International Journal of Hybrid Information Technology
Vol.7, No.2 (2014), pp.249-268
http://dx.doi.org/10.14257/ijhit.2014.7.2.23

Ant Colony System: An Improved Approach for Robot Path Planning

Subhadeep Chakraborty

Department of Electronics and Communication Engineering, Calcutta Institute of Technology, West Bengal, India

subha.journal@gmail.com

Abstract

In Robot Colony System(RCS), based on the swarming nature of Ants, the path for the agent robots is designed to perform to search operation to find the shortest path from source to destination for collection of objects and go back to their hive-like home node. The agent always move through the shortest path to reach to the destination. There is a possibility of having a number of path in between the source and destination as defined in the Path Map(PM) in offline mode. If the prefferd shortest path is blocked by means of some obstraction, there must be another way to reach to the destination which have the weight of minimum among the other possible paths. The algorithm shown in this paper for shortest path, based on Kruskal's Algorithm, shows the way to find the alternative shortest path and the moving direction of the agent. In this paper, the junction-to-junction connectivity is proposed where the path search is replaced by the node search which minimises the computaional time and hence increases the effectiveness and efficiency in agent moving towards the destination from source and also in its reverse direction.

Keywords: *Ant Colony System, Shortest path, Shortest distance, Kruskal's Algorithm, Junction-to-Junction connectivity, Junction Connectivity Database.*

1. Introduction

In Robot Colony System (RCS), the movement of the agent is designed by following the natural behaviour of the ants. In Ant Colony System (ACS), the real ants sense the smell of the food and start journey from the hive and reach to the destination that is to the food source and after collecting the food they go back to their hive by placing pheromone on their movement path and by avoiding the obstacles. In case of facing any obstacle on the path, they use to change their movement path and creat a new path to the food source. In RCS, the behavoiur of the agent robot is similar to that of the real ants except the matter of pheromone. The agent moves towards the destination by means of sensing the connected path [1, 2]. Agent start journey from their present position to the object or information destination by following the shortest path. If the shortest path is blocked for some concrete obstraction which cannot be overcome and the current position of the agent contains no node, the problem arises to which direction it will move towards the destination. A modified and new technique can solve the RPP problem [2, 4, 5]. The problem can be minimized by the junction-to-junction connectivity that is actually the connection in between the junction nodes situated in the RCS.The shortest path in between two nodes can be defined by MACA,based on Kruskal's algorithm [3, 4]. The agent robot can move towards the destination without any external information or human interpretation [4, 5, 6]. The algorithm proposed in this paper

ISSN: 1738-9968 IJHIT

can efficiently design the agent movement by using the predefined Path Map(PM) and modifying this when a problem will occur. So, the main objective is to reach to the destination through the shortest nearest node connection rather than finding the shortest path to minimize the computational time and increase the efficiency and time accuracy.

2. Ant Colony System and Previous Work

In real ant colony system, the typical behaviour that can be observed is to finding the shortest path from their hive to food source. Another important characteristics can be found in the real ant is to ability to avoid the obstacle on their path by smelling the pheromone placed on their path[7][8][9][10]. With help of these two freatures of ants in their colony, the artificial ant or agent (agnt) as well as their colony can be suitably designed.

2.1 Real Ants

In real ant system, the worker ants search for the food and after finding the source of the food, they collect the food and go back to the hive. For fulfilment of the operation, the informer ants first travel forward in search of the food source by smelling the food. By sensing the concentration of the smell of the food, the informer ants can find the food source and go back to the nest inform other ant about the food source. After that, all other worker ants then go forward to collect the food by following the pheromone trail, placed on the path by the informer ants. While moving, the former ants also place the pheromone on the path so that the follower ant smell it and can follow the path to colloct food. After collecting the food, they return back to hive in the reverse path by following the pheromon trail in the similar way [9, 10]. For the search operation, the ants always follow the highly concentrated pheromone path to determine the shortest path [7, 8, 9, 10].

The real ant can go forward to the food source by avoiding the obstacles on their path. If the hive and the food source can be imagined as two nodes, n1 and n2 and the there are two paths inbetween them of equal distacne, then they travel form n1 to n2 with equal number that is of probanility of 0.5 or in other sense of equal probability. If any obstacle is found on their path, they suitably avoid this and make themselves forward smoothly but in a lesser probability on the high obstacle path [6, 8, 10]. This process is shown in Figure 1.

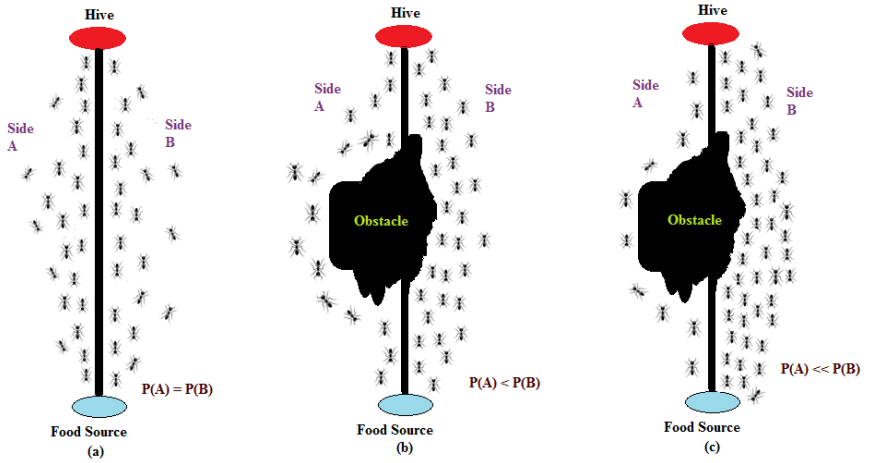


Figure 1. (a) Ants travels from Hive to food source (P(A) = P(B)); (b) Ants face obstacle and changing path direction(P(A) < P(B)); (c) More ants gatherd in Side B (P(A) << P(B))

2.2 Ant system and State Transition Rule

Transition means to travel from one place to another. Ant system can be well defined by the state transition as they travel from one place to another for collection of food. The artificial agent (agnt) can move one node or state to another node or state by the influence of former agent (agnt) in the defined colony by receiving some information rather than smelling the pheromone which the real ant performs [8, 12]. While moving, they care for the matter that they should visit a single node at a single time. That is they will not travel a node in multiple time [16, 17, 18]. For this purpose, Kruskal's algorithm is highly efficient for determining the shortest path from source node to destination without visitind a node twice [13, 14, 15]. This can be defined by the state transition rule. The State Transition Rule is represented as follows [8, 9, 16, 17, 19, 21],

$$n2 = \begin{cases} \arg\max_{u \in S_{agnt}(n1)} \{[\tau(n1,n2)] \cdot [\eta(n1,n2)]^{\beta}\} & \text{if } q \le q_0 \ldots\ldots(a) \\ N & \text{otherwise} \ldots\ldots(b) \end{cases}$$

……(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

N = random variable chosen based on Eq. 2 that shows the probability of an agent (agnt) in node n1 to visit node n2.

