International Journal of Hybrid Information Technology
Vol.6, No.6 (2013), pp.11-30
http://dx.doi.org/10.14257/ijhit.2013.6.6.02

Ant Colony System: A New Concept to Robot Path Planning

Subhadeep Chakraborty

Department of Electronics and Communication Engineering,
West Bengal University of Technology,West Bengal, India

subha.journal@gmail.com

Abstract

A variation of Ant Colony System (ACS) is represented in this paper and applied for the Robot Path Planning (RPP) purpose. The algorithm shows a new way to find the shortest path from source to destination in offline mode with the application of in-build path map by following the Robot Path Algorithm (RPA), introduced in this paper. Robot always follow the path map provided to it to find the shortest path as well as it can achieve the knowledge that in which direction, i.e. from one node to the next node, it will have to move. The movement of the robots is based on the movent technique of the ants in the ant colony. Among all the algorithms for finding the shortest path, the proposed Shortest Path Algorithm (SPA), based on Kruskal algorithm, is much more effective and accurate for the RPP problem and will take less computational time and hence increase the efficiency of the work process of the robot system.

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

1. 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.

2. 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

ISSN: 1738-9968 IJHIT

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.

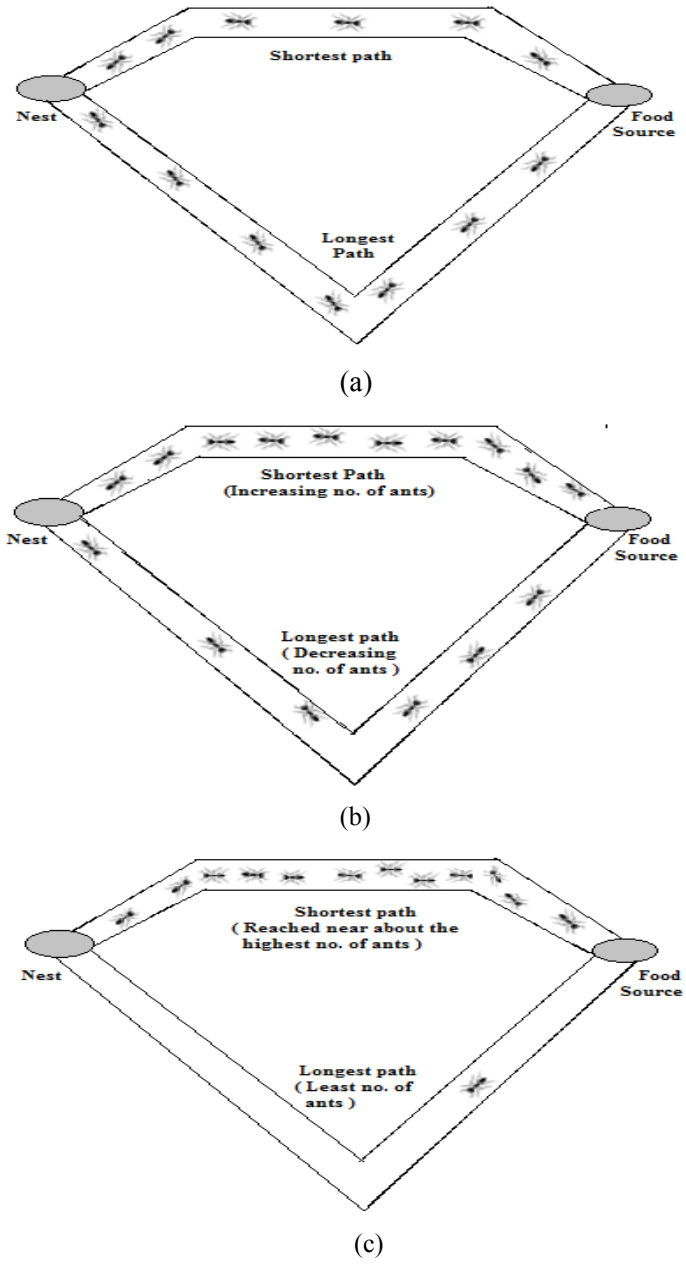


Figure 1. Double bridge experiment: (a) Ants starts their journey (t=0); (b) Ants find the shortest path(t=1); (c) Ants follow the shortest path and accumulate on that path(t=2)

2.1 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)).

