International Journal of Hybrid Information Technology
Vol.8, No.3 (2015), pp.213-234
http://dx.doi.org/10.14257/ijhit.2015.8.3.20

# Ant Colony System: An Approach for the Design of a Single Database to Establish Any Node-to-node Connectivity for Robot Path Planning in a Robot Colony

Subhadeep Chakraborty

*Software Analyst and Programmer Sons India Software Private Limited*
*subha.journal@gmail.com*

***Abstract***

*In Ant Colony System, the agents or the ants travel in search of foods by following one another that is they show the swarming behavior along the searching path of the food source and return path towards the hive. In the artificial ant system or the robot system, the same operation of the robot can be observed by the suitable design of the travelling path. The path design and the travelling technique are already proposed in my two previous works. Those designs include different types of algorithms for different opeartions. Kruskel's algorithm is applied to find the shortest path in between any two nodes, one is the source and another is the destination node. But the disadvantage of this design is the longer simulation time as three algorithms is required to travel from one node to another node and naturally several numbers of the databases are required to hold all the information about the entire robot colony. In this paper, a single database is proposed that can hold all the connectivity and which can be driven with a single algorithm. The proposed method is compared with the other previous techniques of finding the shortest path and the result set represented here showing the efficiency of the proposed method.*

**Keywords:** *Kruskal's Algorithm, Shortest path, Swarming, Ant Colony System, Node-to-Node Connectivity Database, Robot Colony*

## 1. Introduction

The Robot Colony System (RCS) is the very useful application of the ACS where the agent robots are designed similar to the behavior of the ants in the ACS. The ants travel on the searching path by placing the pheromone which is actually a hormone that have a specific smell that helps the rest ants to travel on the same way to food source or towards the hive [1, 3, 8, 9]. The path of the robots in the robot colony can be designed with the similar concept with a suitable replacement of the pheromone concept such as the magnetic line or some other else. That is robot can move forward with help of connected path in between the source and destination node [5, 8]. But there obviously a technique of finding the shortest in between two locations as because there is a probability for having different paths in between a source and the destination. In the robot colony, there may be some obstacles in the robot travelling path. While the robot agents are travelling in the RCS with the knowledge of shortest path [1-3, 31, 32], there must be some means to detect and overcome the obstacles so that they can move smoothly and can achieve the goal. So, the entire system must have the description of all node connections including the blocked areas which can be defined as the obstacles [9, 20, 21, 36]. This paper deals with the connected network among all nodes. So there are some algorithms are required to keep all the definitions of the nodes and their connections. In this paper a new approach is introduced which includes a single database and is capable of holding all the information

ISSN: 1738-9968 IJHIT

and driven by the algorithms which in turns helps to process the entire system by reducing time.

## 2. Ant System and Shortest Path: Previous Works

Ants are the creature of cooperation in their colony to perform a specific task that is they are swarms [1, 2, 19]. Robot is a specific idea taken from the swarming of ants that is robot or mobile robot is the artificial agents [1, 9, 20]. Robot or mobile robot can be so referred as artificial ants [20]. In Ant System, ants move forward by smelling the pheromone which is placed on the path by their previous ant. It is not that the trail is created in a single way but the pheromone is placed in different way to create different travelling path. But the following ant always follows the specific path which has the pheromone with highest concentration. Highest pheromone concentration means that most of the previous ants have visited or used the path that is the specific path is actually the most visited path. The pheromone is evaporated after sometime and if the total pheromone is evaporated, the trail is vanished [4, 8, 21, 33, 36]. The ant creates different paths with different probabilities to find the shortest path in between them. Some of the path is vanished due to evaporation and some path sustained due to ant transportation. After some iteration, they can find the shortest path from hive to food source and then most of the ants follow a specific path and for this reason this path is much more concentrated with pheromone. In this way the ants can find the shortest path. On the basis of this idea, the path of the robot can be designed. For this purpose, different algorithms are available for finding the shortest path in between two nodes in a colony or tree.

M. Dorigo have introduced the idea of the ant colony system through which the path of the mobile robots can be designed [2]. Later M. Dorigo, *et al.*, [1, 3, 7, 20, 25] have reviewed their work and proposed different algorithms to design a proper path plan for the travelling agents. N. B. Sariff and O. Buniyamin [4] have proposed some different ideas on robot path planning based on ant system in global static environment. N. Buniyamin, *et al.*, [5] proposed a variation of ant colony system algorithm. Zakzouk, *et al.*, [10], J. Sims, *et al.*, [22] Hojjat, *et al.*, [14] have proposed an approach for solving the shortest path problem and the routing system. There are different theories and algorithms available to find the shortest path in a graph. Again the idea of graph can be applied to the colony of the mobile robots to find the shortest path. In this paper, the shortest path can be found by using the kruskal's algorithm [6]. Young, *et al.*, [11] proposed an improved ant colony algorithm for robot path planning. Garro, *et al.*, [12], Seung, *et al.*, [13] and Chen, *et al.*, [16] have proposed a different and improved approach for the mobile robot path planning in preference based technique. In Ant System, it is essentially required to design a path that is dynamically available may be in network or locally. Xioping, *et al.*, [17], H. Mei, *et al.*, [18] proposed an approach for the optimal path planning of the mobile robot in dynamic environment. .The agents with swarming nature must support the cooperation in between all the other members while they are moving towards something to achieve a goal. So, ant, as a swarm agent and likewise the mobile robots which are designed with help of the ant behavior, will obviously support the cooperation among all the agents who are involved in a particular work which can be successfully finished by a team play. M. Dorigo, *et al.*, [7] and Behera, *et al.*, [19] have proposed a cooperative learning approach for the path design of the mobile robots. Foundas, *et al.*, [21] works and proposed the pheromone model of the ant system which is actually the idea behind the path planning of the mobile robot.

In my two previous work [8, 9], the path map have been designed which includes the general nodes and the junction nodes. The robot path map is designed to describe all the nodes and their locations [8]. The nodes are interconnected. The path map modified to Robot Modified Path Map where all the descriptions remain same but all the colony area is divided into some segments where in each segment there is at least one junction node [9]. MSP algorithm describes the total shortest path in between any two nodes with help of two different databases that are JCD and UNCD. But keeping two different databases with several different algorithms it may take a longer simulation time and as a result the processing seems to be longer. To reduce the probability of enlarging the processing time and make the system faster enough, the number of databases will have to be shortened and thereby reducing the number of algorithms by combining them to form less number of algorithms which can run on the single database. In this paper, single databases with required algorithms are represented for the design of the faster swarm robots or mobile robots in the robot colony.

## 3. Ant System: Robot Colony

### 3.1 Real Ant System

In real ant colony, the typical behavior of the ants is to find the food source and to travel toward the food source from hive and after successful collection of the food; they go back to their hive to store the food. To perform the operation, some informer ants first starts journey to find the food and when they find the food source they informs the other ants about that and the rest ants starts journey to find the food. While the traveling time the ants continuously smell the pheromone, actually a hormone placed by the former ants, and they follow the route which have the highest concentration of the pheromone [1, 8, 9]. This is the actual strategy of the ants in their entire colony system to collect the foods.

The routing process of the ants from the hive to food source to hive is demonstrated below in Figure 1.

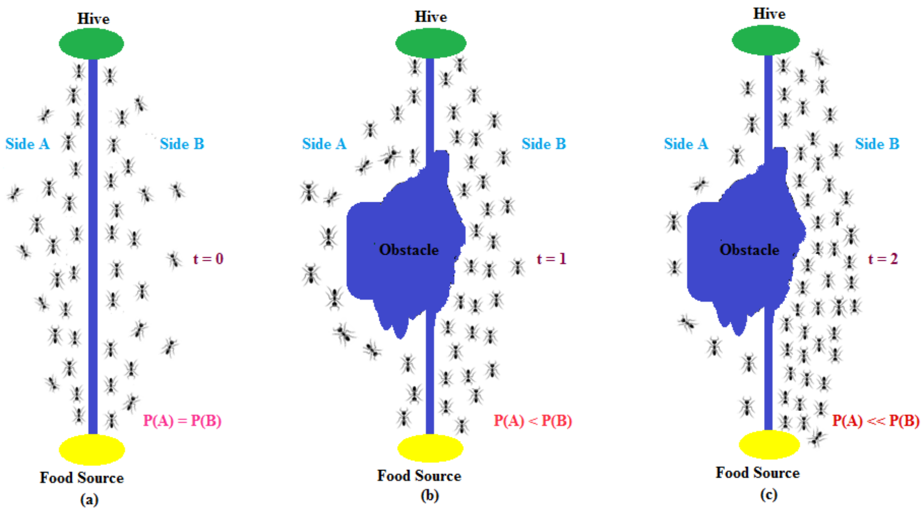



**Figure 1. Traveling Path of Ants in Ant Colony**

**(a) Travels Towards Hive to Food Source [t=0, $P(A) = P(B)$]**
**(b) Ants Start Gathering in Side B after Facing Obstacle [t=1, $P(A) < P(B)$]**
**(c) Large Number of Ants Populated at Side B [t=2, $P(A) \ll P(B)$]**