The agent (agnt), that is moving from node n1 to node n2, the movement of this particular agent (agnt) can be defined as follows [4, 8, 9],

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

..…(2)

The distance in between the two nodes is denoted as $dist_{12}$.The inverse of the distance is called the visibility, basically Applicable for controlling and directing the ant search for the next agent (agnt) to inform it about the surrounding nodes of the present one where it is now located, denoted by η_{12} *i.e.*, [4, 18, 19, 20, 21],

$$\eta_{12} = 1/dist_{12}$$

.....(3)

2.3 Pheromone Update Rule

In ant colony, while the movement of the ant, they place their pheromone on tha path. But after sometime the pheromone will evaporate gradually. Let one and is moving on the path and is placing its pheromone and leaving the particular place, the pheromone is evaporating with time constant. In this way, other ants travel and placing their pheromone and the same process will continue. So, in ant colony, there a phemone trail or a chemical trail is formed through which the real ants collect the proper information of movement. Pheromone placing and its evaporation will follow the pheromone update rule, defined below.

The trail about to be disappeared when the pheromone is evaporated and is a new trail can be formed when the pheromone is placed on tha path. This can be shown as follows [2, 4, 11, 18, 21],

$$\tau_{n1,n2}(\text{new trail}) \leftarrow (1-\rho)\cdot\tau_{n1,n2}(\text{old trail})$$

……(4)

Where,

$\tau_{n1,n2}$ = amount of pheromene diposited on the path from n1 to n2

ρ = Pheromone evaporation coefficient

So, each and every time the trail is updated by the diposited pheromone. Let, there are "n" number of ants are available in the colony.The trail is formed with the pheromone deposition of k-th ant can be shown as,

$$\tau_{n1,n2}(\text{new trail}) \leftarrow (1-\rho)\cdot\tau_{n1,n2} + \sum_{k}\Delta\tau^{k}_{n1,n2}$$

……(5)

Where,

$\Delta\tau^{k}_{n1,n2}$ = Amount of pheromone diposited by k-th ant, can be defined as,

$$\Delta\tau^{k}_{n1,n2} = \begin{cases} Q/L_k & \text{if curvature choosen in between n1,n2} \\ 0 & \text{otherwise} \end{cases}$$

……(6)

Where,

L_k = Cost or length of the path of k-th ant tour

Q = Constant

3. Modified Robot Colony System

The path of the artificial agent or the robot can be design suitably by the Modified Ant Colony Algorithm (MACA) by the definition of Path Map(PM) that is predefined in the robot memory [4]. PM is actually the database of all possible nodes available in the RCS. They use

to find the shortest path by calculating the distance between nodes. Through the shortest path, they travel and collect the information and go back to their prescribed loaction. MACA defines in a various way about the shortest path or the multi or singular path [4]. Whlile following the MACA, the agents simultaneously check for the forward and the reverse path also by obeying the RPA-1 and RPA-2 algorithm [4]. So, by following these three algorithms, the agents will visit to the information source and after collection of information, they go back to their position just from where they had started their journey. The forward and the reverse path for the agents can be defined as follows [4],

The forward path: The robot starts its journey towards the destination to collect information or something else for necessity.

The reverse path: The robot, after collecting the information or the necessary things, go back to it was previously located.

The Modified Ant Colony Algorithm (MACA) is shown in Figure 2 [4],

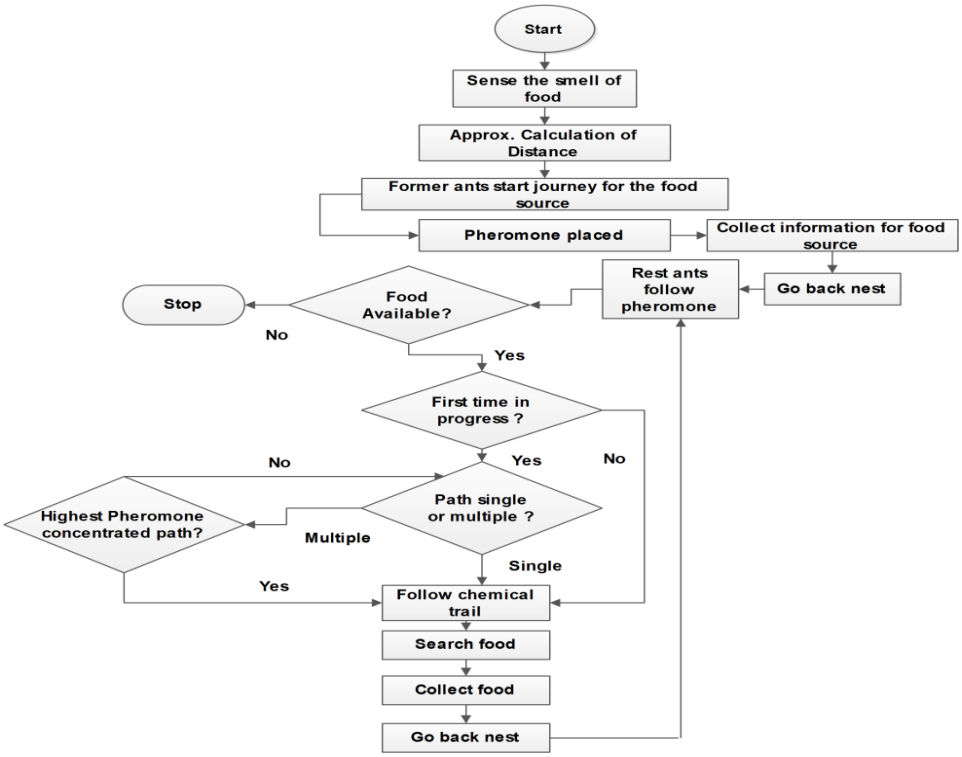


Figure 2. Modified Ant Colony Algorithm (MACA)

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 [4, 13, 14].

