definition the shortest path, the planned Shortest Path Procedure (SPA), based on Kruskal algorithm, is much more operative and precise for the RPP problem and will take less computational time and hence upsurge the competence of the work procedure of the robot system.

Keywords— **Ant Colony System, Pheromone, Shortest path, Kruskal's algorithm, two bridge experiment, Node connectivity database.**

I. . INTRODUCTION

In Ant Colony System (ACS), the ants search the food by following some typical procedures. The ants realize about the food source by smelling it and follow the smell to reach to the food source. The ants use to follow the chemical trail formed by their pheromone on that moving path. They also use the technique to avoid or overcome the obstacles on their path. The newer approach of the ant system is presented in this paper to solve the Robot Path Planning (RPP) problems [1, 2]. The Modified Ant Colony Algorithm (MACA), with the application of Kruskal's Algorithm [3], is capable of finding the shortest path efficiently with the implementation of Path Map (PM). In this paper, the path planning for the robots is introduced such that they can find and follow their path towards the targeted location without any human intervention[4][5] but by using the PM which is actually the predefined graph that includes all the definitions of the path of the area where the robot colony is located. The PM is basically installed in the robot's memory and robot follow that PM while moving towards the destination. The main similarity between the ant system and the robot system is that the ants can smell of the food and the robot can sense for the path to reach to the desired location.

II. ANT COLONY SYSTEM

The ants reach to the food source by following the shortest path. It was found by applying the double bridge experiment where two path of different lengths are connected from the nest of the ants to the food source. It was found that the Argentine ants, though they cannot see very well, can find the shortest path after sometimes, because they do not practically use the vision technique to find the shortest path rather they use to smell the concentration of the pheromone on the path and finally they can find the shortest path easily [6, 7, 8, 9]. The double bridge experiment is shown in Figure 1.

2.1 Ant behavior

The ants places the pheromone on the path while moving from nest to food source. In

Figure 1, it is clear that the ants after sometimes can find the shortest path by realizing the pheromone concentration on the path. In double bridge experiment, one path is shorter than another. When ants stat their journey to the food source (Figure 1(a)), the find two different paths and divide into two groups as because the probability of moving in each path is 0.5 for the first time. Naturally, the ants on the shortest path come back to the nest faster than the ants in longest path. That means, the pheromone concentration will be much higher in the shortest path with compared to the longest path. Then after a short time the ant follow the higher pheromone concentrated path (Figure 1(b)). In this way, at last most of the ants follow the highest pheromone concentrated path i.e. the shortest path [9, 10, 11] (Figure 1(c)).

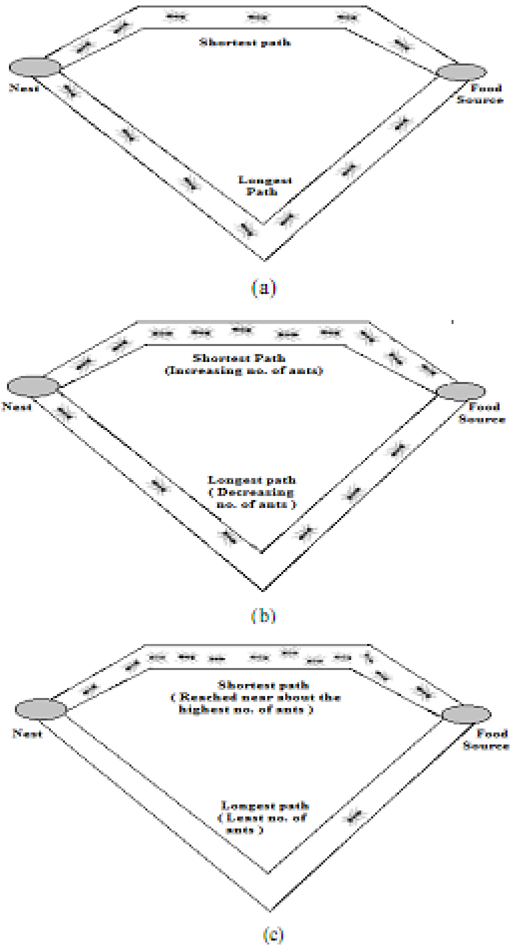

Figure 1. Double bridge experiment: (a) Ants starts their journey (t=0); (b) Ants find the shortest path(t=1); (c) Ants follow the shortest path and accumulate on that path(t=2)

2.2 Algorithm for shortest path

The Modified Ant Colony Algorithm (MACA) for the Ant Colony System is shown in Figure 2.