Ants follow the State Transition Rule while moving from source to destination. Ant system can be well defined by the state transition as they travel from one place to another to collect food [9, 16]. The state can be defined by the source and the destination and while they are moving from source to destination, the place or node change can be defined as the transition [1, 8, 20]. The State Transition Rule for a ant to travel from one node $n_i$ to another node $n_j$ is defined as follows:

$$n_j = \begin{cases} argmax_{u \in S_{ant}(n_i)} \left\{ \left[\tau(n_i, n_j)\right] \bullet \left[\eta(n_i, n_j)\right]^{\beta} \right\} & if\ q < q_0 \quad \ldots\ldots(a) \\ N & Otherwise \quad \ldots\ldots(b) \end{cases}$$
…..(1)

Where, $\tau(n_i, n_j)$ = pheromone information
$\eta(n_i, n_j)$ = heuristic information
$\beta$ = heuristic coefficient
$q$ = 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)
$S_{ant}(n_i)$ = list of all nodes to be visited by the ant
$N$ = random variable shows the probability of an ant in node $n_i$ to visit node $n_j$

Now, while the ants are traveling in the colony, the pheromone is placed by them on the way to their respective destination. But after sometime the pheromone placement process is updated by the evaporation and replacement of the pheromone on the traveling route [1, 8, 9]. Actually the pheromone creates a trail for the following ants to make the movement easier. The pheromone trail is updated by the following equation [1, 2, 8, 9]:

$$\tau_{n_i,n_j}(T_{ant_{n+1}}) \leftarrow (1-\rho)T_{ant_n}$$
…….(2)

Where,
$\tau_{n_i,n_j}$= amount of pheromene deposited on the path from $n_i$ to $n_j$
$T_{ant_{n+1}}$= New trail
$T_{ant_n}$ = Old trail
$\rho$ = Pheromone evaporation coefficient

### 3.2 Robot Colony: Swarm Robot

The Robot Colony can be designed with help of the idea of real ant colony system. Robot, the artificial ant agent ($agnt$), will move from one point or node to state to another point or node or state with the influence of the former agent ($agnt$) and obviously will follow the state transition rule as the real ants do. As the paper deals with the connected network, the shortest path in between two nodes anywhere within the colony can be found with help of Kruskal's algorithm [8]. With help of the Kruskal's algorithm the shortest path can be found in between two nodes without visiting the same intermediate node twice which actually defined by the TSPs [8, 9, 28, 32]. The state transition rule for the mobile robot can be defined as follows [9]:

$$n_j = \begin{cases} argmax_{v \in S_{agnt}(n_i)} \left\{ \left[\tau(n_i, n_j)\right] \bullet \left[\eta(n_i, n_j)\right]^{\beta} \right\} & if\ q < q_0 \quad \ldots\ldots(a) \\ N & Otherwise \quad \ldots\ldots(b) \end{cases}$$
…(3)

Where,
$S_{agnt}(n_i)$ = List of all nodes to be visited by the agent
$N$ = random variable chosen based on Eq. 4 that shows the probability of an ant in
node $n_i$ to visit node $n_j$ .

The movement of this particular agent ($agnt$), moving from node $n_i$ to node $n_j$, can be defined in the following Equation [1, 8, 9],

$$P_{agnt}(n_i, n_j) = \frac{[\tau(n_i,n_j)]\bullet[\eta(n_i,n_j)]^{\beta}}{\sum_{X\epsilon S_{agnt}(n_i)}[\tau(n_i,n_j)]\bullet[\eta(n_i,n_j)]^{\beta}}, \quad \text{if } X\epsilon S_{agnt}(n_i) \quad \ldots\ldots\ldots(4)$$

Where,

$P_{agnt}(n_i, n_j)$ = Probability of an agent ($agnt$) to move from node $n_i$ to node $n_j$.

While the movement process of the mobile robot or the swarm robots, the agent may or may not face obstacles. If the agent faces obstacle while traveling from one node to another, the possible measure is described by the database that is preloaded in it. The main objective is to move the agent from one node to another node through the junction nodes, if required. Thus in the robot colony, the swarm robots can roam around the entire area of nodes.

### 3.3 Comparison: Real Ant and Swarm Robot

The term Swarm means to follow one another to perform some specific task particularly by team play. Real ants are the swarming creature as they follow their former while moving from source place to destination place. The source or destination place may be the hive or food source depending upon their forward move or reverse move [9]. While their movement, they follow the State Transition rule and the Pheromone Update Rule [8, 9]. The Robot or the Mobile Robots can be activated or automated by giving it some specific instruction so they can move from one node to another to collect some information and can adapt the environment. While traveling, the Swarm Robots will be capable of finding the shortest path in between two nodes like the real ants do. So in true sense, the design of the robot colony must be based on the technique that is applied by the real ants. The main difference of the robot colony with that of the real ant colony is the movement of the ants is performed by the smelling of the pheromone and the movement of the artificial agent or the mobile robots is performed by the means of connected lines or magnetic line or some other means with help of signaling.

## 4. Advanced Robot Colony System

The design of the RCS can be organized with help of the Modified Ant Colony Algorithm (MACA) that is already defined in my previous work [9]. Actually this algorithm is very helpful, less time consuming thus efficient to design the RCS. MACA is again dependent on RPA-1 and RPA-2 route algorithms. These three algorithms together can form a suitable RCS with help of a path describing map, i.e. RMPM which includes all the definition of the route and routing procedures of the artificial agents [9, 35]. The MACA algorithm can work for the connected path but the RMPM can work both for connected path and connectionless path. In this paper, the single database is introduced with help of RMPM and MACA for connected path.

### 4.1 Modified Ant Colony Algorithm

The Modified Ant Colony Algorithm is shown in Figure 2

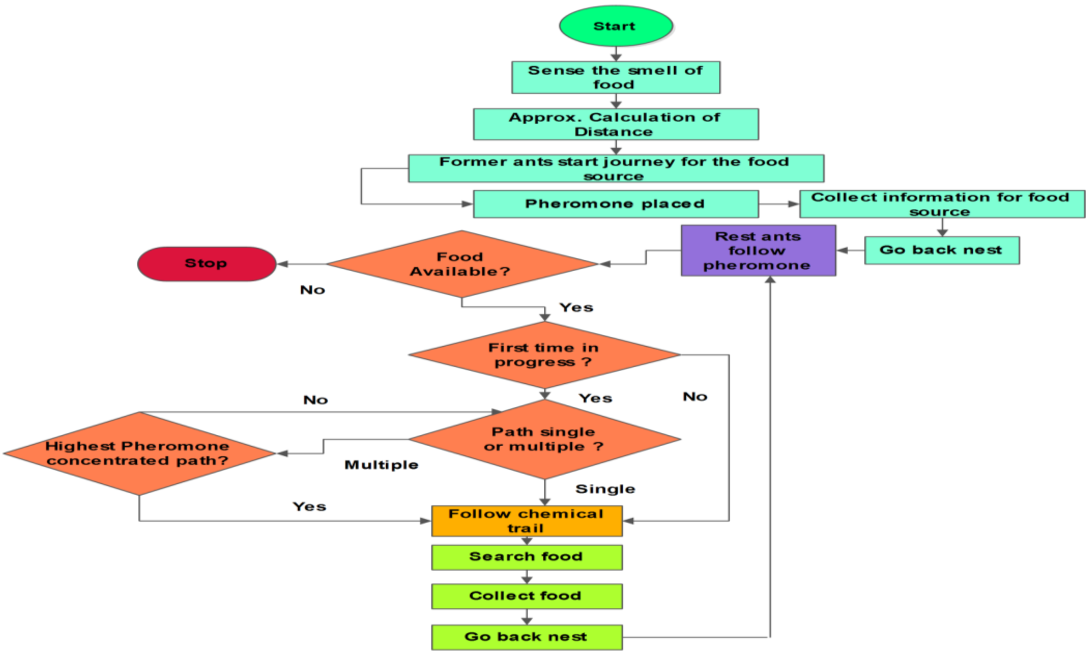



**Figure 2. Modified Ant Colony Algorithm**

Basically, MACA is designed with help of the Kruskal's Algorithm which is actually an efficient and less time consuming algorithm which can find the shortest path accurately for the mobile robots or artificial agents [9]. The next sub section describes the Robot Colony Algorithm (RCA) which is designed using Modified Ant Colony Algorithm.

### 4.2 Robot Modified Path Map

The Robot Modified Path Map (RMPM) is actually a route database that holds all the connectivity in between the general nodes and the junction nodes. The entire colony is segmented into some parts to make the operation easy. The main objective behind the segmentation of the colony is that the possibility of route engagement is reduced by means of bypassing the route of the second agent through some other junction nodes to another region. The RMPM is shown below in Figure 3

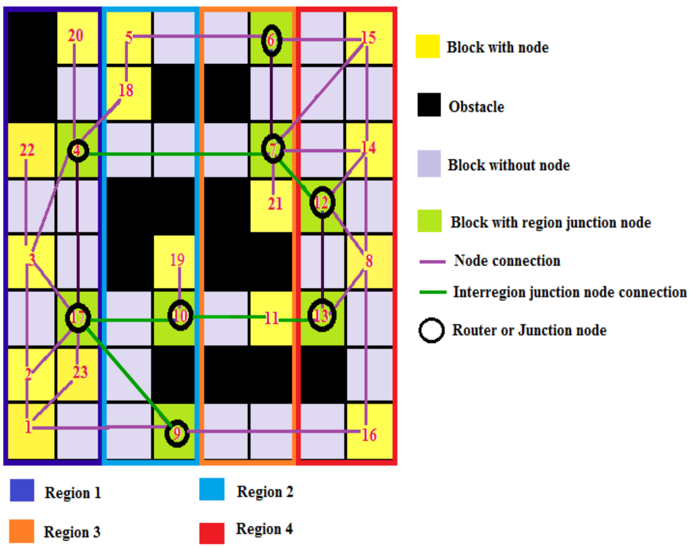



**Figure 3. Robot Modified Path Map (RMPM)**

In RMPM, the entire colony is segmented into four regions and they are Region1, Region 2, Region 3 and Region 4. There is a slight modification of RMPM with that of described in the previous paper [9], that is previously there is a default home node available in RMPM from where the Robot will start the journey but now the default home node is deducted as because the Robot can start journey from anywhere and any node and can stop its journey at any node. So there is no meaning of having default home node in RMPM. The default distance between two centers of two side-by-side block is 1 unit [8, 9]. The unit block is shown below in Figure 4.