3.1 Robot Modified Path Map (RMPM)

The path map (PM) for a robot colony is shown in Figure 3 [4],

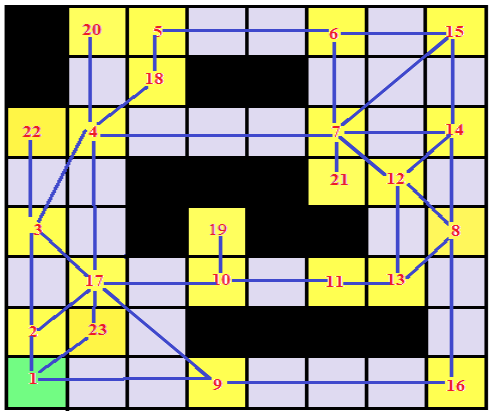


Figure 3. Path Map(PM)

In the RCS, there are 23 interconnected nodes are in the PM. So, the agent can travel through the connected path and can reach to the destination. But, if the shortest path is aborted by means of some obstacle, then the problem will arise as because then a new calculation is needed to find the shortest path. This requires a greater computational time. To avoid this circumstances, the junction-to-junction connectivity is introduced where each junction node in a preefined region is connected to the junction node of another region. If the actual shortest path is aborted, there will be no problem in reaching to the destination as because in this case, all the agent,presently working, are moved to the junction node from where they are routed to their destination via respective region router node or junction node.

So, the modified path map RMPM is shown in Figure 4.

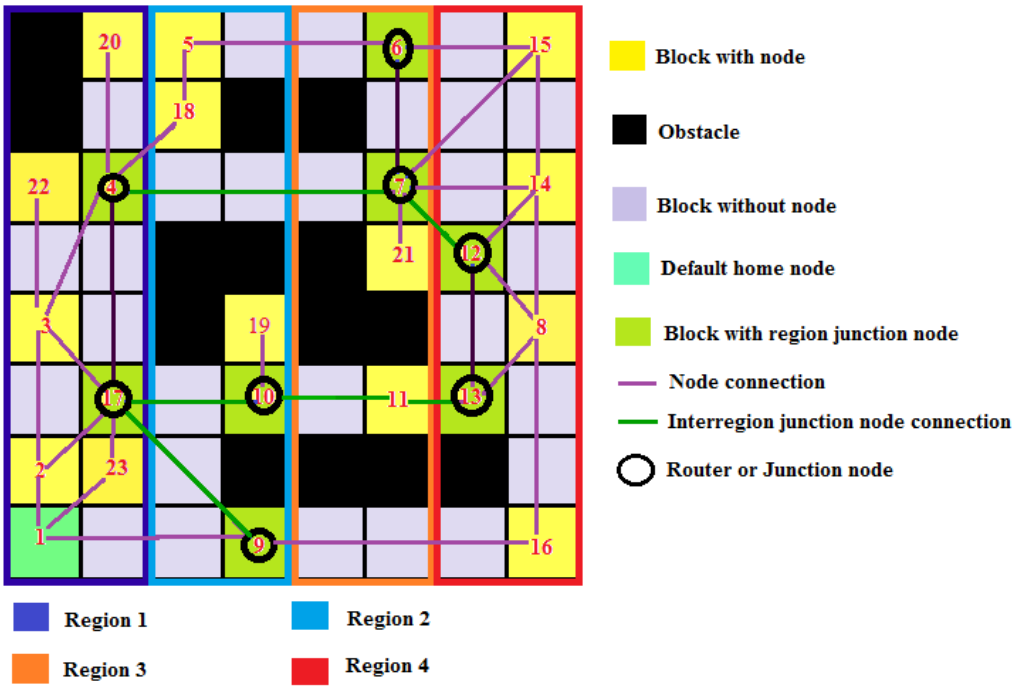


Figure 4. Robot Modified Path Map(RMPM)

In RMPM, the entire colony is segmented into four regions of same area. Each region have two junction nodes that are encircled in Figure 4. These junction node is responsible for routing the agents from one region to another. In this process, the each agent has to arrive at the junction node of the corresponding region. The difference between the basic junction, junction node is that, the basic junction, for example node(3) routes the agents in intraregion and the junction node, for example node(4) routes the agents in interregion.

3.2 Routing process in RMPM

RMPM consists of four regions where in each region two junction nodes are placed at the facing of each region. Every nearest junction nodes are connected together for quick route. The MACA is applicable for both intraregion routing or for interregion routing but the intaregion routing will specially follow MACA. The upgraded version for finding the shortest path based on MACA is applied for interregion routing , named as the Modified Nearest Junction (MNJ). MNJ checks for the nearest junction node of the home node as well as the destination. So, for the case of routing of an agent from one region to another, another complete algorithm is required, named as Modified Shortest Path(MSP). So, the agents will always follow MSP to reach one place to another place.

Let, an agent is starting its journey from "*x*" node and want to reach to the "*y*" node. From its present position, the agent finds the nearest junction node for this region and after successful search, it goes to that junction node. From that junction node, the search process again started for the nearest junction node of the destination node. When the nearest junction node of the destination is found without any obstacle, the agent is routed to that junction node. After reaching to that junction node, it will again apply MACA to reach to the desired node. So, using this searching method, the agent can reach to the destination in fewer time and avoiding the obstacle if there any without bypassing it which is usually seen in Any Colony.

So, for a single agent the MACA will be applied twice, one at its present region and another at the remote region, and MNJ in a single time for searching the nearest connected junction node. These algorithms will be simulated simultaneously to make the search process more compact and making it efficient by lessening the computational time.

The MNJ and the combination of MACA and MNJ, that is the Modified Shortest Path(MSP)are shown below.