This algorithm is very efficient and helpful in the implementation of Robot Colony System (RCS) and its routing path. For this implementation, the Kruskal's Algorithm is essentially needed to calculate the shortest path [12, 13, 14].

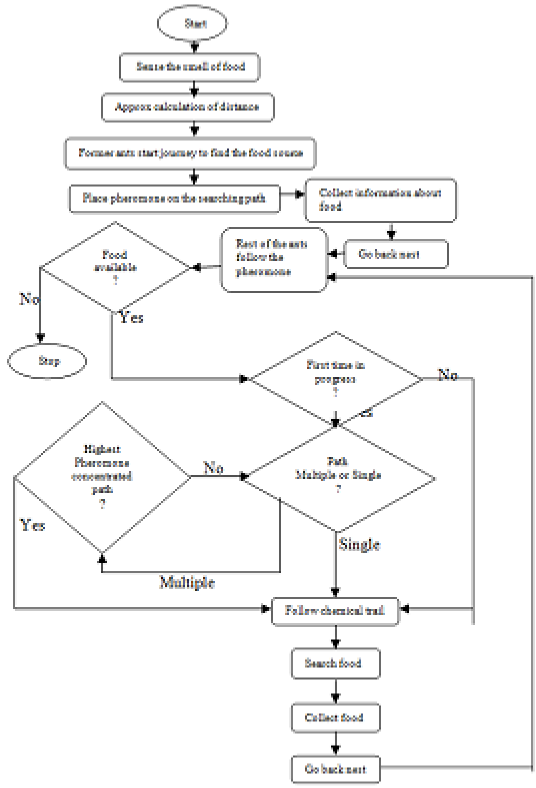

Figure 2. Modified Ant Colony Algorithm (MACA)

Real-life ants, in Ant Colony System, guide their following ants by placing pheromone on their travelling path from nest to food source and back to nest. There are a number of agents in ACS that but each of the agents have a limited ability for the solution of a problem and so they place the pheromone and create the pheromone trail through which they create a controlling environment over the colony for a successful search of food [7, 11]. In robot automation, the same thing can be observed if an efficient algorithm can be placed there and create an automated environment where a robot is capable of controlling and awakening another robot for a movement in a chain wise just like the ants perform.

III. Robot Colony System (RCS)

In robot colony, there may be one or more than one robot may exist, performing their respective tasks either solely or by following each other. If they work and walk by sensing each other then a sensor system must be there to control on sensing each other and to work or walk smoothly but if it work solely, a routing map for the path that can determine the shortest path to find a particular place, moving from node to node, must be installed in the robot memory that can help it to find out that to which node to move from the present node. In this paper, a new concept of path map is introduced based on Kruskal's algorithm.

3.1 Robot Path Algorithm

The Robot Path Algorithm (RPA) is actually based on the Modified Ant Colony Algorithm (MACA) already shown in Figure 2. This RPA algorithm direct the robot to the proper way of movement.

There are mainly two type of moves:

1. The forward path: The robot starts its journey towards the destination to collect information or something else for necessity.
2. The reverse path: The robot, after collecting the information or the necessary things, go back to it was previously located. So, for the movement of the robot, two algorithms are required for the two types of move and merging them to form a uniform algorithm for the entire movement of the robot from source to destination and vice versa. While moving from node to node, there must be a sensing system be there to ensure the robot that the information has been collected and till then the 1st algorithm will be iterated and soon after the information is collected, the system switch to the 2nd algorithm for going back to its previous position. The algorithms are described below in Figure 3 and Figure 4.

When the robot starts its new journey from one node to its destination node, its memory keeps the record of its previous location where it was previously located. After collecting information from destination node, it will then read the memory for its home node and when it will get information about its home node then it will starts journey to its home node. The fundamental operation incorporated in the reverse path that it will not again find the shortest path because it has already travelled the shortest path and the record is stored in robot memory. The robot then fetch the information of the shortest path from memory and starts its journey to its home node following the corresponding edge.