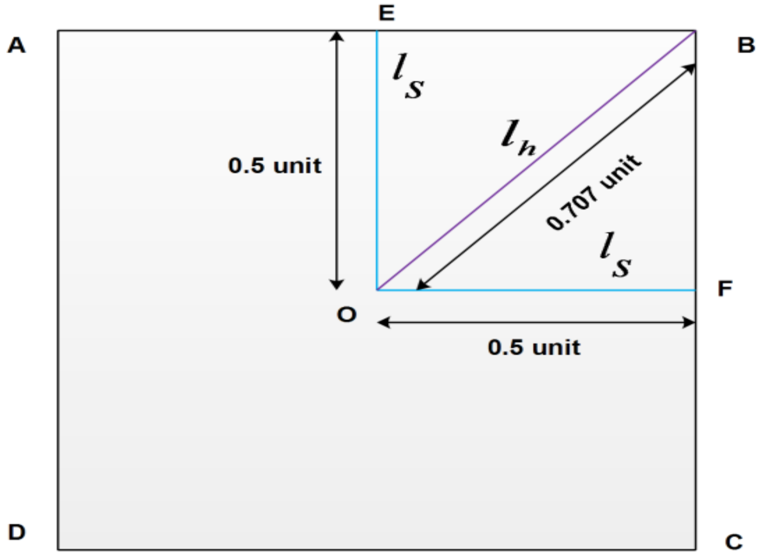



**Figure 4. Unit Block or RMPM**

### 4.3 Robot Colony Algorithm

The Robot Colony Algorithm describes how the artificial agents can move from one node to another may be in same region or inter region. First, the agents check for the node to visit along with the availability of the destination node prior to check the present node index. After the check, the agents start checking the visibility of the destination node. The visibility can be defined as the inverse of the distance that is,

$$\Upsilon_{i,j} = {}^{1}/_{\ell_{i,j}}$$

………(5)

Where,

$\Upsilon_{i,j}$= Visibility of the destination node (node *j*)

$\ell_{i,j}$ = Distance between node $n_i$ and node $n_j$.

Obviously every agent have a finite visibility and if the limit is crossed, the agent will check for the next node and the junction node, if required. Let the limit be $\mathbb{V}$. So, if the visibility of the destination node is within $\mathbb{V}$, the agent smoothly can reach to that node but if the limit crosses, the agent will iteratively check for the next node, may be general or junction, until and unless the destination node comes within the visibility range ($\Upsilon_{i,j} \leq \mathbb{V}$). So, the movement of the mobile robots is entirely dependent on the visibility of the destination node.

The Robot Colony Algorithm is demonstrated below in Figure 5

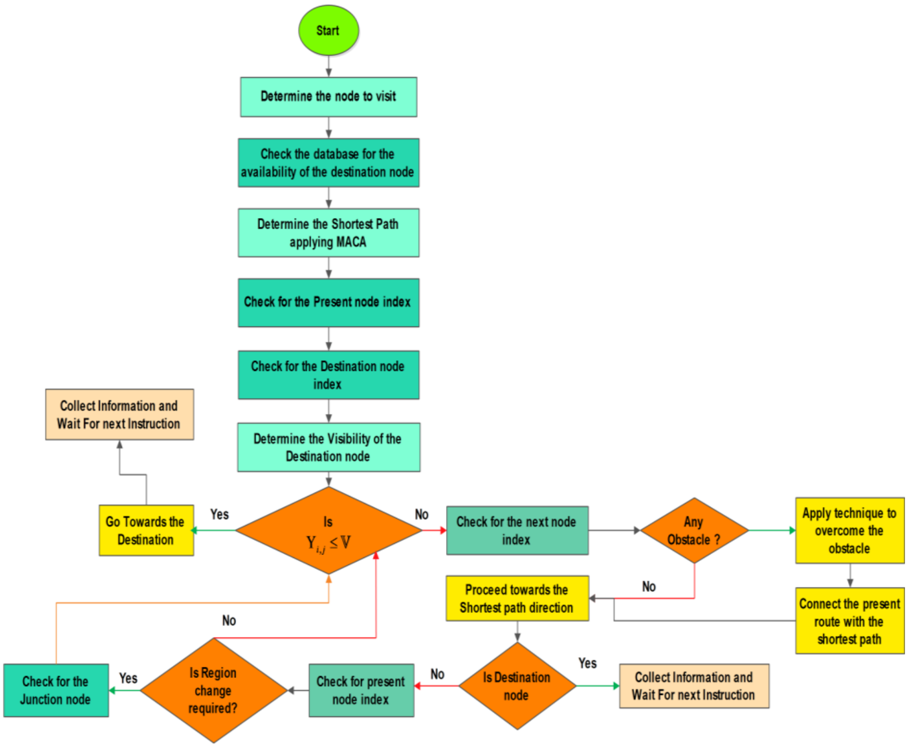



**Figure 5. Robot Colony Algorithm (RCA)**

In this algorithm, when the visibility is beyond $\mathbb{V}$, the agent will check for the next node with shortest edge from the predefined database and also check for any obstacle is there or not. The technique to overcome the obstacle is to find the nearest node through which the route can be diverted. Now, if the destination belongs to the same region to that of the source node, no junction node is need to be used but if the region change is required, the agent must pass through the junction node to reach to the next junction. So, in this technique, the artificial agents can roam around the colony by following the nodes and the instruction and by overcoming the obstacle. The entire information about the nodes, junction and the regions are defined in the RMPM described earlier [9].

## 5. Routing of Robots in RCS

Routing stands for roaming from one place to another with certain process and direction. The term direction comes because in RCS it is desired to not visit the same node twice and so a certain direction is preferred to overcome this type of problems typically seen in TSPs. The routing of the Robots in RCS requires a process and a database which can hold the data. The database is dependent on the RMPM to read the connectivity of the nodes in RCS. The distance in between two nodes can be calculated with the help of RMPM, JCD and UNCD [9]. JCD is used to calculate the distance in between two junction nodes of two regions or in a single region as any region may contains multiple junction nodes. UNCD helps to calculate the distance between the junction nodes and the rest general nodes in the RCS [9]. In the previous two works, two different databases are created but it is true that the processing time is enlarged for reading two separate database which in turns efficiently holds all the required data and information about the node connectivity.

### 5.1 Routing Process

In the previous work, the agents are routed around the colony with help of MNJ and MSP algorithm [9]. In this paper a new algorithm, RCA is introduced which is capable of handling all the operations of both two algorithms. When the agents are moving from node $i$ to node $j$, they will obey the instruction given by RCA algorithm which is actually designed on the basis of MACA algorithm and the visibility of the agents. For each and every iteration, the agent must check for the present node index and the visibility to next nearest node. For traveling different region, they must appear to the regional junction node suitable to the agent as there may me multi junction node available in each region. In previous work, for the routing of the agent, MACA was applied twice which was taking more simulation time. But RCA algorithm will applied only once in the throughout journey from the source to destination. For each and every movement, the agent will continuously check and read the instruction from RCA and obviously check for the visibility as it is the most important factor in the domain of finding the shortest path and reaching to the desired destination. When the agent starts its journey from node $i$ to node $j$, they move through the shortest path and reach to the desired location and after collecting the information from the destination, it will stop moving until it will get the another instruction to move to another node. While moving, the agent will always read a single database, described in the next subsection, to achieve the proper information about the node connectivity.

### 5.2 Node-to Node Connectivity Single Database (NNCSD)

This is an integrated database which holds all the information regarding the connectivity in between the nodes and the weight in between nodes. The database contains the information about the junction nodes and general nodes as well rather than by keeping it separately. NNCSD removes the limitations of JCD and UNCD where two different types of data are stored in two separate database. The NNCSD is shown in Figure 6. The NNCSD contains all the weight values of the all possible distances in between any two node. That is, the connectivity and the value of the distances are described in the database.