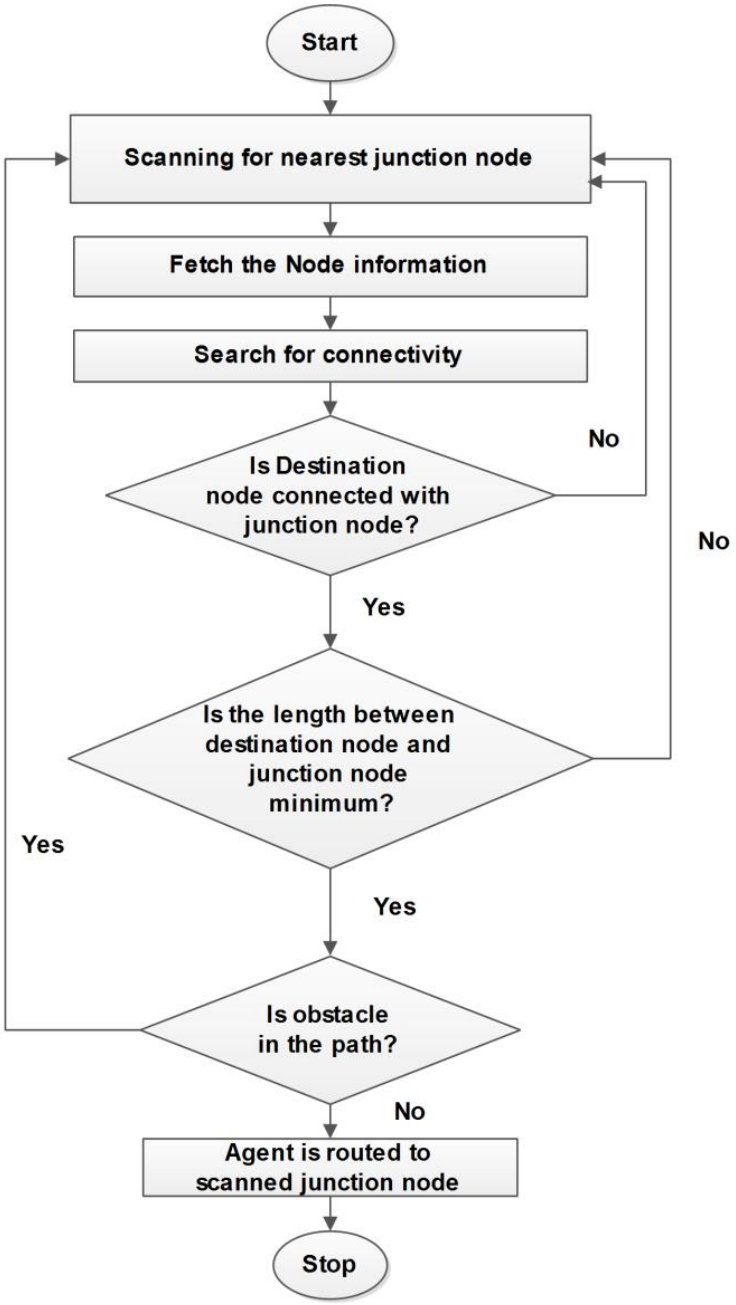


Figure 5. Modified Nearest Junction (MNJ) Algorithm

This MNJ algorithm is used in between two junction nodes whenever the junction-to-junction connectivity will be in search operation. Before reaching to the regional junction node and after starting juorney from the searched junction node to the destination, the MACA will be properly followed by the agent robot. So, the complete algorithm, MSP will be searching algorithm for the agent from the home node to the destination node. MSP is shown below,

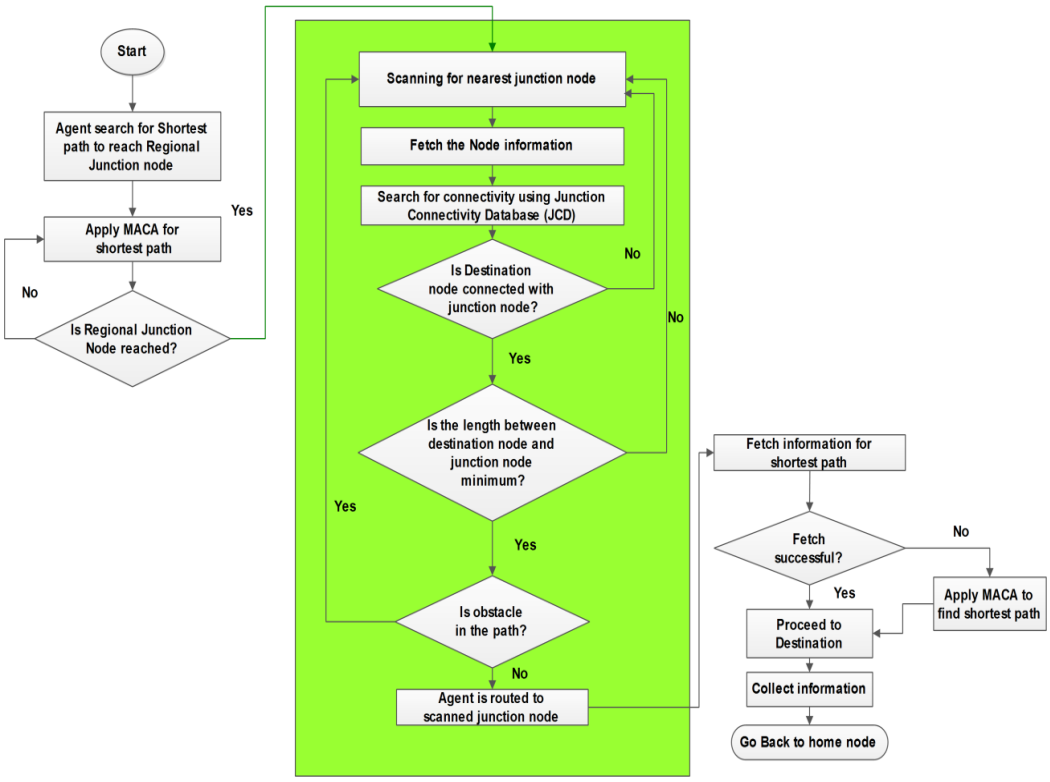


Figure 6. MSP Algorithm

While returning back from destination node to the home node, agent will follow the same algorithm just in a reverse way. The junction connectivity database (JCD) is used to keep the databases of the connectivities of the entire colony.