2.2 Algorithm for shortest path

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

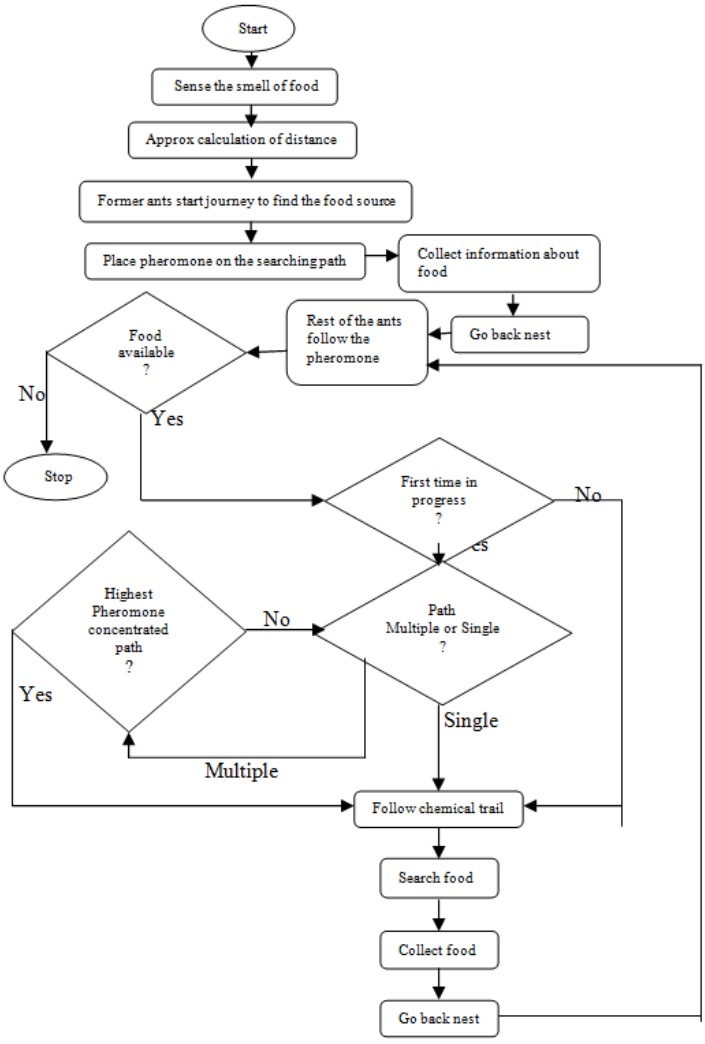


Figure 2. Modified Ant Colony Algorithm (MACA)

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].

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.

2.3 State transition rule

The state transition rule determines the process that an agent *agnt* processing to node *n1* choose another agent to move to node *n2* at time t, *i.e.*, from the meaning of transition, one agent or robot will be sent to a particular node with the influence of another controlling robot in the same colony [17]. In Travelling Salesman Problem (TSP), the same phenomenon is observed where *agnt* is an agent whose duty is to conduct a tour and it visit all the cities and come back to the starting point. While travelling, *agnt* has a list of cities Y_x^W and can remember which cities are already travelled and in this way the agent follow a single city as a single node and complete its journey to the all cities on nodes visiting there by a single time [15, 16, 17].The state transition rule can be represented as follows [7, 8, 15, 16, 18, 20]:

$$n2 = \begin{cases} \arg\max_{u \in S_{agnt}(n1)} \{[\tau(n1,n2)] \cdot [\eta(n1,n2)]^{\beta}\} \\ N \end{cases}$$

$$\text{if } q \le q_0 \ldots\ldots (a)$$

$$\text{otherwise} \ldots . (b)$$

……(1)

Where, $\tau(n1,n2)$ = pheromone information

$\eta(n1,n2)$ = heuristic information

q = randomly chosen variable with uniform probability in [0,1]

q_0 = variable parameter($0 \le q_0 \le 1$) that determines the relative importance between Eq. 1(a) & (b)

β = heuristic coefficient

$n1, n2$ = Node

$S_{agnt}(n1)$ = list of all nodes to be visited

$\mathcal{N}$ = random variable chosen based on Eq. 2 that shows the probability of an agent in node n1 to visit node n2.