The main advantage with RPA-2 algorithm is that the robot need not to find the shortest path route again for the second time rather it will fetch the previous record of route direction and saving the computation time for shortest path search. But if it will happen that the robot starts its journey form one node, reached to destination node and after receiving the information it will have to go to another node to submit information, then it will have to follow the RPA-1 algorithm again to search the shortest path and then proceed to the another destination node. But in this paper, the process includes that the robot starts from its home node to destination node and come back to its previously located node, based on ACS fundamentals [1, 2, 21].

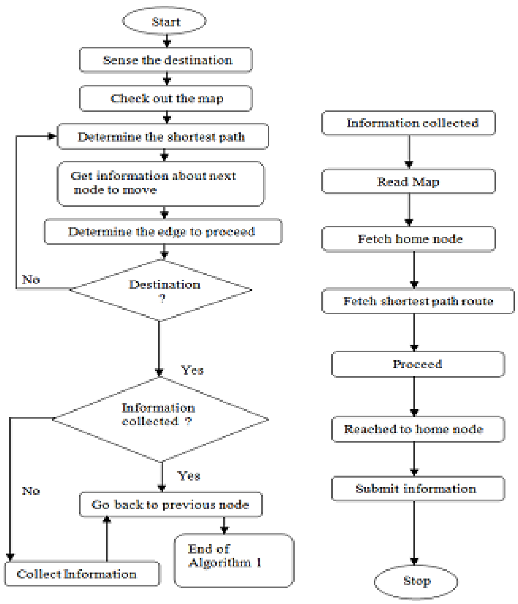

Figure 3. RPA-1 Algorithm (Forward path) Figure 4. RPA-2 Algorithm (Reverse path)

3.2 Pseudo code for ACS based RPA

```
RPA-1
Starts journey;
Sense for information;
For N(m):= 1 to n do
      For i:= 1 to j do
            D = N(mi ) - Np(m);

            If    D = = Es (m )
                  Np(m)  = N(miselect);
                  Save node;
                  Save edge;
            Else
```

```
                  i = i + 1;
                  Repeat;
         End for;
               If    Np(m)=N(n)
                     Save path;
                     Collect Information;
```

```
RPA-2
Collect the information;
Read map;
Fetch previous information;
Select node;
Select edge;
Read home node;
For N(m):= n to 1 do
      Proceed     to      home      node;

      m = n – 1;
      If   m = = 1
            Stop;
      Else
            Repeat;
End for;
```

```
Notations  used:
D= Distance between two nodes
N(m)= Number of  Nodes
Np(m)=Present nodes
N(mi )=Neighbor node;
m ϵ N(mi )… i=1,2,3…..l
N(miselect)=Next node
Es(m )=Shortest edge
```

3.3 Robot Path Map (RPM)

Robot Path Map (RPM) is a database based map system. RPM holds all the data about of the path through which the robot is proceeding towards its goal. The main advantage of having a RPM based robot is that it does not need any online update of the path that it is following as it includes all the data that are being accessed by the robot on the path such as the node value, the edge value, the shortest path, *etc.*, and for that reason it can work offline. The proposed structure of RPM is shown in Figure 5.

In RPM, each square is considered as 1 square unit and the block 1 is considered as the home block for the robot. Each block are connected with each other with a connector line. There are 23 nodes in the 8X8 colony system so that each of the single block can be connected with each other so that there must be no problem with

```
            Else
                    m  = m + 1;
              Repeat;
      End for;
```

collecting information from any of the blocks. While travelling, the robot must follow the algorithms described in Section 3.2.

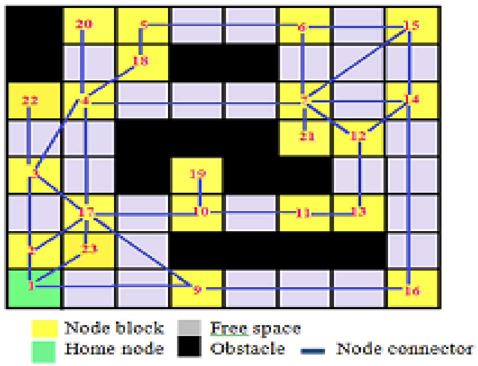

Figure 5. Proposed Robot Path Map (RPM)

IV. Robot Path Map Database

A robot memory includes a map and a database. When a move is executed, the robot follow the map and the corresponding data is fetched from the database. That means the map and the database cooperate each other for the movement. The database includes the following:

1. Node value
2. Edge value
3. Sequential node declaration
4. Proposed shortest path

A prototype of the node connectivity is shown in Figure 6.