3.3 Junction Connectivity Database (JCD)

The regional junction node holds the connection with the enitre nodes in that region. The database contains all the weight or the cost of the connection with each node. When the agent arives at the junction node, they find the nearest junction node for reaching to the destination using the JCD. With help of JCD, the agents are aware of the nearest connection.

JCD reduces the cost search time of the agent and make them more efficient by reducing the computational time. JCD contains the information about all the regions and the regional junction node along with their connectivity with the remaining nodes present in the region. The connectivity is stored in form of a positive or negetive argument with the proper cost or weight. When an input is given into the system, the database helps to connect the virtual shortest lines from the present position to the desired position in a point-to-point technique. With help of this technique, agent can find the proper route to the destination. This technique is more helpful than the Node Connectivity Database (NCD) [4].

The proposed JCD is shown below.

Junction Nodes	4	6	7	9	10	12	13	17
4	Same Node	5.414	4.000					3.000
6	5.414	Same Node	2.000					
7	4.000	2.000	Same Node			1.414		
9		13.000	11.000	Same Node		8.414	8.414	2.828
10					Same Node		3.000	2.000
12			8.414			Same Node	2.000	
13				8.414	3.000	2.000	Same Node	
17	3.000			2.828	2.000			Same Node

N.B: All the lengths shown in the figure are in terms of unit.

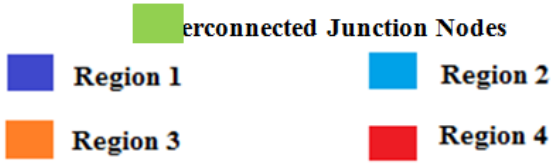


Figure 7. Junction Node Connectivity Database (JCD)

3.4 Calculation of inter-junction node connection

In the colony the agents, while moving, they should watch the JCD for accuring the knowledge of connection procedure. The connection actually represents the interconnection between the junction nodes along with their weights.

Now, let have a quick look on RMPM, it can be seen that there are a number of blocks which in group forms the wntire colony. The length of the square like unit are assumed ae 1 unit. That is depicted below with all the necessary informations.

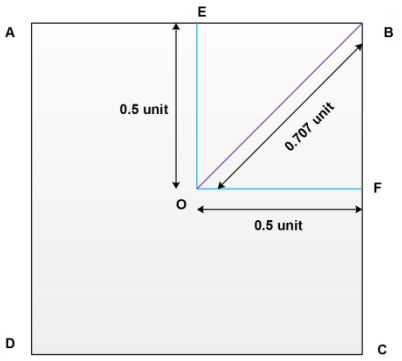


Figure 8. Unit block of RMPM

In the Figure 8,

AB=BC=CD=DA= 1 unit

OE=OF= 0.5 unit

$\angle EOF = 90^{\circ}$

So, the distacnce OB can be calculated by Pythagorean theorem of right triangle,

$$OB^2 = OE^2 + OF^2 \quad \ldots\ldots(7)$$

So,

$$OB = \sqrt{OE^2 + OF^2}$$

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

or $$OB = 0.707 \text{ unit}$$

So, in this process, all the connected length can be found out. These inter-junction node weights are shown in JCD.

The agents always look for the connectivity of their home node with the nearest junction node and then will check for the JCD to reach to the nearest junction node of the destination node. The entire node connections, actually the Updated Node Connectivity Database (UNCD), dependent on the JCD, is shown below,

Junction Nodes \ Basic Nodes	1	2	3	20	22	23	5	18	19	11	21	8	14	15	16
4	5.236	1.414	2.236	2.000	4.236		2.414	1.414							
17	2.414	4.236	1.414		3.414	1.000									
9	3.000											7.000	9.000	11.000	4.000
10									1.000	2.000					
6							3.000	4.414					4.000	2.000	9.000
7											1.000		2.000	2.828	7.000
12												1.414	1.414	3.414	4.414
13										1.000		1.414	3.414	5.414	4.414

Basic nodes connected with junction nodes

No Connection

Figure 9. Updated Node Connectivity Database (UNCD)

The agents wandering in the colony, while moving from the source to the nearest junction node in the region, will follow the UNCD. UNCD deals with the direct connection in between the basic junctions and the junction node that is, while measuring the distance in between the basic node and the junction node, no intermediate junction node will be in consideration. When the agent arrives at the junction node, they search for the shortest distance to the

junction node of that region where the destination node resides. While the junction node searching process, they will read the entire data from the JCD. The pseudocode for the entire process is given tn the next section.

4. Pseudocode of Agent Routing

In the RCS, the agents follow separate algorithms junction node searching and the destination searching. The pseudocode and the simulations are shown below.

4.1 Pseudocode

The pseudocode for nearest junction node search based on RPA-1 and RPA-2 [4] and the features of ACS, is as follows,

4.1.1 Pseudocode for Nearest Junction Node Search (NJNS)

```
NJNS
Fetch region;
Fetch N_x(l) for present region;
Search for nearest junction node:
for N_x(l) := 1 to k  do
          for m:= 1 to j do
               Dist =| N_p(n) −  N_p(n_m) |;
               if  N_p(n_m) == N_x(l)
                    Save distance;
                    Save node;
                    Save edge;
               else
                   N_p(n) = N_p(n_m) ;
                   m++;
                   go back & Repeat;
          end for;
          if  D_s  == Dist;
              Add all edges for the route;
              Calculate  D_s ;
              Save route( D_S1 );
          Else
              l++;
              Repeat;
end for;
Starts journey;
```

Notations used:

$N_x(l)$ = Junction nodes in the region

$N_p(n)$ = Agent present node

$N_p(n_m)$ = Present neighbour node

D_s = Shortest distance

The agent robot will find the exact shortest route through NJNS by searching the nearest junction nodes. They iteratively search the nearest node and continuously save the edge after respective search. Once it finds the nearest junction node, it connects the route by joining all the intermemiate basic nodes. After making the connection from the home node to the nearest junction node, it starts its journey towards the checked nearest junction node. The basic difference in between the Robot Path Algorithm [4] and the NJNS algorithm is that, in the former case, the robot will start its journey first and then search for the nearest node for shortest path and in the second case, the robot will first check for the nearest junction node in the present region and after proper detection it will start journey.