NNCSD is actually an $n \times n$ matrix where $n$ is the number of nodes present in the colony. As the colony consist of 23 nodes, so there will be 529 number of distances. But somewhere it is found that in the matrix, same nodes interconnects and the distance in between the same node will be “Zero” and is denoted by “------”. The interconnections in between the junction nodes, found in the RMPM, are marked by pink color and the junction nodes itself are colored with Yellow. From the NNCSD, distance between any two nodes can be found in form of Distance Weight representation. When the user will give and input for the destination node to move, the agent will first fetch the data from RMPM and NNCSD accordingly and after finding the shortest path, it will move to the destination node. The database is shown below.

| i/p | Source Node | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Destination Node | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 1 | — | $W_{1,0}$ | $W_{3,0}$ | $W_{3,1}$ | $W_{4,2}$ | $W_{7,2}$ | $W_{7,1}$ | $W_{6,2}$ | $W_{1,3}$ | $W_{3,1}$ | $W_{5,1}$ | $W_{8,1}$ | $W_{6,1}$ | $W_{8,2}$ | $W_{7,3}$ | $W_{5,3}$ | $W_{1,1}$ | $W_{3,2}$ | $W_{4,1}$ | $W_{5,1}$ | $W_{8,1}$ | $W_{5,0}$ | $W_{0,1}$ |
| 2 | $W_{1,0}$ | — | $W_{2,0}$ | $W_{2,1}$ | $W_{3,2}$ | $W_{6,2}$ | $W_{6,1}$ | $W_{5,2}$ | $W_{0,3}$ | $W_{2,1}$ | $W_{4,1}$ | $W_{7,1}$ | $W_{5,1}$ | $W_{7,2}$ | $W_{6,3}$ | $W_{4,3}$ | $W_{0,1}$ | $W_{2,2}$ | $W_{3,1}$ | $W_{4,1}$ | $W_{7,1}$ | $W_{4,0}$ | $W_{1,1}$ |
| 3 | $W_{3,0}$ | $W_{2,0}$ | — | $W_{0,2}$ | $W_{1,3}$ | $W_{4,3}$ | $W_{4,2}$ | $W_{4,4}$ | $W_{0,3}$ | $W_{2,1}$ | $W_{4,1}$ | $W_{7,1}$ | $W_{5,1}$ | $W_{7,2}$ | $W_{4,4}$ | $W_{4,3}$ | $W_{0,1}$ | $W_{0,3}$ | $W_{3,1}$ | $W_{2,2}$ | $W_{5,2}$ | $W_{2,0}$ | $W_{1,1}$ |
| 4 | $W_{3,1}$ | $W_{2,1}$ | $W_{0,2}$ | — | $W_{1,1}$ | $W_{4,1}$ | $W_{4,0}$ | $W_{4,2}$ | $W_{3,2}$ | $W_{5,0}$ | $W_{7,0}$ | $W_{4,1}$ | $W_{6,1}$ | $W_{6,0}$ | $W_{4,2}$ | $W_{7,2}$ | $W_{3,0}$ | $W_{0,1}$ | $W_{6,0}$ | $W_{2,0}$ | $W_{5,0}$ | $W_{2,2}$ | $W_{4,0}$ |
| 5 | $W_{4,2}$ | $W_{3,2}$ | $W_{1,3}$ | $W_{1,1}$ | — | $W_{3,0}$ | $W_{5,0}$ | $W_{5,2}$ | $W_{4,3}$ | $W_{6,1}$ | $W_{8,1}$ | $W_{5,1}$ | $W_{7,1}$ | $W_{7,0}$ | $W_{5,0}$ | $W_{5,6}$ | $W_{4,1}$ | $W_{1,0}$ | $W_{7,1}$ | $W_{3,1}$ | $W_{6,0}$ | $W_{3,3}$ | $W_{5,1}$ |
| 6 | $W_{7,2}$ | $W_{6,2}$ | $W_{4,3}$ | $W_{4,1}$ | $W_{3,0}$ | — | $W_{2,0}$ | $W_{2,2}$ | $W_{4,6}$ | $W_{5,3}$ | $W_{3,3}$ | $W_{2,1}$ | $W_{4,1}$ | $W_{4,0}$ | $W_{2,0}$ | $W_{5,2}$ | $W_{7,1}$ | $W_{4,0}$ | $W_{6,3}$ | $W_{6,1}$ | $W_{3,0}$ | $W_{6,3}$ | $W_{8,1}$ |
| 7 | $W_{7,1}$ | $W_{6,1}$ | $W_{4,2}$ | $W_{4,0}$ | $W_{5,0}$ | $W_{2,0}$ | — | $W_{0,2}$ | $W_{7,2}$ | $W_{5,1}$ | $W_{3,1}$ | $W_{0,1}$ | $W_{2,1}$ | $W_{2,0}$ | $W_{0,2}$ | $W_{3,2}$ | $W_{7,0}$ | $W_{4,1}$ | $W_{6,1}$ | $W_{6,0}$ | $W_{1,0}$ | $W_{6,2}$ | $W_{8,0}$ |
| 8 | $W_{6,2}$ | $W_{5,2}$ | $W_{4,4}$ | $W_{4,2}$ | $W_{5,2}$ | $W_{2,2}$ | $W_{0,2}$ | — | $W_{7,0}$ | $W_{3,1}$ | $W_{1,1}$ | $W_{0,1}$ | $W_{0,1}$ | $W_{2,0}$ | $W_{4,0}$ | $W_{3,0}$ | $W_{5,1}$ | $W_{4,3}$ | $W_{4,1}$ | $W_{6,2}$ | $W_{1,2}$ | $W_{6,4}$ | $W_{6,1}$ |
| 9 | $W_{1,3}$ | $W_{0,3}$ | $W_{0,3}$ | $W_{3,2}$ | $W_{4,3}$ | $W_{4,6}$ | $W_{7,2}$ | $W_{7,0}$ | — | $W_{2,2}$ | $W_{4,2}$ | $W_{7,1}$ | $W_{7,1}$ | $W_{9,0}$ | $W_{11,0}$ | $W_{4,0}$ | $W_{0,2}$ | $W_{0,6}$ | $W_{3,2}$ | $W_{5,2}$ | $W_{8,2}$ | $W_{2,3}$ | $W_{1,2}$ |
| 10 | $W_{3,1}$ | $W_{2,1}$ | $W_{2,1}$ | $W_{5,0}$ | $W_{6,1}$ | $W_{5,3}$ | $W_{5,1}$ | $W_{3,1}$ | $W_{2,2}$ | — | $W_{2,0}$ | $W_{5,0}$ | $W_{3,0}$ | $W_{5,1}$ | $W_{7,1}$ | $W_{6,1}$ | $W_{2,0}$ | $W_{5,1}$ | $W_{1,0}$ | $W_{7,0}$ | $W_{6,1}$ | $W_{4,1}$ | $W_{3,0}$ |
| 11 | $W_{5,1}$ | $W_{4,1}$ | $W_{4,1}$ | $W_{7,0}$ | $W_{8,1}$ | $W_{3,3}$ | $W_{3,1}$ | $W_{1,1}$ | $W_{4,2}$ | $W_{2,0}$ | — | $W_{3,0}$ | $W_{1,0}$ | $W_{3,1}$ | $W_{5,1}$ | $W_{4,1}$ | $W_{4,0}$ | $W_{7,1}$ | $W_{3,0}$ | $W_{9,0}$ | $W_{4,1}$ | $W_{5,1}$ | $W_{5,0}$ |
| 12 | $W_{8,1}$ | $W_{7,1}$ | $W_{7,1}$ | $W_{4,1}$ | $W_{5,1}$ | $W_{2,1}$ | $W_{0,1}$ | $W_{0,1}$ | $W_{7,1}$ | $W_{5,0}$ | $W_{3,0}$ | — | $W_{2,0}$ | $W_{0,1}$ | $W_{2,1}$ | $W_{3,1}$ | $W_{7,0}$ | $W_{4,2}$ | $W_{6,0}$ | $W_{6,1}$ | $W_{1,1}$ | $W_{6,3}$ | $W_{8,0}$ |
| 13 | $W_{6,1}$ | $W_{5,1}$ | $W_{5,1}$ | $W_{6,1}$ | $W_{7,1}$ | $W_{4,1}$ | $W_{2,1}$ | $W_{0,1}$ | $W_{7,1}$ | $W_{3,0}$ | $W_{1,0}$ | $W_{2,0}$ | — | $W_{2,1}$ | $W_{4,1}$ | $W_{3,1}$ | $W_{5,0}$ | $W_{6,2}$ | $W_{4,0}$ | $W_{8,1}$ | $W_{3,1}$ | $W_{7,1}$ | $W_{6,0}$ |
| 14 | $W_{8,2}$ | $W_{7,2}$ | $W_{7,2}$ | $W_{6,0}$ | $W_{7,0}$ | $W_{4,0}$ | $W_{2,0}$ | $W_{2,0}$ | $W_{9,0}$ | $W_{5,1}$ | $W_{3,1}$ | $W_{0,1}$ | $W_{2,1}$ | — | $W_{2,0}$ | $W_{5,0}$ | $W_{7,1}$ | $W_{6,1}$ | $W_{6,1}$ | $W_{8,0}$ | $W_{3,0}$ | $W_{9,2}$ | $W_{8,1}$ |
| 15 | $W_{7,3}$ | $W_{6,3}$ | $W_{4,4}$ | $W_{4,2}$ | $W_{5,0}$ | $W_{2,0}$ | $W_{0,2}$ | $W_{4,0}$ | $W_{11,0}$ | $W_{7,1}$ | $W_{5,1}$ | $W_{2,1}$ | $W_{4,1}$ | $W_{2,0}$ | — | $W_{7,0}$ | $W_{7,2}$ | $W_{6,0}$ | $W_{8,1}$ | $W_{8,1}$ | $W_{1,2}$ | $W_{8,3}$ | $W_{8,2}$ |
| 16 | $W_{5,3}$ | $W_{4,3}$ | $W_{4,3}$ | $W_{7,2}$ | $W_{5,6}$ | $W_{5,2}$ | $W_{3,2}$ | $W_{3,0}$ | $W_{4,0}$ | $W_{6,1}$ | $W_{4,1}$ | $W_{3,1}$ | $W_{3,1}$ | $W_{5,0}$ | $W_{7,0}$ | — | $W_{4,2}$ | $W_{7,3}$ | $W_{7,1}$ | $W_{9,2}$ | $W_{4,2}$ | $W_{6,3}$ | $W_{5,2}$ |
| 17 | $W_{1,1}$ | $W_{0,1}$ | $W_{0,1}$ | $W_{3,0}$ | $W_{4,1}$ | $W_{7,1}$ | $W_{7,0}$ | $W_{5,1}$ | $W_{0,2}$ | $W_{2,0}$ | $W_{4,0}$ | $W_{7,0}$ | $W_{5,0}$ | $W_{7,1}$ | $W_{7,2}$ | $W_{4,2}$ | — | $W_{3,1}$ | $W_{3,0}$ | $W_{5,0}$ | $W_{8,0}$ | $W_{2,1}$ | $W_{1,0}$ |
| 18 | $W_{3,2}$ | $W_{2,2}$ | $W_{0,3}$ | $W_{0,1}$ | $W_{1,0}$ | $W_{4,0}$ | $W_{4,1}$ | $W_{4,3}$ | $W_{0,6}$ | $W_{5,1}$ | $W_{7,1}$ | $W_{4,2}$ | $W_{6,2}$ | $W_{6,1}$ | $W_{6,0}$ | $W_{7,3}$ | $W_{3,1}$ | — | $W_{6,1}$ | $W_{2,1}$ | $W_{5,1}$ | $W_{2,3}$ | $W_{4,0}$ |
| 19 | $W_{4,1}$ | $W_{3,1}$ | $W_{3,1}$ | $W_{6,0}$ | $W_{7,1}$ | $W_{6,3}$ | $W_{6,1}$ | $W_{4,1}$ | $W_{3,2}$ | $W_{1,0}$ | $W_{3,0}$ | $W_{6,0}$ | $W_{4,0}$ | $W_{6,1}$ | $W_{8,1}$ | $W_{7,1}$ | $W_{3,0}$ | $W_{6,1}$ | — | $W_{8,0}$ | $W_{7,1}$ | $W_{5,1}$ | $W_{4,0}$ |
| 20 | $W_{5,1}$ | $W_{4,1}$ | $W_{2,2}$ | $W_{2,0}$ | $W_{3,1}$ | $W_{6,1}$ | $W_{6,0}$ | $W_{6,2}$ | $W_{5,2}$ | $W_{7,0}$ | $W_{9,0}$ | $W_{6,1}$ | $W_{8,1}$ | $W_{8,0}$ | $W_{8,1}$ | $W_{9,2}$ | $W_{5,0}$ | $W_{2,1}$ | $W_{8,0}$ | — | $W_{7,0}$ | $W_{4,2}$ | $W_{6,0}$ |
| 21 | $W_{8,1}$ | $W_{7,1}$ | $W_{5,2}$ | $W_{5,0}$ | $W_{6,0}$ | $W_{3,0}$ | $W_{1,0}$ | $W_{1,2}$ | $W_{8,2}$ | $W_{6,1}$ | $W_{4,1}$ | $W_{1,1}$ | $W_{3,1}$ | $W_{3,0}$ | $W_{1,2}$ | $W_{4,2}$ | $W_{8,0}$ | $W_{5,1}$ | $W_{7,1}$ | $W_{7,0}$ | — | $W_{7,2}$ | $W_{9,0}$ |
| 22 | $W_{5,0}$ | $W_{4,0}$ | $W_{2,0}$ | $W_{2,2}$ | $W_{3,3}$ | $W_{6,3}$ | $W_{6,2}$ | $W_{6,4}$ | $W_{2,3}$ | $W_{4,1}$ | $W_{5,1}$ | $W_{6,3}$ | $W_{7,1}$ | $W_{9,2}$ | $W_{8,3}$ | $W_{6,3}$ | $W_{2,1}$ | $W_{2,3}$ | $W_{5,1}$ | $W_{4,2}$ | $W_{7,2}$ | — | $W_{3,1}$ |
| 23 | $W_{0,1}$ | $W_{1,1}$ | $W_{1,1}$ | $W_{4,0}$ | $W_{5,1}$ | $W_{8,1}$ | $W_{8,0}$ | $W_{6,1}$ | $W_{1,2}$ | $W_{3,0}$ | $W_{5,0}$ | $W_{8,0}$ | $W_{6,0}$ | $W_{8,1}$ | $W_{8,2}$ | $W_{5,2}$ | $W_{1,0}$ | $W_{4,0}$ | $W_{4,0}$ | $W_{6,0}$ | $W_{9,0}$ | $W_{3,1}$ | — |