$$P_{agnt}(n1,n2) = \frac{[\tau(n1,n2)]\cdot[\eta(n1,n2)]^{\beta}}{\sum_{x \in S_{agnt}(n1)} [\tau(n1,x)]\cdot[\eta(n1,x)]^{\beta}}, \text{if } x \in S_{agnt}(n1) \quad \ldots.(2)$$

Let, a set of nodes, we choose two nodes labeled as node 'a' and node 'b'. The distance in between the two nodes is denoted as d_{ab} and the inverse of the distance is called the visibility, basically employed for controlling and directing the ant search for the next robot to inform it about the surrounding nodes of the present one where it is now located, denoted by η_{ab}, *i.e.*, [17, 18, 19, 20],

$$\eta_{ab} = 1/d_{ab} \quad \ldots\ldots(3)$$

There is the possibility that many nodes are to be connected with a node. The state transition rule will follow the shortest edge for the transition for high amount of pheromone trail. The Eq. 1 shows the transition process whether it utilize the knowledge about different amount of pheromone trail on different edges (Exploitation) or it proceed towards the shortest and high pheromone trail edge (Exploration) [9, 20].

2.4 Pheromone update rule

In ACS, when ants move from one to another, some of the pheromone are reduced due to evaporation and each time pheromone is updated and a new trail is created [2, 17, 20]. The construction of the new trail can be shown by the formula given in Eq. 4 as follows:

$$\tau_{nx,ny}(\text{new trail}) \leftarrow (1-\rho)\cdot\tau_{nx,ny}(\text{old trail}) \quad \ldots..(4)$$

Where,

nx, ny = edges connecting nodes nx and ny, the best awakened edge for the ant to move.

ρ = pheromone update parameter, shows the pheromone decay on the path.

3. 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.

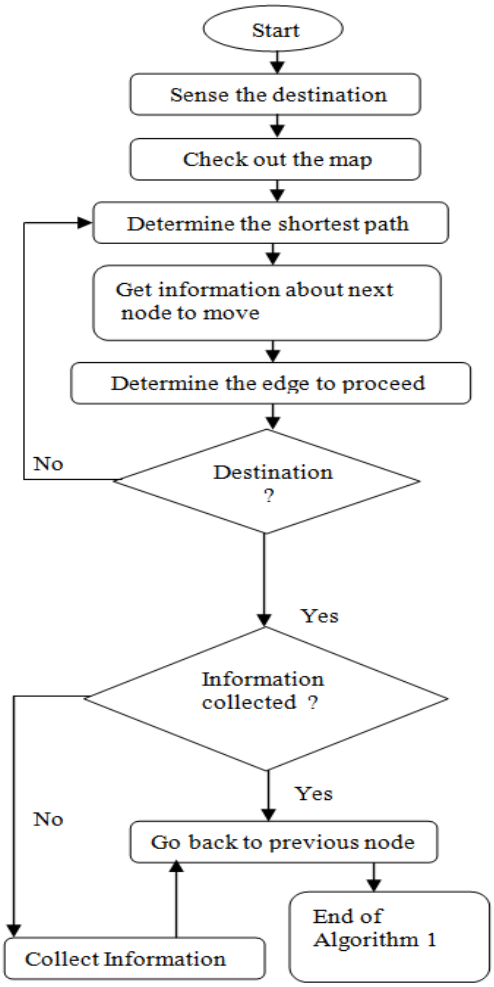
There are mainly two type of moves:

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.



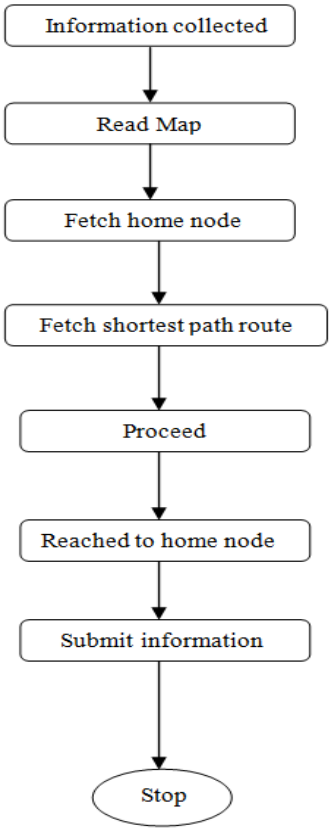


Figure 3. RPA-1 Algorithm (Forward path) Figure 4. RPA-2 Algorithm (Reverse path)

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].