4.1.2 Pesudocode for Inter-Junction Node Search (IJNS)

```
IJNS
Fetch the regions(rm);
Fetch the residing region of destination node;
for q:= 1 to x
      N_x(l) = N_p(n);
      D_j = | N_p(n) − N_q(n) |
      if D_j == minimum
          save D_j ;
          D_S2 = D_j + D_S1 ;
          Save D_S2 ;
      else
          q++;
          repeat;
end for;
```

Notations used:

D_j =interjunction shortest distance

$N_p(n)$ =Agent present position

$N_q(n)$ =Destination junction node

Once the robot or the agent finds the destination junction by applying IJNS algorithm, it will move to that junction node from its present junction node. The distance in between its actual home node to the destination junction node is saved in the variable D_{S2}. This algorithm is required for detecting the shortest distance in between two regions.

4.1.3 Pseudocode for Destination Junction node to Destination Node Search (DJDS)

```
DJDS
Read destination node address;
if  N_d(n) != N_q(n)
    for t:= m to 0
            t++;
            N_q(n) = N_fd(n);
            D_djnc = | N_fd(n) − N_d−t(n) |;
            N_d−t(n) = N_fd(n);
            D_djnc = D_int;
    repeat;
    D_int =  D_S3;
    Save path;
    D_short = D_S2 + D_S3;
    Save nodes;
    Save route;
else if  N_d(n) == N_q(n)
             D_short = D_S2;
end for;
```

Notations used:

$N_{fd}(n)$ = Robot present position (Destination Junction node)

$N_d(n)$ = Destination node

D_{djnc} = Distance between present position to next intermediate node

D_{int} = Variable to store D_{djnc}

D_{short} = Total shortest length between robot/agent home node and destination node

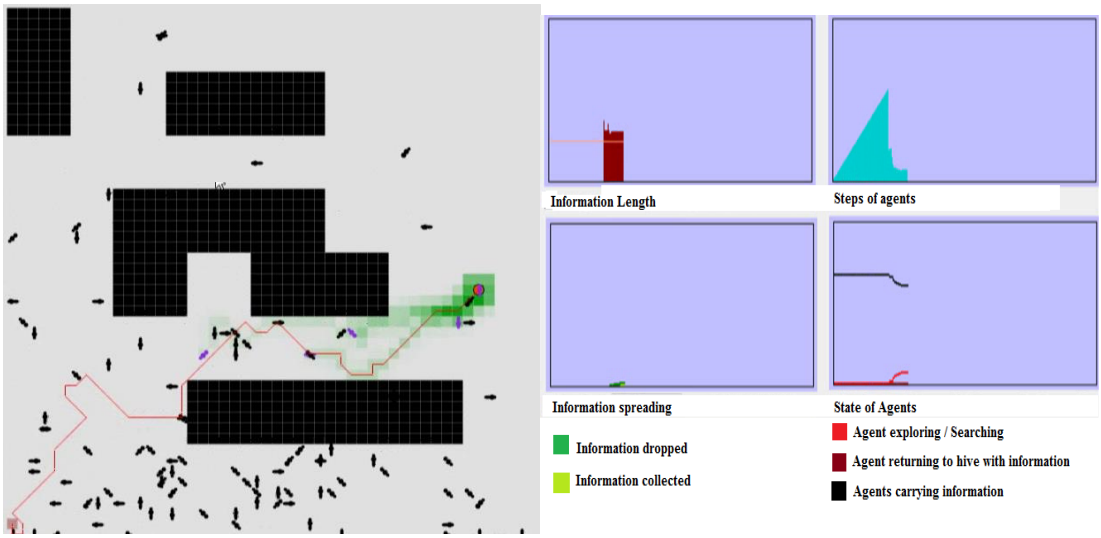
The agent or the robot can find the shortest unobstracted path in between its home node and the destination node.After getting the shortest path route, the route is saved in the momery as it will have to go back in the same route to reach to the home node.

5. Simulation Result

The simulation is designed for the Robot Colony of 200 agents. The simulation is based on the above mentioned algorithms which actually obeys the Kruskal's Algorithm. In this simulation, the agents starts moving the home node and reach for the preferred location or node where the object is placed. In this simulation, the agaent run is designed from the home node to node 8. The path is designed without obstacle and with obstacle seperately. Here, 500 iteration or steps are shown both for the path with obstacle and without obstacle to observe the agent routing in the colony. The thin red line indicates the shortest path among all the possible path from the home node to node 8.

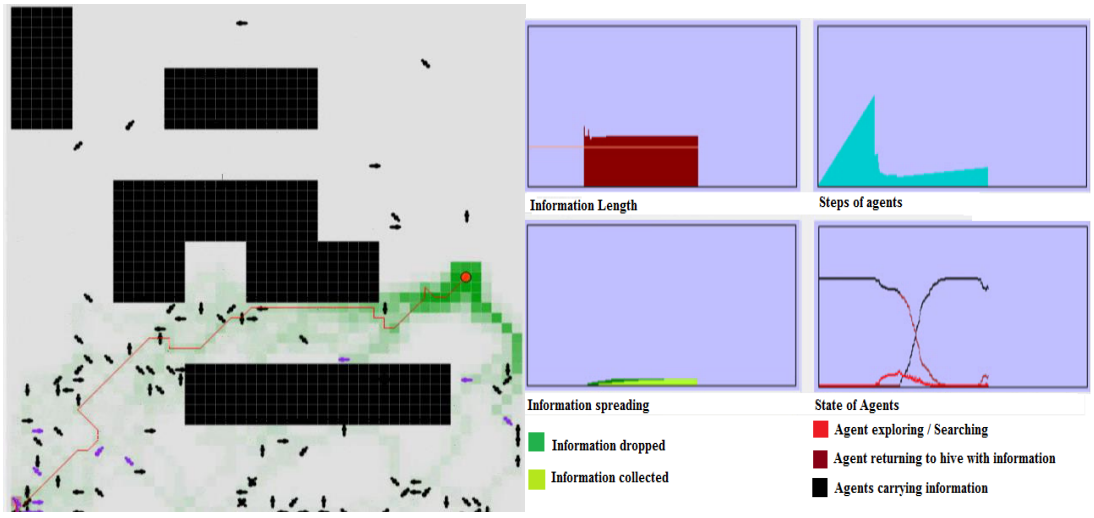
While moving from the home node to node 8, they follow and travel through the shortest path and simultaneously save the total path with the total weight as they tends to return back from node 8 to home by following the same path. Some of the agents will move towards node 8, some of them returning with information towards home node and rest are carrying the information and are wandering about the colony. All the above mentioned matters are shown in the simulation.

