Major Project Report
On

ROUTE OPTIMIZATION IN MOBILE AD-HOC NETWORKS USING ANT COLONY OPTIMIZATION TECHNIQUE

Submitted in partial fulfillment of the requirement

For the award of the degree of

MASTER OF TECHNOLOGY
(INFORMATION SYSTEM)

Submitted by

SHEEL KUMAR

(14/IS/10)

Under the Guidance of

Ms. ANAMIKA CHAUHAN

(Asst. Professor)

Dept. of Information Technology

Department of Information Technology

DELHI TECHNOLOGICAL UNIVERSITY

Bawana Road, Delhi-110042

2010-2012

CERTIFICATE

This is to certify that the Major Report entitled as "Route Optimization in Mobile Ad-hoc Networks using Ant Colony Optimization Technique" is being submitted by **Sheel Kumar**, 14/IS/10, in partial fulfillment for the award of degree of Master of Technology in Information System from Delhi Technological University, Delhi, is the original work carried out by him under the guidance of **Ms. Anamika Chauhan (**Asst. Prof. IT Department), DTU. The matter contained in this thesis report has not been submitted elsewhere for reward of any other degree.

(Project Guide)

Ms. Anamika Chauhan

Assistant Professor
Dept. of Information Technology
Delhi Technological University,
Delhi

ACKNOWLEDGEMENT

First of all, I would like to present my sincere thanks to my project guide **Ms. Anamika Chauhan**, Asst. Prof. of Information Technology, DTU, Delhi for her insightful guidance and constant encouragement during the preparation of this project report.

Secondly, I would like to thank **Dr. O.P.Verma**, Head of Dept. of Information Technology, DTU, Delhi, for giving me the freedom to pursue my own research interest and study diverse topics throughout my M.Tech study. I would thank all the members of DTU and my friends who have helped me in the completion of my thesis report.

SHEEL KUMAR

M.Tech

(Information System)

14/IS/10

ABSTRACT

Routing is the core of any network, strongly affecting the overall network performance. The general routing problem is the problem of defining path flows to forward incoming data traffic such that the overall network performance is optimized. At each node data is forwarded according to a decision policy parameterized by a local data structure called routing table. The route finding algorithm starts as soon as a data packet needs to be sent from a source to a given destination. Aim of routing is to find the route in the shortest possible time and to guarantee finding a path to the destination when such a path exists. Routing in Mobile Ad-hoc Networks faces difficulty of frequent topology change, as Mobile Ad-hoc networks are often characterized by a dynamic topology due to the fact that nodes change their physical location by moving around. Swarm intelligence is overviewed and it is found that the Ant Colony Optimization Technique exhibits features which are highly desirable in Ad-hoc Networks. Ant Colony Optimization Technique suites the routing requirements of Mobile Ad-hoc Networks because of its Dynamic topology, Local information and Support for multi-path. Therefore here we have used the ant colony optimization technique for finding an optimized route in the Mobile Ad-hoc Networks. The goal of this thesis is to design a new adaptive routing technique for finding an optimized route in Mobile Ad-hoc Networks. The work provides an overview of routing protocol in Ad-hoc networks. Various mechanisms that are commonly encountered in Ad-hoc routing are experimentally evaluated under situations as close to real life as possible. Where possible, enhancements to the mechanism are suggested. Finally, an optimized route, suitable for Ad-hoc Networks is defined and compared with one of the exiting algorithm.

Keywords: Routing, Routing Table, MANET, ACO, Dynamic Topology.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1 INTRODUCTION

1.1 SWARM INTELLIGENCE

Swarm intelligence (SI) is a type of artificial intelligence based on the collective behavior of decentralized, self-organized systems. The expression was introduced by Gerardo Beni and Jing Wang in 1989 in the context of cellular robotic systems [1]. SI systems are typically made up of a population of simple agents interacting locally with one another and with their environment. The agents follow very simple rules and although there is no centralized control structure dictating how individual agents should behave local and to a certain degree random interactions between such agents.