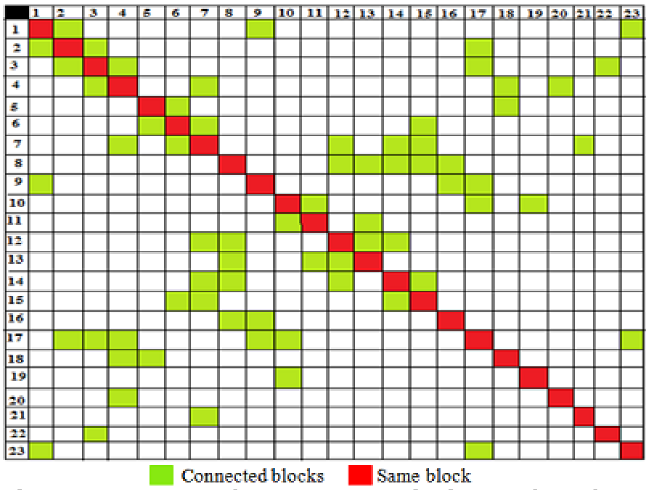

Figure 6. Node connectivity database (NCD)

Figure 6 shows the node to node connectivity of the total of 23 nodes as prototype. This connectivity database will be checked for the first time to acquire the knowledge about the neighbor nodes. Then the robot will check for the nearest node preferred for the shortest path towards the goal. Here a question may arise that, is finding a shortest edge each time is enough for finding the shortest path? The answer is obviously NO. Because, each time the shortest edge from the nodes may not construct the shortest path. For this case, the robot must follow the shortest path whether it travels through an edge which is not the shortest one. For example, to travel from node-1 to node-6 there are three possible shortest paths

1. 1-2-3-4-18-5-6
2. 1-2-17-4-18-5-6
3. 1-23-17-4-18-5-6

For the first case, the corresponding edge values are respectively,
(1+2+2.236+1.414+1+3)units = 10.65 units

For the second case, the corresponding edge values are respectively,
(1+1.414+3+1.414+1+3) = 10.828 units
For the third case, the corresponding edge values are respectively,
(1.414+1+3+1.414+1+3) = 10.828 units

Now, from the above calculation, we can see that, node-2 is nearer of node-17 than that of node-3, but to follow the shortest path, robot will follow the node-2□ node-3 fashion to achieve the goal. So the preferred path from node-1 to node-6 will be,

node-1□ node-2□ node-3□ node-4□ node-18□ node-5□ node-6

4.1 Shortest path determination algorithm

The actual shortest path from node to destination can be efficiently determined by the algorithm shown in Figure 7. First, the robot select the destination node, then it read the NCD and acquire the knowledge about its neighbor nodes. Next it read its memory about the preferred shortest path among all the possible way to reach to the destination node. Just after determining the shortest path, it comes to know that through which nodes it will travel so that the shortest path can be achieved. The all data about the nodes travelled, the edge value, the shortest path etc. are recorded into the robot memory and all the records will be replayed while it travel from the destination node to its home node. The algorithm is shown in Figure 7.