5.1 Simulation for the path without obstacle



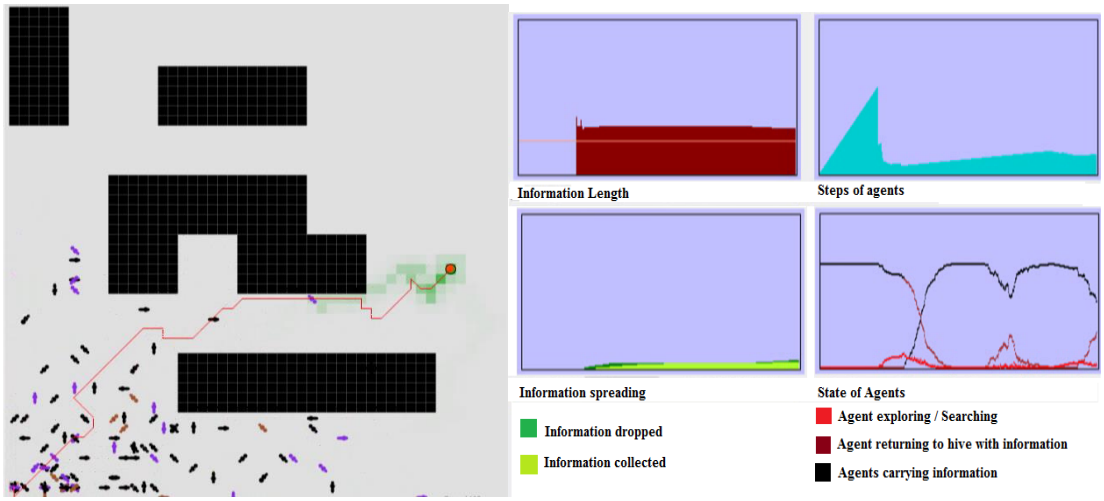
(a) (b)

Figure 10. Agents start moving (Iteration 50): (a) Agent moving in colony towards node 8; (b) Routing statistics



(a) (b)

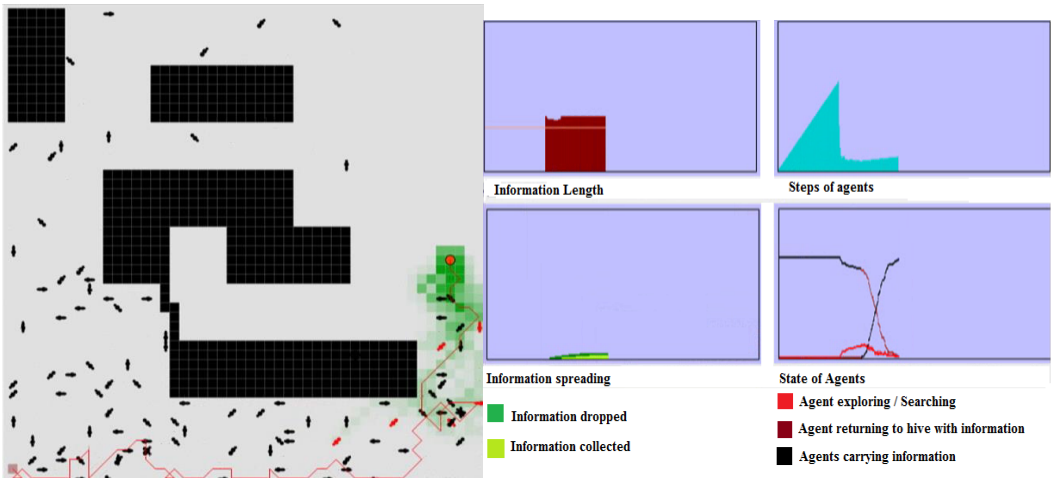
Figure 11. Agents collecting information (Iteration 250): (a) Agents collecting information form node 8 and some are returning; (b) Routing statistics



(a) (b)

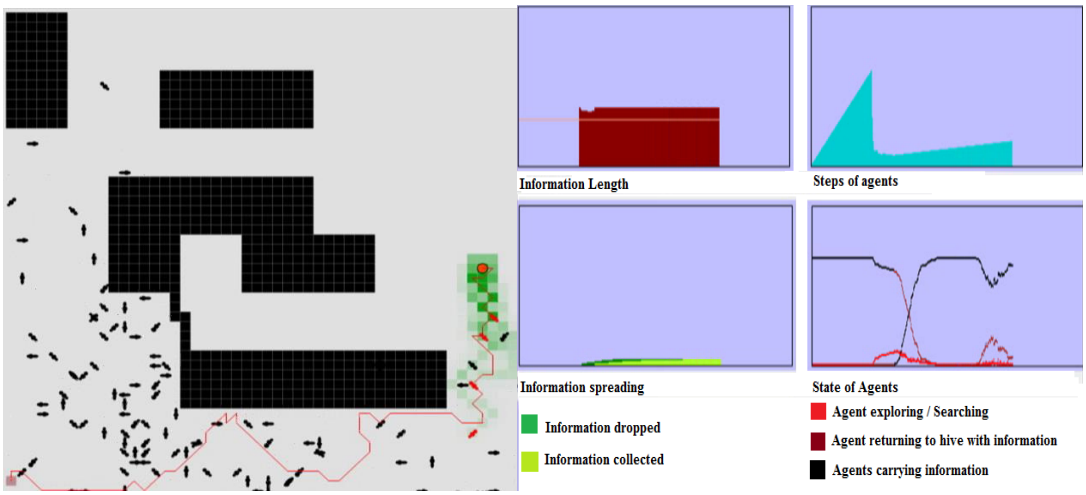
Figure 12. Involvement of more agents for information collection (Iteration500): (a) More agents are collecting information by creating deeper information path; (b) Routing statistics