1.2 ANT COLONY OPTIMISATION

The ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. This algorithm is a member of ant colony algorithms family in swarm intelligence methods. The first algorithm was aiming to search for an optimal path in a graph based on the behavior of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of Numerical problems and as a result several problems have emerged drawing on various aspects of the behavior of ants.

The Nature gives us a solution to find the shortest path. The ants in their necessity to find food and brings it back to the nest manage not only to explore a vast area but also to indicate to their peers the location of the food while bringing it back to the nest. Thus they know where their nest is and also their destination without having a global view of the ground. Most of the time, they find the shortest path and adapt to ground changes hence proving their great efficiency toward this difficult task.

A. THE ANTS

Ant as a single individual has a very limited effectiveness. But as a part of a well-organized colony it becomes one powerful agent working for the development of the colony. The ant lives for the colony and exists only as a part of it. Each ant is able to communicate, learn, cooperate, and all together they are capable of developing themselves and colonies a large area. They manage such great successes by increasing the number of individuals and being exceptionally well organized. The self organizing principles they are using allow a highly coordinated behavior of the colony. Pierre Paul Grasse, a French entomologist was one of the first researchers who investigate the social behavior of insects. He discovered that these insects are capable to react to what he called significant stimuli signals that activate a genetically encoded reaction. He observed that the effects of these reactions can act as new significant stimuli for both the insect that produced them and for the other insects in the colony. Grasse used the term stigmergy to describe this particular type of indirect communication in which ‘the workers are stimulated by the performance they have achieved’.

Stigmergy is defined as a method of indirect communication in a self-organizing emergent system where its individual parts communicate with one another by modifying their local environment. Ants communicate to one another by laying down pheromones along their trails so where ants go within and around their ant colony is a stigmergic system.

In many ant species ants walking from or to a food source deposit on the ground a substance called pheromone. Other ants are able to smell this pheromone and its presence influences the choice of their path that is, they tend to follow strong pheromone concentrations. The pheromone deposited on the ground forms a pheromone trail which allows the ants to find good sources of food that have been previously identified by other ants. Using random walks and pheromones within a ground containing one nest and one food source, the ants will leave the nest, find the food and come back to the nest. After some time the way being used by the ants will converge to the shortest path.

B. THE PHEROMONES

Pheromones represent in some ways the common memory. The fact that it is external and not a part of the ants/agents confer to it as an easy access for everyone. The memory is saved in without regarding the configuration of the ground, the number of ants etc. It is totally independent and still remains extremely simple.

Pheromones just proceed to one task; nature will take care of it in the real life, although it is a simple process in algorithms. In course of time a global reduction of the pheromones by a certain factor is applied, simulating the evaporation process. Thus the non-succeeding path will see their concentration of pheromones reduced although good solutions will stay full of pheromones as the ants keep using it.

1.3 THE DOUBLE BRIDGE EXPERIMENT

The ants begin by walking randomly. They cannot see the ground and have a very limited view of what is around them. Therefore if the ground has not been explored yet they will just wander and take random decision at each crossroads.

After a while the places around the nest will be all explored. The ants will get to know that by the marking done by the previous ants. Indeed they will leave behind them the famous pheromones and inform the other ants that the way is already explored. The concentration of pheromones depends on the number of ants who took the way, the more ants taking the way the more pheromones.