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

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
$N_P(m)$=Present nodes
$N(m_i)$=Neighbor node; $m \in N(m_i) \ldots i=1,2,3\ldots..l$
$N(m_{iselect})$=Next node
$E_S(m)$=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.

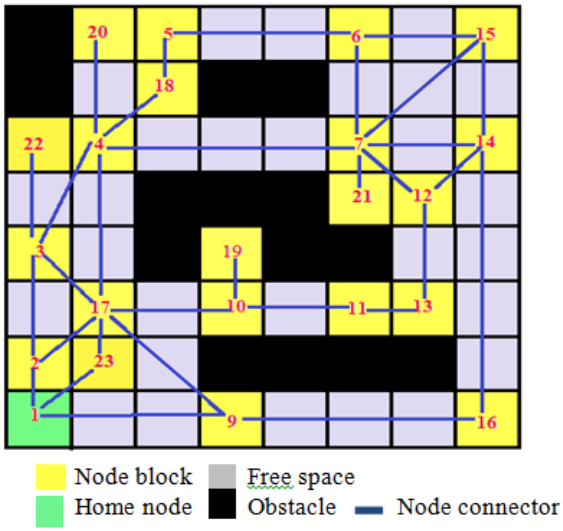


Figure 5. Proposed Robot Path Map (RPM)

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

3.4 Calculation of Distance

One of the essential operation of the RPA algorithm is to measure the distance in between two nodes. The distance between two nodes can be measured directly (*e.g.*, 1→2 or 4→17, *etc.*) or by using the trigonometric technique (*e.g.*, 3→4 or 7→15, *etc.*). As the each square block is considered as an area of 1 square unit, so the length of each side is 1 unit.

Let consider a single block of the RPM, in the following figure Figure 6.

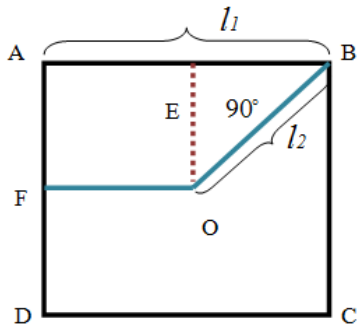


Figure 6. An unit block of RPM

In Figure 6,

1. AB=BC=CD=DA
2. AB= $l1$=1 unit

3. AB=BC=CD=DA=1 unit
4. $\angle$OEB= 90°
5. *l3* = 0.5 unit
6. OF = OE = EB = *l3* = 0.5 unit

From the above data, the length of OB= *l3* can be calculated using Pythagorean theorem of right triangle, *i.e.*,

$$OE^2 + EB^2 = OB^2 \qquad \text{....(5)}$$

So the length of OB is,

$$OB^2 = (0.5)^2 + (0.5)^2$$

or, $OB = \sqrt{(0.5^2 + 0.5^2)}$

or, $OB = \sqrt{(0.25 + 0.25)}$

or, $OB = \sqrt{0.5} = 0.7071$ unit

So, for example the length of the path connecting the node 3 and node 17 is,

2x *l2*= 2x0.7071 = 1.4142 unit.

Now, let we consider the length of the path that is connecting nodes 3 and 4. This can be done simply by trigonometric law. The connection can be drawn as follows in Figure 7:

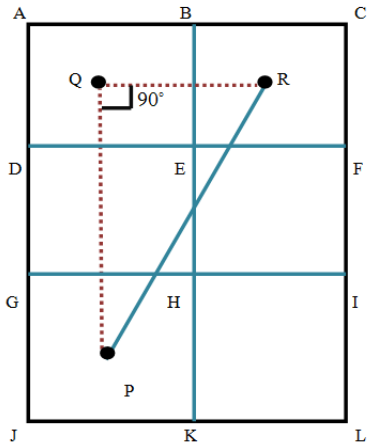


Figure 7. Three inter connected blocks

In Figure 7,

1. PQ = 2 unit
2. QR = 1 unit
3. $\angle$PQR = 90°

From the above data, the length of PR can be calculated using Eqn. (5), *i.e.*,

$$PQ^2 + QR^2 = PR^2 \qquad \text{.....(6)}$$

So, the length of PR is,

$$PR^2 = 2^2 + 1^2$$
$$\text{or, } PR = \sqrt{(2^2 + 1^2)} = \sqrt{5} = 2.236 \text{ unit}$$

So, all the length of the edges can be determined using Pythagorean theorem of right triangle. Those edge values are predefined in the database of the robot memory. The database includesall the data that are essentially required to read and calculate the distance of the total path.