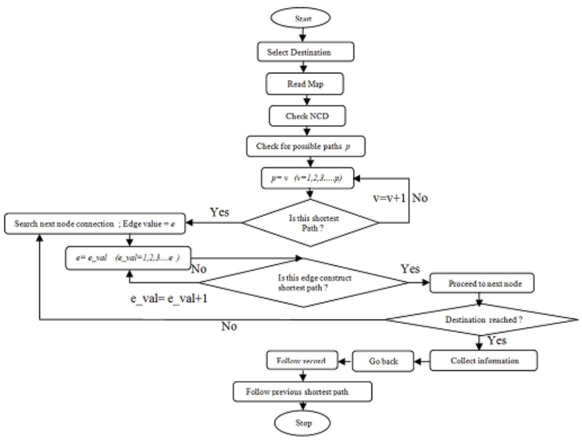

Figure 7. Shortest Path Algorithm (SPA) for Robot colony

V. Conclusion

The RPS-1 and RPA-2, based on Modified Ant Colony Algorithm, introduced in this paper, are very useful to route the robots in robot colony system in offline mode. The NCD is introduced to check for the connectivity of one node with the another while moving. All the above mentioned algorithms and database are combined together and it forms an algorithm called Shortest Path Algorithm (SPA) which is the final algorithm for the robot colony system. This algorithm is actually helps the robot for routing and hence it is an powerful approach for robot path planning driven by the database and controlled by the algorithms to find the shortest path. This method is again reflected in the simulation files where a 450 iterations are shown. Iteration is nothing but the repeatative search for the objects that are placed in different places or in different nodes. The simulation results shows the process of search of the object and the movement of the agents in the colony. This design is useful for the implementation of Robot Path Planning as well as the Travelling salesman and also in other place where agent based methods can be used.

REFERENCES

[1] N. B. Sariff and O. Buniyamin, "Ant Colony System for Robot Path Planning in Global Static Environment", Selected Topics In System Science And Simulation In Engineering, ISSN: 1792-507X, ISBN: 978-960-474230-1, pp. 192-197.

[2] N. Buniyamin, N. Sariff, W. A. J. Wan Ngah and Z. Mohammad, "Robot Global Path Planning Overview And A Variation Of Ant Colony System Algorithm", International Journal Of Mathematics And Computers In Simulation, vol. 5, Issue 1, **(2011)**, pp. 9-16.

[3] V. Osipov, P. Sanders and J. Singler, "The Filter-Kruskal Minimum Spanning Tree Algorithm", Copyright © by SIAM, pp. 52-61.

[4] H. Mei, Y. Tian and L. Zu, "A Hybrid Ant Colony Optimization Algorithm for Path Planning of Robot in Dynamic Environment", International Journal of Information Technology, vol. 12, no. 3, **(2006)**, pp. 78-88.

[5] L. K. Behera and A. Sasidharan, "Ant Colony Optimization for Co-operation in Robotic Swarms", Advances in Applied Science Research, vol. 2, no. 3, **(2011)**, pp. 476-482.

[6] M. Dorigo, M. Birattari and T. Stutzle, "Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique", Iridia – Technical Report Series: TR/IRIDIA/2006-023.

[7] M. Dorigo, G. Di Caro and L. M. Gambardella, "Ant Algorithms for Discrete Optimization", Artificial Life, vol. 5, **(1999)**, pp. 137–172.

[8] M. Dorigo, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, **(1997)**.

[9] M. Dorigo, V. Maniezzo and A. Colorni, "The Ant System: Optimization by a colony of cooperating agents", IEEE Transactions on Systems, Man, and Cybernetics–Part B, vol. 26, no. 1, **(1996)**, pp. 1-13.

[10] H. S. Khandre, "Review Of Application Of Ant Colony Optimization", International Journal of Engineering Science and Technology (IJEST), ISSN: 0975-5462, NCICT Special Issue, **(2011)** February.

[11] E. Foundas and A. Vlachos, "Pheromone models in ant colony optimization (ACO)", Journal of Interdisciplinary Mathematics, vol. 9, no. 1, **(2006)**, pp. 157-168.

[12] M. Gupta, M. Dua and N. Kumar, "CNS using restricted space algorithms for finding a shortest path", International Journal of Engineering Trends and Technology, (2011) July - August.

[13] G. Nirmala and K. Uma, "Fuzzy Shortest Route Algorithm for Telephone Line Connection", International Journal of Scientific and Research Publications, vol. 2, Issue 8, **(2012)** August, ISSN 2250-3153.

[14] J. Sims and N. Meghanathan, "Construction And Evaluation Of Meshes Based On Shortest Path Tree Vs. Steiner Tree For Multicast Routing In Mobile Ad Hoc Networks", Journal of Theoretical and Applied Information Technology © 2005 – 2010.

[15] S. -C. Chu, J. F. Roddick and J. -S. Pan, "Ant colony system with communication strategies", Information Sciences, vol. 167, **(2004)**, @ 2003 Elsevier, Inc., pp. 63–76.

[16] C. -H. Chen, Y. Ze and C. -J. Ting, "An Improved Ant Colony System Algorithm For The Vehicle Routing Problem", Journal of the Chinese Institute of Industrial Engineers, vol. 23, no. 2, **(2006)**, pp. 115-126.