**Figure 6. Node-to Node Connectivity Single Database (NNCSD)**

In this database, all the distances in between any two nodes are stored. The database is actually designed in $n \times n$ matrix fashion where $n = 23$ that is a square matrix. The programmer will provide two input, one is for the present node and another is for destination node. The operation and the adaptive distance calculation are shown in the next section. When the programmer or the user provides two input for source and destination nodes respectively, the agent or mobile robot will first check the RMPM and NNCSD to determine the shortest route in between two nodes. There may be two or more route available for reaching to the destination node, but the robot will check the database for the shortest distance and find out the shortest route by matching the corresponding weight of lengths by Kruskal's Algorithm.

### 5.3 Calculation of Shortest Route

The mobile agent can find the shortest route in between the source node to the destination node. There must be no fixed source or home node. After the completion of $(r-1)^{th}$ work, the agent must hold any of the nodes and for starting the $r^{th}$ work, this node is the source or present node and will be in consideration. In the predefined RMPM, there are two type of paths, one is the Straight path$(d_s)$ and another id the hypotenuse path $d_h$. The general distance equation in between two nodes $a, b$ can be described as,

$$d_{a,b} = \sum_{\substack{i=0 \\ j=0}}^{n} d_{\ell_{s_i}, \ell_{h_j}} \qquad \dots\dots\dots (6)$$

Where,

$d_{a,b}$ = Distance between node $a$ and node $b$

$\ell_{s_i}$ = Straight length dependant on $i$ $(2 \times l_s)$
$\ell_{h_j}$= Hypotenuse length dependant on $j$ $(2 \times l_h)$

Now for reaching to node $b$ from node $a$ there may or may not be some intermediate nodes. The distance in between those intermediate nodes may be straight or may be hypotenuse in length. Now the calculation of the shortest route in between any two nodes is shown below.

Let the source node is $node$ 14 and the destination node is $node$ 17. There are several routes in between the $node$ 14 and $node$ 17 and they are:

**Table 1. Various Distances between $Node$ 14 and $Node$ 17**

| $a = node\ 14$ & $b = node$ 17 | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Sl. No.** | **Route No.** | **Route description (Node connections)** | **Length** | | | **Applicable for Shortest path** | **Distance Weight Representation** |
| | | | $d_1 = \ell_{s_i} \times 1$ | $d_2 = \ell_{h_j} \times 1.414$ | $d_{a,b} = d_1 + d_2$ | | |
| 1 | #1 | 14-8-16-9-17 | $9 \times 1 = 9$ | $2 \times 1.414 = 2.828$ | 11.828 | No | $W_{9,2}$ |
| 2 | #2 | 14-12-13-11-10-17 | $7 \times 1 = 7$ | $1 \times 1.414 = 1.414$ | 8.414 | Yes | $W_{7,1}$ |
| 3 | #3 | 14-8-13-11-10-17 | $7 \times 1 = 7$ | $1 \times 1.414 = 1.414$ | 8.414 | Yes | $W_{7,1}$ |
| 4 | #4 | 14-7-4-17 | $9 \times 1 = 9$ | $0 \times 1.414 = 0$ | 9 | No | $W_{9,0}$ |
| 5 | #5 | 14-15-6-5-18-4-17 | $11 \times 1 = 11$ | $1 \times 1.414 = 1.414$ | 12.414 | No | $W_{11,1}$ |
| 6 | #6 | 14-7-6-5-18-4-17 | $11 \times 1 = 11$ | $1 \times 1.414 = 1.414$ | 12.414 | No | $W_{11,1}$ |
| 7 | #7 | 14-12-7-4-17 | $7 \times 1 = 7$ | $2 \times 1.414 = 2.828$ | 9.828 | No | $W_{7,2}$ |
| 8 | #8 | 14-7-4-3-17 | $6 \times 1 = 6$ | $3 \times 1.414 = 4.242$ | 10.242 | No | $W_{6,3}$ |

When the mobile robot will be instructed to go to the node 17 where the present location or node of that robot is node 14, it will check NNCSD first to determine the minimum distance weight from node 14 to node 17. After determining the minimum distance, it will look into the RMPM for searching the route direction of that specific distance weight from node 14 to node 17. Here in RMPM, the mobile robot can find a number of paths in between node 14 to node 17. After that it will match the specific weight with all that available paths and after matching it will pass through the minimum weight path which can be determined by Kruskal's Algorithm through Modified Ant Colony Algorithm. So, the basic workflow of the complete operation is as follows,