4. 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 8.

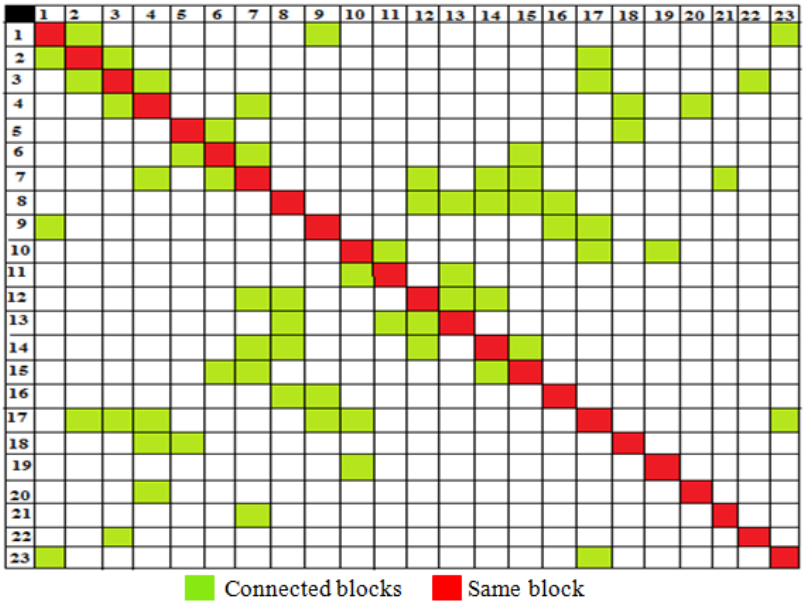


Figure 8. Node connectivity database (NCD)

Figure 8 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 9. 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 9.