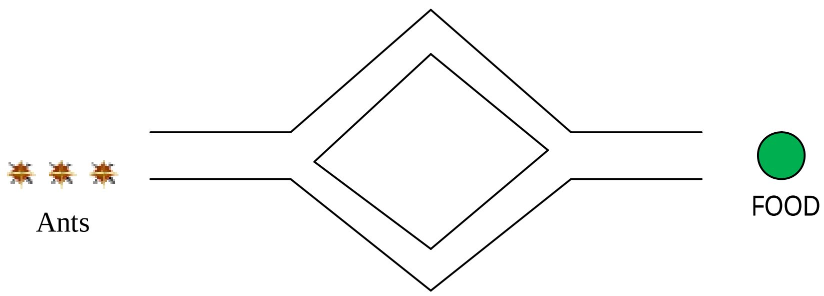

Fig. 1.1: Double Bridge Experiment with equal length paths

The configuration is as shown in figure 1.1; the nest of a colony of ants is connected to the food via two bridges of the same length. In such a setting ants start to explore the surroundings of the nest and eventually reach the food source. Along their path between food source and nest ants deposit pheromones. Initially each ant randomly chooses one of the two bridges. However due to random fluctuations after some time one of the two bridges presents a higher concentration of pheromones than the other and therefore attracts more ants. This brings a further amount of pheromone on that bridge making it more attractive with the result that after some time the whole colony converges toward the use of the same bridge.

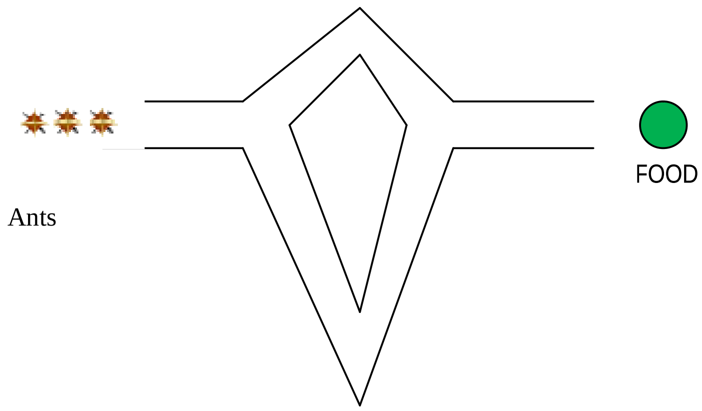

Fig. 1.2: Double bridge experiment with one path twice the length of other path.

The second experimentation fig. 1.2 also gives two paths to the food source but one of them is twice longer than the other one. Here again the ants will start to move randomly and explore the ground. Probabilistically 50% of the ants will take the short way while the 50% others will take the long way as they have no clue about the ground configuration. The ants taking the shorter path will reach the food source before the others and leave behind them the trail of pheromones. After reaching the food they will turn back and try to find the nest. At the cross one of the paths will contain pheromones although the other one will be not explored. Hence the ant which carries the food will take the path already explored as it means it is the way to the nest.

As the ant is choosing the shortest way and will continue to deposit pheromones, the path will therefore become more attractive for others. The ants who took the long way will have more probability to come back using the shortest way and after some time they will all converge toward using it. Consequently the ants will find the shortest path by themselves (fig. 1.3) without having a global view of the ground. By taking decision at each cross according to the pheromones amount they will manage to explore find the food and bring it back to the nest in an optimized way.

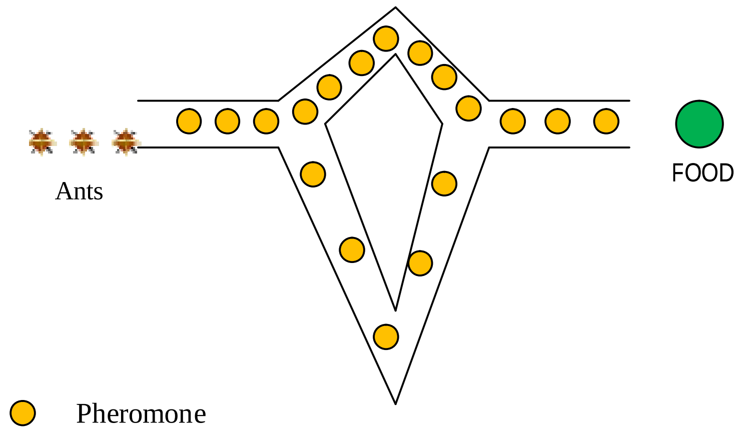

Fig. 1.3: Pheromone deposition

1.4 MAIN ACO ALGORITHMS

Several special cases of the ACO meta-heuristic have been proposed. Here we briefly overview in the historical order in which they were introduced, the three most successful ones: ant system (Dorigo 1992), ant colony system (ACS) (Dorigo & Gambardella 1997) and MAX-MIN ant system (MMAS) (Stutzle & Hoos 2000). In order to illustrate the differences between them clearly we use the example of the traveling salesman problem.