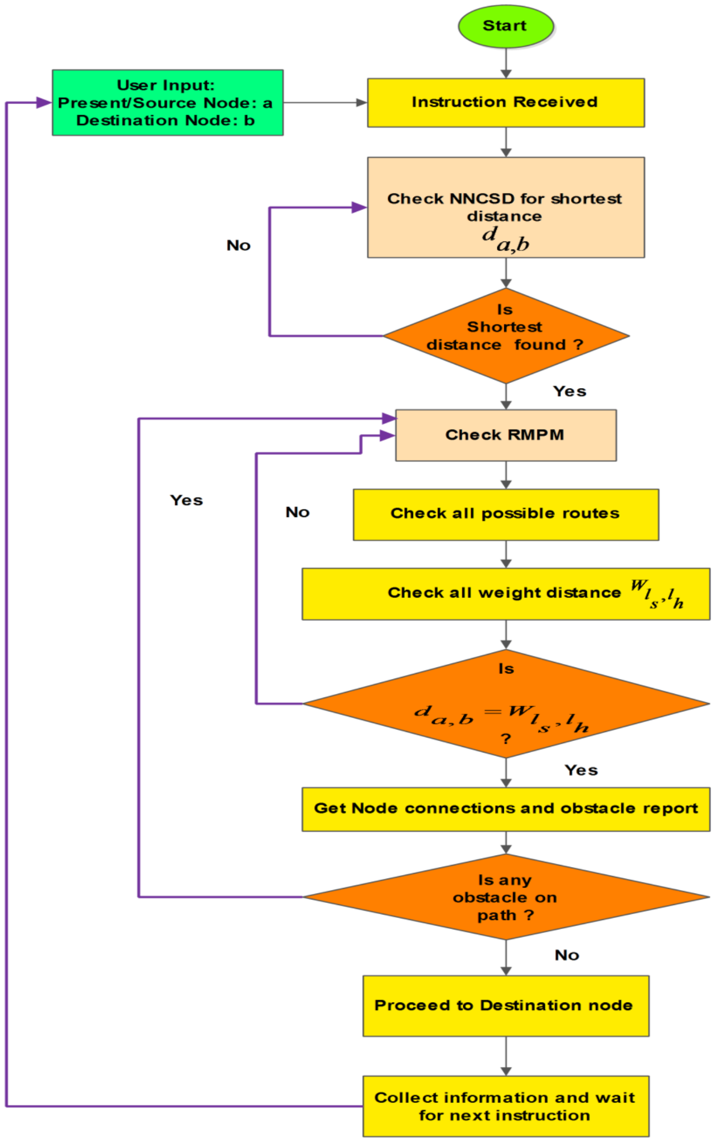



**Figure 7. Complete Operation of Routing-Routing Algorithm (RA)**

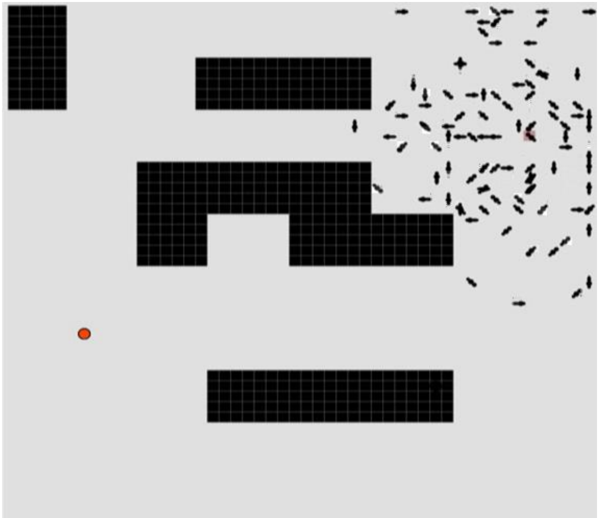
The mobile robot will follow the RA for its successful operation. This can be considered as a smart search for the mobile agent who are roaming in the colony. The Routing algorithm (RA) itself is collaborated with MACA, RMPM and NNCSD and so it can fetch the data and instruction from those easily and thereby increasing the efficiency of the routing and searching operation. The simulations for the routing of the mobile agents are given in the next section.

## 6. Simulation Result

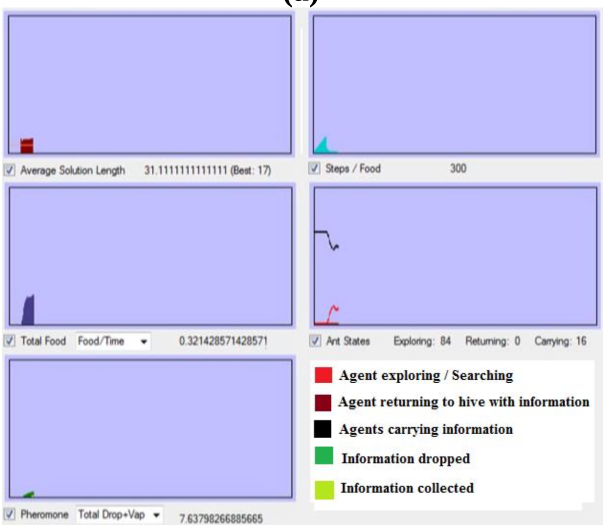
The simulation is done using AntSim Electronics Ant Simulator with the same colony structure that is defined in RMPM. In the simulation we can see that the agent is placed in Node 14 which is the home node at present and is instructed to move to Node 17 to collect the information. The agent will then match the distance weight with each possible path direction and can find that Route#2 and Route#3 are the same and with the same weight that is fetched from NNCSD. So the agent randomly chooses any one of the selected route and move towards Node17.

The Simulation shows the exact and it will move through the shortest route as decided from NNCSD and RMPM. The simulation results are shown below:

***i) Iteration 10:***



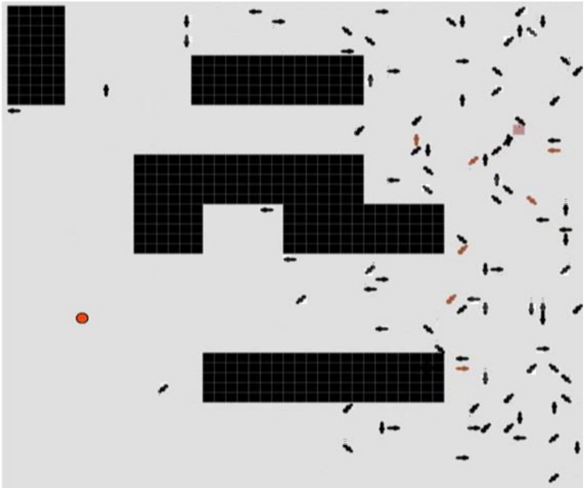
**(a)**



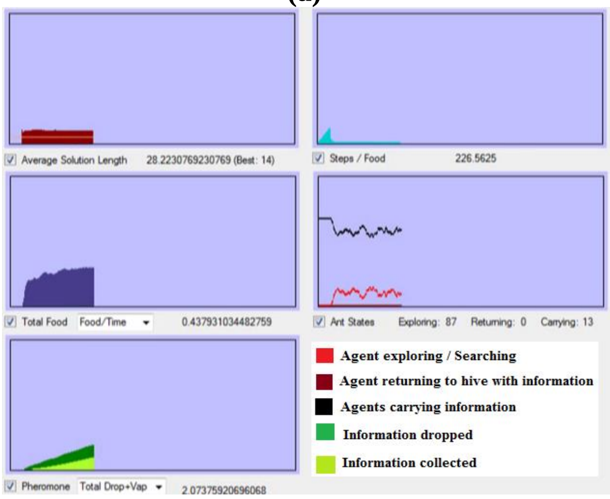

**(b)**

**Figure 8. Agents Start Moving from Node 14 (a) Agents are Searching for Node 17, (b) Statistics of Movement**

***ii) Iteration 50:***



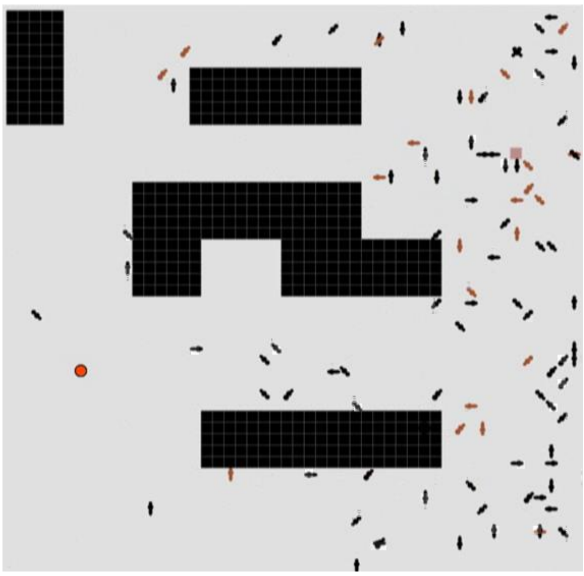
(a)




(b)