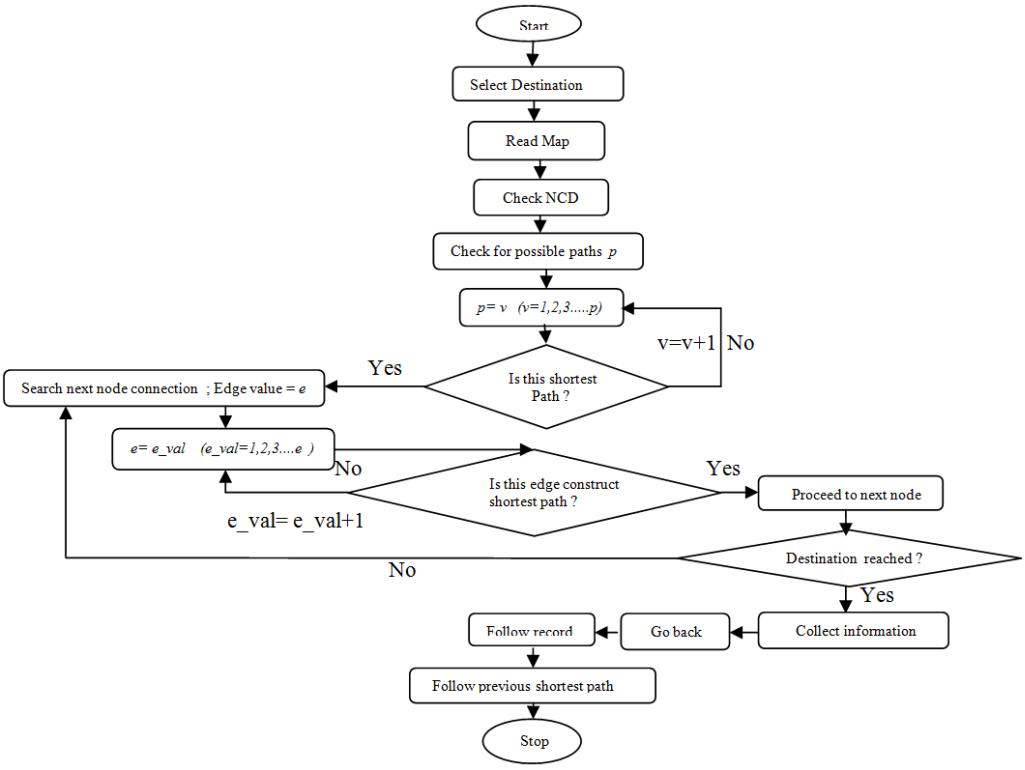


Figure 9. Shortest Path Algorithm (SPA) for Robot colony

So, by executing the algorithm, the robot will be able to find the shortest distance from source to destination by following the shortest path rather by following the shortest path each and every time while travelling from one node to another.

4.2 Robot Path design and Simulation using Ant Movement

A Robot path of 200 agent with specific home and a number of object is designed by the concept of above mentioned algorithms and with help of the Kruskal's algorithm. In this design, after a number of iterations, the agents can reach the destination and can collect the preferred object by saving the specific line of journey which may be helpful to other agents to find the way to reach to the destination. After leaving the home node, the travel in a random way to find the object. After some iteration, when one of the objects, placed in the colony, is found, they collect that and the corresponding line of journey is saved and then they go back to the home node. So, in this technique, the agents will move towards the object and after successful collection of the object, the come back to the home with the information that they have successfully collected the object and after receiving the information more agents will fillow the line and can reach to this and other object for more collection.

The simulation files from Figure 10 to Figure 19 are shown below with an interval of 50 iterations.

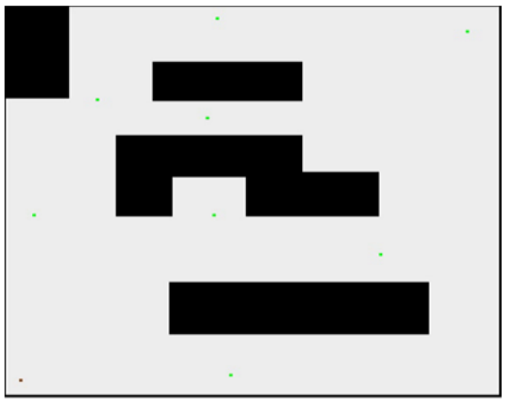


Figure 10. Agent Colony System (Iteration=0)

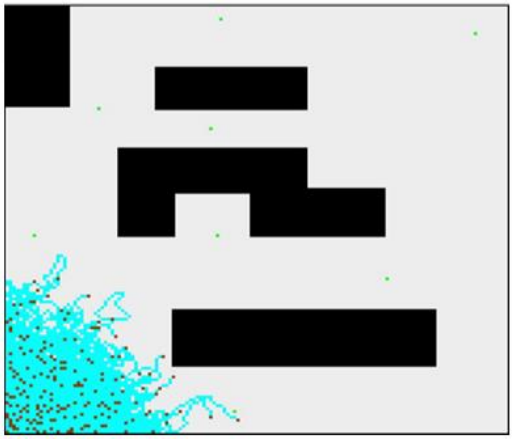


Figure 11. Agents star moving (Iteration=50)

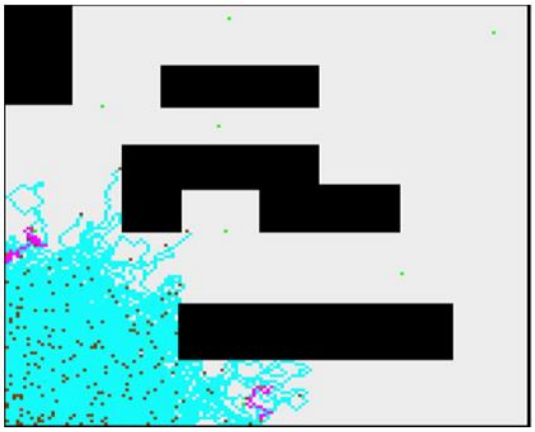


Figure 12. Agents collects the object (Iteration=100)