1.4.1 Ant System

Ant system (AS) was the first ACO algorithm to be proposed (2, 3). Its main characteristic is that at each iteration the pheromone values are updated by all the m ants that have built a solution in the iteration itself. The pheromone τ_{ij} associated to the edge joining cities i and j is updates as follows:

$$\tau_{ij}=(1-\rho)*\tau_{ij}+\sum_{k=1}^{m}\Delta\tau_{ij}^{k}$$

Where ρ Є (0, 1] is the evaporation rate, m is the number of ants and $\Delta\tau_{ij}^{k}$ is the quantity of pheromone deposited over the edge (i, j) by the k^{th} ant.

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{L_K} & \text{if kth ant used edge (i, j) in its tour,} \\ 0 & \text{otherwise} \end{cases}$$

Where L_k is the tour length of the k^{th} ant.

1.4.2 MAX−MIN Ant System (MMAS)

This algorithm [4] is an improvement over the original Ant System. Its characterizing elements are that only the best ant updates the pheromone trails and that the value of the pheromone is bound. The pheromone update is implemented as follows:

$$\tau_{ij}=\left[(1-\rho)*\tau_{ij}+\sum_{k=1}^{m}\Delta\tau_{ij}^{best}\right]_{\tau_{min}}^{\tau_{max}}$$

τ_{max} and τ_{min} are respectively the upper and lower bounds imposed on the pheromone.

$$\Delta\tau_{ij}^{best} = \begin{cases} \frac{1}{L_{best}} & \text{if edge (i, j) in the best tour} \\ 0 & \text{otherwise} \end{cases}$$

Where L_{best} is the length of the tour of the best ant. This is may be (subject to the algorithm designer decision) either the best tour found in the current iteration (iteration-best) or the best solution found since the start of the algorithm (best-so-far).

1.4.3 Ant Colony System (ACS)

The most interesting contribution of ACS [5, 6, 7] is the introduction of a local pheromone update in addition to the pheromone update performed at the end of the construction process (called off line pheromone update).

The local pheromone update is performed by all the ants after each construction step. Each ant applies it only to the last edge traversed:

$$\tau_{ij} = (1-\varphi)\tau_{ij} + \varphi\,\tau_o$$

Where $\varphi \in (0, 1]$ is the pheromone decay coefficient, τ_o is the initial value of the pheromone.

The main goal of the local update is to diversify the search performed by subsequent ants during iteration by decreasing the pheromone concentration on the traversed edges, ants encourage subsequent ants to choose other edges and hence to produce different solutions. This makes it less likely that several ants produce identical solutions during one iteration.

The offline pheromone update, similarly to MMAS is applied at the end of each iteration by only one ant, which can be either the iteration-best or the best-so-far. However the update formula is slightly different.

$$\tau_{ij} = \begin{cases} \tau_{ij} = (1-\rho)\tau_{ij} + \rho\,\Delta\tau_{ij} \\ \tau_{ij} \end{cases} \quad \text{if edge } (i, j) \text{ in the best tour otherwise}$$

Here ρ is decay constant and $\Delta\tau_{ij} = 1/L_{best}$.