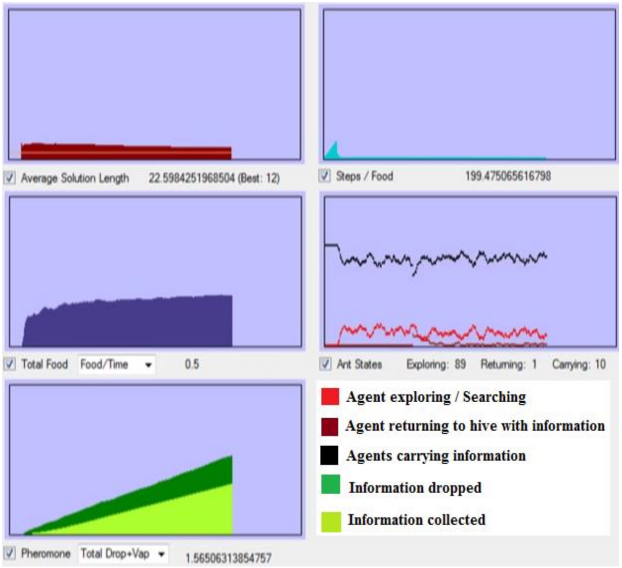
**Figure 9. Agents are Roaming around Colony (a) Agents are Searching for Node 17 & Collecting Information (b) Statistics of Movement**

***iii) Iteration 100:***



(a)

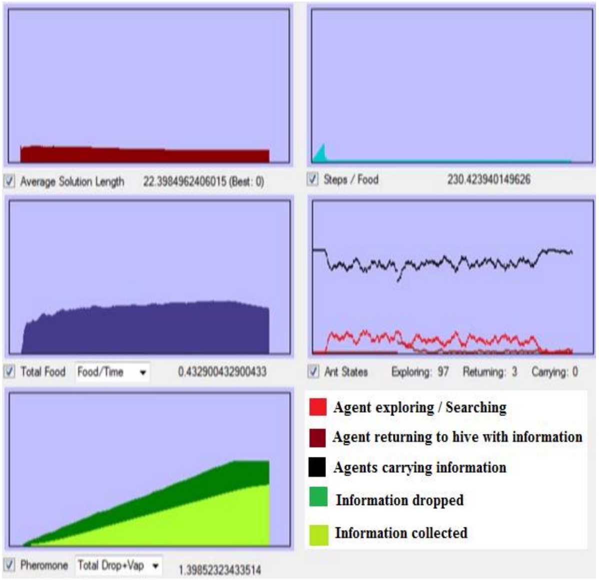



**(b)**

**Figure 10. Agents are Exploring around Colony (a) More Agents are Exploring by Collecting Information (b) Statistics of Movement**

***iv) Iteration 150:***

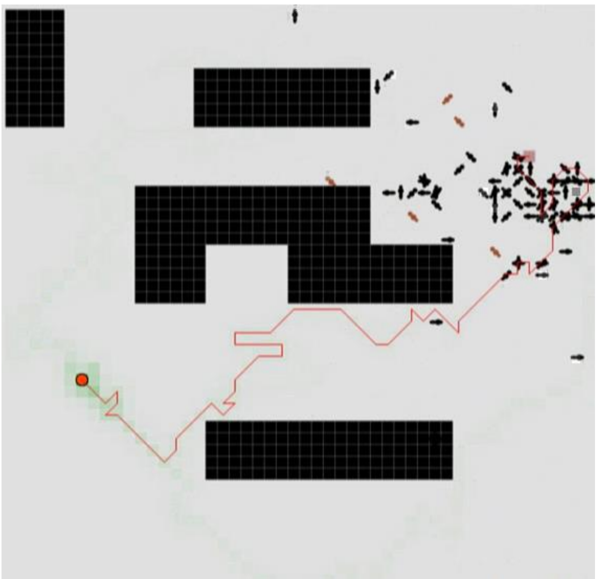
**(a)**



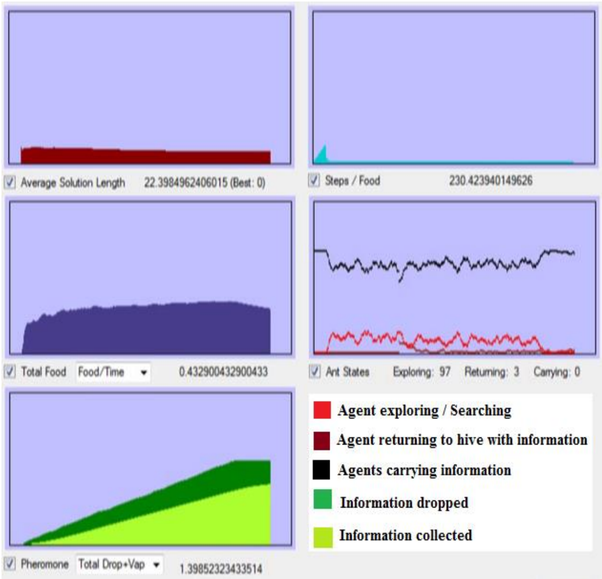

(b)

**Figure 11. Agents Find the Shortest Route for Node 14 to Node 17 (a) Agents Finally Find the Shortest Route Same as RMPM (b) Statistics of Movement**

***v)Iteration 200:***



(a)

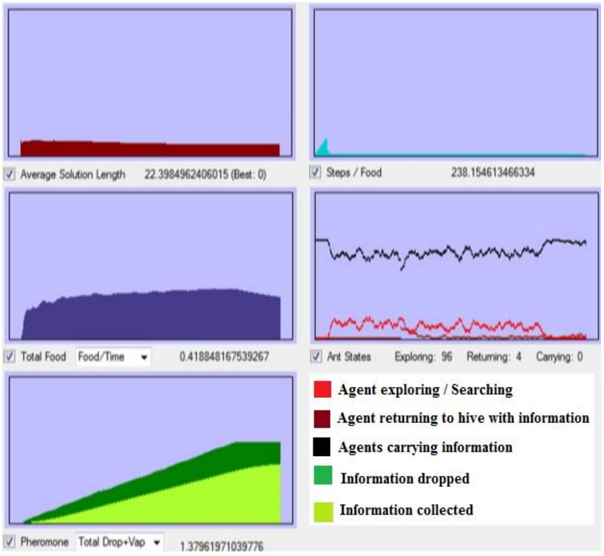



(b)

**Figure 12. Agents are Engaged for Information Collection (a) Agents Follow the Shortest Route to Collect Information from Node 17 (b) Statistics of Movement**

### vi) Iteration 300:

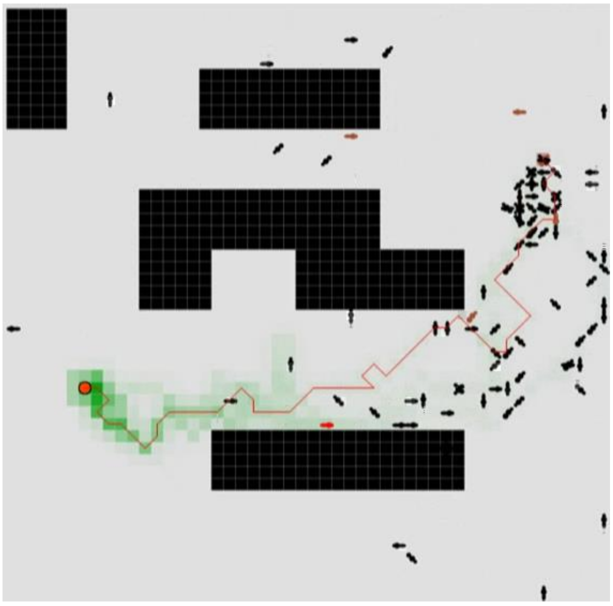
(a)




(b)

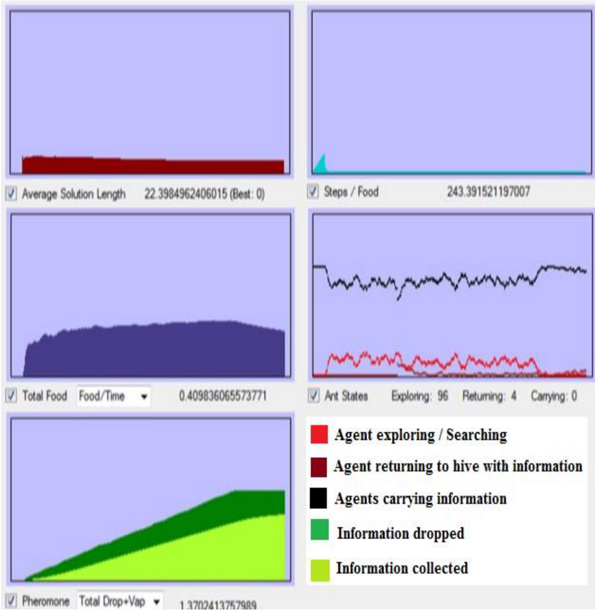
**Figure 13. More Agents are Engaged for Information Collection (a) Agents are Still Collecting Information form Node 17 (b) Statistics of Movement**

***vii) Iteration 400:***



(a)