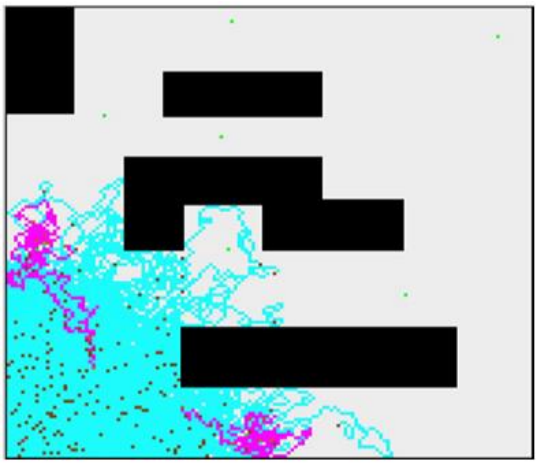


Figure 13. Collection information is spreading (Iteration=150)

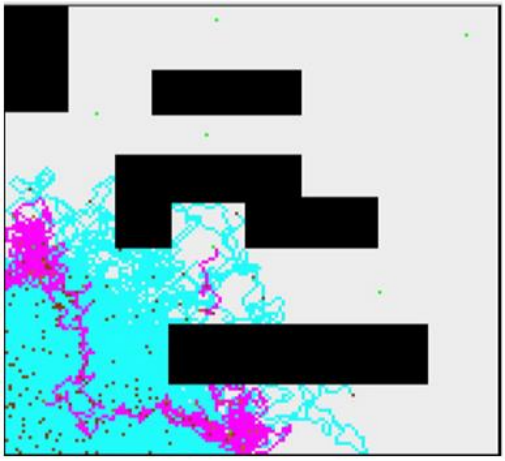


Figure 14. Information spread in more agents (Iteration=200)

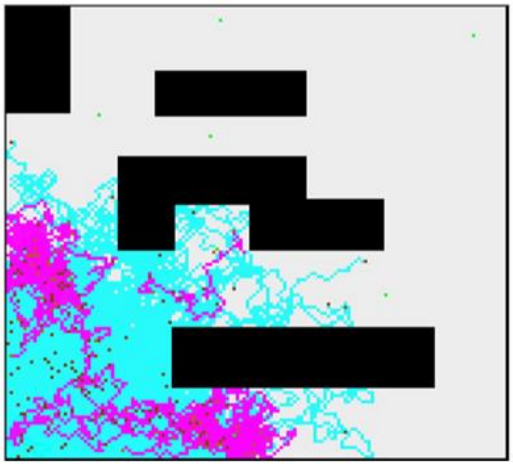


Figure 15. More agents are involved in collection (Iteration=250)

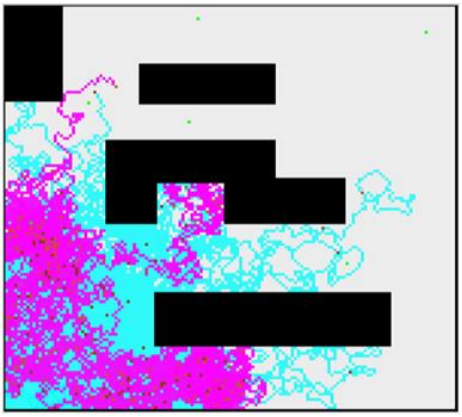


Figure 16. Agents move towards other objects (Iteration=300)

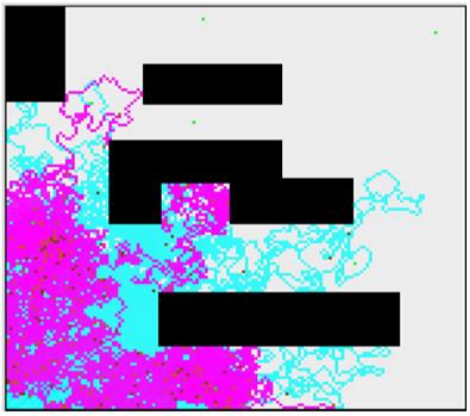


Figure 17. Agents are aware of more object information (Iteration=350)