1.5 ROUTING: Definition and characteristics

Routing is at the core of any network control system, strongly affecting the overall network performance. Routing can be characterized in the following general way. Let the network be represented in terms of a directed weighted graph G = (V, E), where each node in the set V represents a processing and forwarding unit and each edge in E is a transmission system with some capacity/bandwidth and propagation characteristics. Data traffic originates from one node (end-point) and can be directed to another node (unicast traffic), to a set of other nodes (multicast traffic) and/or to all the other nodes (broadcast traffic). The node from where the traffic flow originates is also called source or starting end-point while the nodes to which traffic is directed are the final end-points or destinations. The nodes in-between that forward traffic from sources to destinations are called intermediate or relay nodes. A flow is a vector in R |E| that for a traffic pair (s, D), $s \in V$, $D \subseteq V$ assigns a way of forwarding the data traffic from s to the nodes in D across the network.

The general routing problem is the problem of defining path flows to forward incoming data traffic such that the overall network performance is maximized. At each node data is forwarded according to a decision policy parameterized by a local data structure called routing table.

The route finding algorithm starts as soon as a data packet needs to be sent from a source to a given destination. Aim is to find the route in the shortest possible time and to guarantee finding a path to the destination when such a path exists.

In spite of the aspects that characterize each specific routing problem at hand any routing strategy virtually consists of the same set of activities carried out at each node as shown in Algorithm:

At each network node:

1. Acquisition and organization of up-to-date information concerning the local state that is information on the local traffic flows and on the status of the locally available resources.

2. Build up a view of the global network state possibly by some form of exchanging of the local state information.

3. Use of the global view to set up the values of the local routing table and consequently to define the local routing policy with the perspective of optimizing some measure of network performance.

4. Forward of the user traffic according to the defined routing policy.

5. Asynchronously and concurrently with the other nodes repeat the previous activities over time.

The goal of every routing algorithm is to direct traffic from sources to destinations optimizing at the same time several measure of network performance as throughput (correctly delivered bits per time unit), packet delays and resources utilization.

1.5.1 Classification of routing algorithms:

A. Control architecture: Centralized vs. Distributed

In centralized algorithms a main controller is responsible for the updating of all the node routing tables and/or for every routing decision. Centralized algorithms can be used only in particular cases and for small networks. In general the controller has to gather information about the global network status and has to transmit all the decisions/updates. The relatively long time delays necessarily involved with such activities as well as the lack of fault-tolerance make centralized approaches unfeasible in practice. From now on only non-centralized that is distributed routing systems are considered. In distributed routing systems, every node autonomously decides about local data forwarding. At each node a local routing table is maintained in order to implement the local routing policy. The distributed paradigm is currently used in the majority of network systems.

B. Routing tables: static vs. dynamic

Routing tables can be statically assigned or dynamically built and updated. It is evident that the performance of the two approaches can be radically different and the appropriateness of one approach over the other tightly depends on the characteristics of the network scenario under consideration.

In both cases routing tables are built in order to possibly optimize some network-wide criteria which are made depending in turn on costs associated to network elements. That is to each link or whatever network resource of interest (e.g. available processing power of a routing node), a value (integer, real, nominal), here called cost is assigned according to some metric in order to have a measure of either utilization level or physical characteristics (e.g. bandwidth, propagation delay). Therefore the process of finding routing paths optimized with respect to the chosen criteria can be actually intended as the minimization process with respect to the defined costs (e.g. the overall cost criterion can be expressed in terms of a sum of the link costs or of the path/link flows).

Static routing

In static (or oblivious) routing systems the path to forward traffic between pairs of nodes is determined without regard to the current network state. The paths are usually chosen as the result of the offline optimization of some selected cost criterion. Once defined the paths to be used for each source-destination pair, data are always forwarded along these paths. Routing tables can be also assigned on the basis of some a priori knowledge about the expected input traffic.

Dynamic routing

Dynamic (or adaptive) routing goes beyond static routing by admitting the possibility of building/changing the routing tables online according to the current traffic events. It is useful to distinguish between the ability of adapting to the changing traffic conditions and to topological modifications (e.g. link/node failures, link/node addition/removal).