5.2 Simulation for the path with obstacle



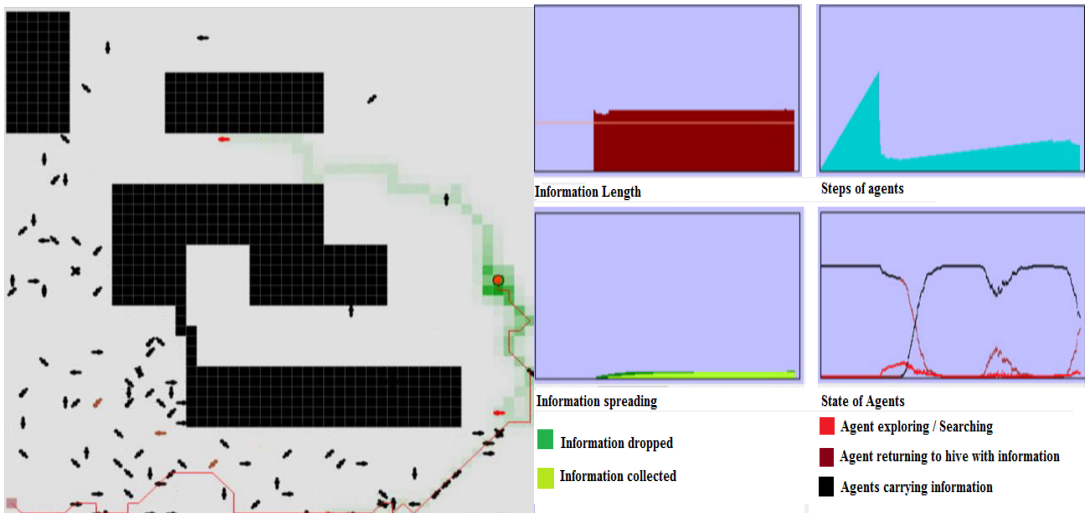
(a) (b)

Figure 13. Agents start moving (Iteration 50): (a) Agent moving in colony towards node 8 through alternative path; (b) Routing statistics



(a) (b)

Figure 14. Agents collecting information (Iteration 250): (a) Agents collecting information form node 8 and some are returning through alternative path; (b) Routing statistics



(a) (b)

Figure 15. Involvement of more agents for information collection (Iteration500): (a) More agents are collecting information by creating deeper information path; (b) Routing statistics

The simulations are done seperately for the path with obstacle and without from the home node to node 8. If the simulation result is compared with the RMPM in Figure 4, it can be found that for reaching to node 8, the agents are fllowing the shortest path. In RMPM, the shortest path route from home node to node 8 is 1→22→17→10→11→13→8 .In between the nodes of the route, the green highlighted nodes, *i.e.*, node 17, node 10 and node 13 are the junction nodes. The weight of the shortest path of the mentioned route is 8.828 unit. But, when the obstacle is placed on the mentioned shortest route, the agents then are following the alternative shortest route form home node to node 8.

The alternate shortest path that is defined in RMPM is 1→9→16→8 of weight 10 unit.In betwwen the mentioned nodes, node 9 is the junction node. This is defined in the RMPM and again matched with the simulation results from Figure 13 to Figure 15.

So, in this way, the agent robot will be routed by following the shortest path for reaching to the destination node to collect information and coming back to the home node. The features of Proposed RMPM are matched with the simulation.

6. Conclusion

The UNCD is designed on the basis of JCD which introduces a new concept of finding the shortest path in between two nodes by means of junction node connectivity. In the previous work, the NCD was introduced which is able to find the shortest path in between two nodes but the computational time was larger when compared to UNCD. UNCD reduces the computational time of shortest path finding as the agents are first routed to the nearest junction node of the source region and then they are moved to the nearest junction node of the region where the destination resides. The MNJ algorithm and MSP algorithm based on MACA, proposed in this paper to perform the efficient searching operation in the RMPM. The RPM is modified to design the RMPM, where the total area of the colony is segmented into some region so that the routing operation of the agent is much more fast and easier compared to that earlier introduced design. The simulation result shows that the actual shortest path of the agent satisfies the calculation of the above two algorithms. These designs are also helpful to solve the Travel Salesman Problem (TSP) as the agents are routed in the way that they will not travel a node twice while moving from the source to destination.

Acknowledgements

A modified approach of the robot colony system is introdued in this paper using the kruskal's algorithm and thereafter the design of the JCD and UNCD with help of the MACA algorithm are helpful for the design of the Robot Colony system and agent routing. The advanced approach of the shortest path is introduced in this paper, that is the nearest junction node connectivity by replacing the direct node-to-node connectivity. The NJNS, IJNS and DJNS efficiently can produce the simulation result which are shown above in the simulation section. The idea of the 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] S. Chakraborty, "Ant Colony System: A New Concept to Robot Path Planning", International Journal of Hybrid Information Technology, ISSN: 1738-9968 Copyright @ 2013 SERSC, vol. 6, no. 6, **(2013)**, pp. 11-30.

[5] 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.

[6] 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.

[7] 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.

[8] M. Dorigo, G. Di Caro and L. M. Gambardella, "Ant Algorithms for Discrete Optimization", Artificial Life, vol. 5, **(1999)**, pp. 137–172.
[9] M. Dorigo, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, **(1997)**.
[10] 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.
[11] 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.
[12] E. Foundas and A. Vlachos, "Pheromone models in ant colony optimization (ACO)", Journal of Interdisciplinary Mathematics, vol. 9, no. 1, **(2006)**, pp. 157-168.
[13] 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.
[14] 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.
[15] 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.
[16] 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.
[17] 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.
[18] 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.
[19] Z. Liu, M. Z. Kwiatkowska and C. Constantinou, "A Swarm Intelligence Routing Algorithm For Manets".
[20] 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".
[21] 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, Robotics, Artificial intelligence, Microprocessor .