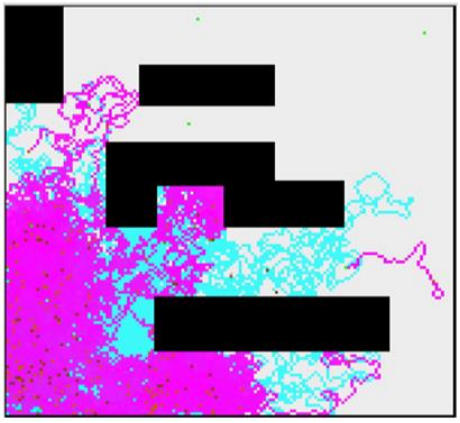


Figure 18. More collection informations are created (Iteration=400)

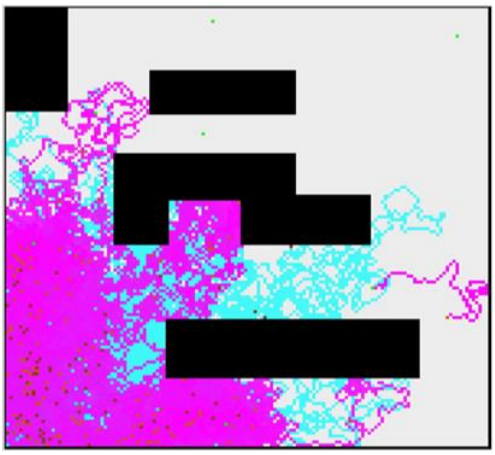
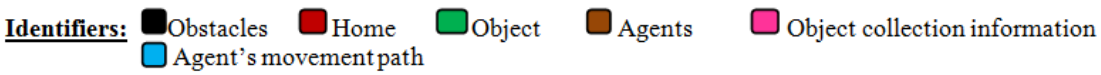


Figure 19. Maximum agents are aware of the nearest objects and collect them (Iteration=450)



From Figure 10 to Figure 19, it can be seen that, when the objects are placed in the colony, the agents start moving towards them for collection. They first search the object place and when find out the place of the object, they collect those and this information is conveyed over all the other agents so they can go and collect those. Once an object is successfully collected, they move back to their home node. After that, other agents move towards for search in another objects and other agents follow the path. So, in this strategy, in agent based system, the agent move to their destination and after successful collection of the object, they go back to their home node. The sumulation are based on the Robot Path Map(RPM) that is the design of the simulation just follows the path structure described in Figure 5. The procedure of the simulation, incorporated in this paper, follows the Shortest path algorithm (SPA), described in Figure 9. The Robot Path Algorithms (RPA) are also very needful for designing the colony structure.

5. 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.

Acknowledgments

In this paper, a new approach to ant colony system is introduced for robotics using the Kruskal's algorithm for the shortest path design. The RPM is designed in Fig.5 for the agent movement following the RPA algorithms. Finally the NCD is designed to update the robot for node-to-node connectivity which makes the robot easier to move and follow the shortest path rather than following the shortest edge each time by applying the SPA algorithm. The idea of this paper is taken from the list of references given below.

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: 978-960-474-230-1, 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.

[17] L. M. Gambardella and M. Dorigo, "Solving Symmetric and Asymmetric TSPs by Ant Colonies", IEEE Conference on Evolutionary Computation (ICEC'96), **(1996)** May 20-22, Nagoya, Japan.
[18] Z. Liu, M. Z. Kwiatkowska and C. Constantinou, "A Swarm Intelligence Routing Algorithm For Manets".
[19] B. Roy, S. Banik, P. Dey, S. Sanyal and N. Chaki, "Ant Colony based Routing for Mobile Ad-Hoc Networks towards improved Quality of Services".
[20] D. G. Bucatanschi, "The Ant Colony System for the Freeze-Tag Problem", MCURCSM 2004, Granville, OH, **(2004)**, pp. 61-69.

Author

Subhadeep Chakraborty, born in 1986, is Assistant Professor in Calcutta Institute of Technology. He received the B.Tech degree from Saroj Mohan Institute of Technology, WBUT, India and M.Tech degree from Kalyani Govt. Engineering College, WBUT, India in Electronics and Communication Engineering in 2008 and 2010 respectively.The author has been teaching in Calcutta Institute of Technology for 3 years. His primary research interest includes Digital Signal Processing,Image Processing, Embedded System, Artificial intelligence, Microprocessor and Swarming.