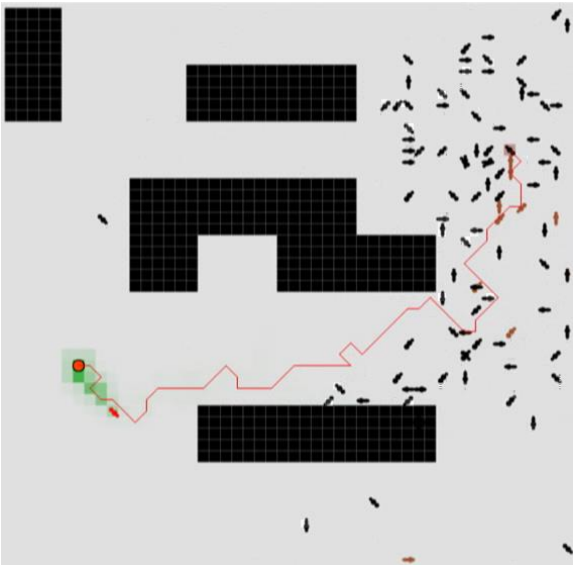
 



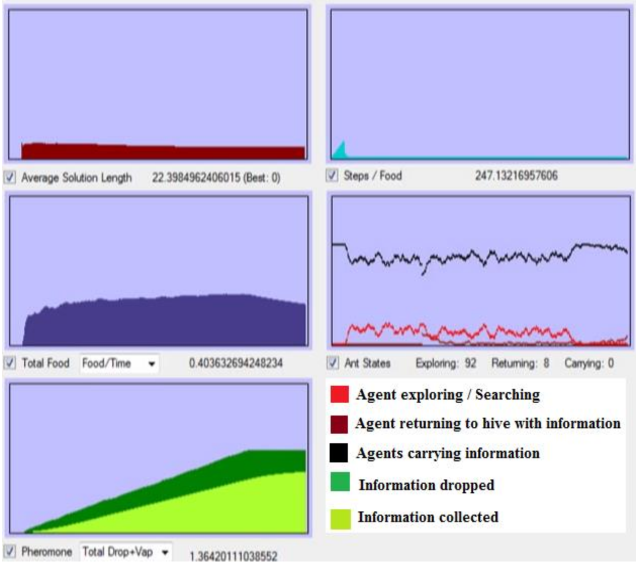

**(b)**

**Figure 14. Agents are Collecting Information by More Involvement (a) Agents are Following the Chemical Trail to Collect More Information (b) Statistics of Movement**

***viii) Iteration 500:***



**(a)**




**(b)**

**Figure 15. Agents are Spreading in a Large Amount for Information Collection (a) Agents are Collecting the Information and Returning Back to Home Node (b) Statistics of Movement**

## 7. Conclusion

The simulation result shows that while moving from $Node$ 14 to $Node$ 17, the agents will always select the shortest route. In the simulation, it can be observed that the agents are selecting the Route #2 as described in Table-1. The NNCSD is designed to determine the various distances in between any two nodes in the colony. This database is entirely independent of JCD and UNCD. So, to design NNCSD, JCD and UNCD are not actually required. So the working algorithms of the agents are RCA which is dependent on MACA. The working database for the agents is the NNCSD. Now on the basis of RCA and NNCSD, the routing algorithm (RA) is designed which is actually use to drive the agent routing process in the colony. In this paper the travelling in between $Node$ 14 to $Node$ 17 is shown and in my previous paper the travelling between $Node$ 1 to $Node$ 8 was observed and in both cases the agents follow the shortest path and so these observations are very helpful for the solution of the TSPs and VRPs.

## Acknowledgements

In this paper, the single database NNCSD is introduced which reduces the computation and simulation time that was taken by JCD and UNCD. More database means more computation time. So with the introduction of NNCSD, the efficiency of RCS is much more enhanced. To route the agents around the colony, RA algorithm is essentially required which is also useful to solve the Travelling Salesman Problem and the Vehicle Routing Problem. The selection of the proper shortest path is done by matching the single database and route direction can be found by Echolocation or bio-sonar process which can be described by Bat Algorithm, will be discussed in the next research work. The idea of this paper is taken from the list of references.

## References


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

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

[3] 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.

[4] 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.

[5] 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.

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

[7] M. Dorigo, Gambardella L.M, "Ant Colony System: a cooperative learning approach to the travelling salesman problem",IEEE Transaction on Evolutionary Computing, vol. 1,Issue 1, **(1997)** April, pp 53-66.

[8] S. Chakraborty, "Ant Colony System: A New Concept to Robot Path Planning", International Journal of Hybrid Information Technology, Copyright @ 2013 SERSC, vol. 6, no. 6, **(2013)**, pp. 11-30.

[9] S. Chakraborty, "Ant Colony System: An improved approach to Robot Path Planning", International Journal of Hybrid Information Technology, Copyright @ 2013 SERSC, vol. 7, no. 2, **(2013)**, pp. 249-268.

[10] A. A. A. Zakzouk, H. M. Zaher and R. A. Z. El-Deen, "An ant colony optimization approach for solving shortest path problem with fuzzy constraints", 7$^{th}$ International Conference on Informatics and Systems(INFOS), **(2010)**, pp. 1-8.

[11] A. Yang, L. Gao and Y. Luo, "An improved Ant Colony System Algorithm for Optimal Path Planning Problem of Mobile Robots", Second International Conference on Computer Modeling and Simulation, vol. 1, **(2010)**, pp. 526-530.

[12] B. A. Garro, H. Sossa and R. A. Vezquez, "Evolving ant colony System for optimizing path planning in mobile robots", Robotics and Automotive Mechanics Conferences, CERMA, **(2007)** September, pp. 444-449.

[13] S. H. Ok, W. J. Seo, S. Kang and B. Moon, "An Any Colony Optimization Approach for the Preference-Based Shortest Path Search", Communication and Networking Communications in Computer and Information Science, vol. 56, **(2009)**, pp. 539-546.

[14] H. Salehinejad and S. Talebi, "Dynamic Fuzzy Logic-Ant Colony System-Based Routing Selectin System", Applied Computational Intelligence and Soft Computing, **(2010)**.

[15] M. Farhanchi, R. Hassanzadeh, I. Mahadavi and N. Mahdavi-Amiri, "A modified ant colony system for finding the expected shortest path in networks with variable arc lengths and probabilistic nodes", Applied Soft Computing, Elsevier, August 2014,Vol. 21, pp. 491-500.

[16] C.-C. Hsu, R.-Y. Hou and W.-Y. Wang, "Path Planning for Mobile Robots Based on Improved Ant Colony Optimization", IEEE Conference on Systems, Man and Cybernatics, **(2013)** October, pp. 2777-2782.

[17] X. Fan, X. Luo, S. Yi, S. Yang and H. Zhang, "Optimal path planning for mobile robots based on intensified ant colony optimizing algorithm", IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, vol. 1, **(2003)** October, pp. 131-136.

[18] 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.

[19] 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.

[20] 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.

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

[22] 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)**.

[23] S. -C. Chu, J. F. Roddick and J. -S. Pan, "Ant colony system with communication strategies", Information Sciences, @ 2003 Elsevier, Inc., vol. 167, **(20064)**, pp. 63–76.

[24] 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.

[25] 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.
[26] Z. Liu, M. Z. Kwiatkowska and C. Constantinou, "A Swarm Intelligence Routing Algorithm For Manets". 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".
[27] D. G. Bucatanschi, "The Ant Colony System for the Freeze-Tag Problem", MCURCSM 2004, Granville, OH, **(2004)**, pp. 61-69.
[28] T. Neng, C. Guo and T. Ning, "Solving VRP Using Ant Colony Optimization Algorithm", 5th International Conference on Information and Computing Science (ICIC), IEEE, **(2012)** July, pp. 15-18.
[29] X. S. Chen, M. H. Lim and Y. S. Ong, "An Ant Colony Algorithm for path planning in sparse graphs", International Conference on Intelligence and Advanced Systems, IEEE, **(2007)** November, pp 31-36.
[30] F. Mourad, H. Chehade, H. Snoussi, F. Yalaoui, L. Amodeo and C. Richard, "Controlled Mobility Sensor Network for Target Tracking Using Ant Colony Optimization", IEEE Transactions on Mobile Computing, vol. 11, Issue 8, **(2012),** pp. 1261-1273.
[31] Z. F. Jun and W. Gao, "Meeting Ant Colony Optimization", IEEE Internation Symposium on Knowledge Acquisition and Modeling Workshop, **(2008)**, pp. 972-975.
[32] S.-Q. Wang and Z.-Yu Xu, "Ant Colony Algorithm Approach for Solving Traveling Salesman with Multi-agent", WASE International Conference on Information Engineering, IEEE, vol. 1, **(2009),** pp. 381-384.
[33] K. Wei, H. Tuo and Z. Jing, "Improving Binary Ant Colony Optimization By Adaptive Pheromone And Commutative Solution Update", IEEE 5th Internation Conference on Bio-Inspired Computing: Theories and Applications, **(2010)**, pp. 565-569.
[34] Y. Zhang, Z.-L. Pei, J.-H. Yang and Y. C. Liang, "An Improved Ant Colony Optimization Algorithm Based on Route Optimization and Its Applications in Travelling Salesman Problem", IEEE 7th International Conference on Bioinformatics and Bioengineering, **(2007)**, pp. 693-698.
[35] Y. Bi, L. Ding and J. Lu, "On the faster Ant Colony Optimization Algorithm", 5th International Conference on Natural Computation, IEEE, vol. 3, **(2009),** pp. 45-49.
[36] J. Anitha and M. Karpagam, "Ant Colony Optimization using Pheromone updation strategy to solve job shop scheduling", 7th International Conference on Intelligent Systems and Control, IEEE, **(2013)**, pp. 367-372.

# Author

**Subhadeep Chakraborty**, born in 1986, is the Software Analyst and Programmer in SonsIndia Software Private Limited. 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 served Calcutta Institute of Technology for 4 years 7 months. His primary research interest includes Digital Signal Processing, Shot Transition, Robotics, Artificial intelligence, Microprocessor.