1.5.2 Metrics for performance evaluation

The performance of a network and accordingly of the control algorithms active on it is measured according to metrics which depend on the types of services expected to be delivered by the network. Performance is usually measured over a suitable time interval and can be expressed either in term of instantaneous, cumulative or average values.

For this class of networks standard metrics for performance evaluation are:

• ***Throughput***: The number of correctly delivered data bits/sec. Throughput is a global index of performance associated to the quantity of delivered service. It is usually expressed as the sum of correctly delivered bits and/or packets over a specified time interval.

• ***End-to-end delay for data packets***: The time necessary to a data packet to reach its destination node. The values of packet delays can be spread over a wide range. This is an intrinsic characteristic of data networks. Packet delays can range from very low values, for data flows open between adjacent nodes connected by fast links to much higher values, in the case of flows involving nodes very far apart and reachable only through by low band-width links. Because of this in the general case the empirical distribution of packet delays cannot be expressed in terms of a unimodal parametric distribution. Therefore mean and variance of packet delays may not able to capture the most important statistical aspects of the observed data.

• ***Network resources utilization*** considering both data and routing packets. Network resources commonly considered are the link capacities and the memory and processing time of the nodes. Network resources utilization is usually expressed as the used fraction of the over-all available resources.

1.6 Mobile Ad-hoc Network (MANET)

1.6.1 General Concept

An ad-hoc wireless network is a collection of mobile/semi-mobile nodes with no pre-established infrastructure, forming a temporary network. Each of the nodes has a wireless interface and communicates with each other over either radio or infrared. Laptop computers and personal digital assistants that communicate directly with each other are some examples of nodes in an ad-hoc network. Nodes in the ad hoc network are often mobile but can also consist of stationary nodes such as access points to the Internet. Semi mobile nodes can be used to deploy relay points in areas where relay points might be needed temporarily.

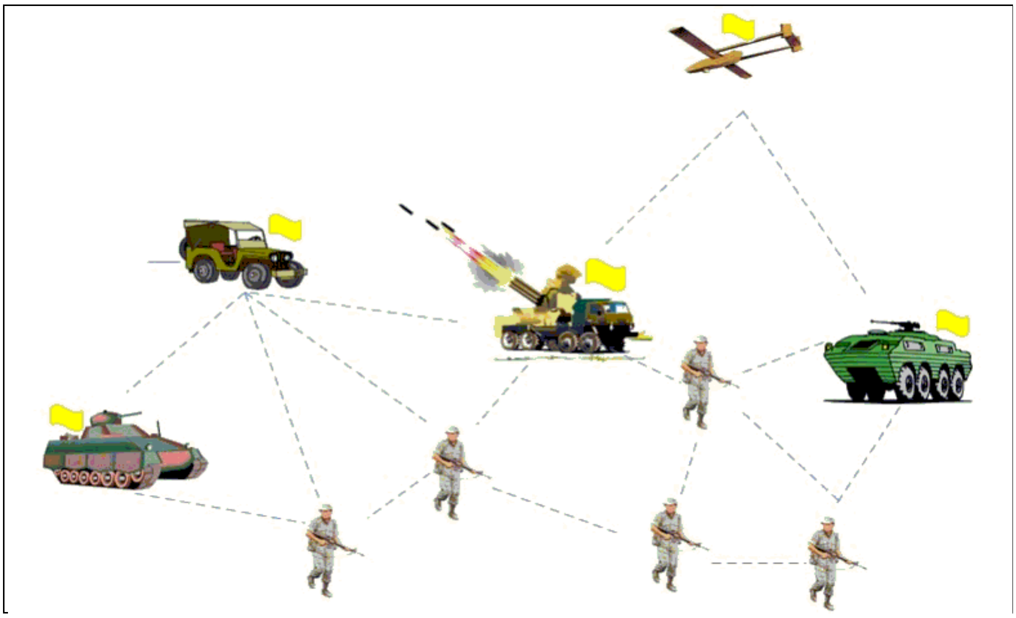

Fig 1.4: Depicts the application of MANET in battlefields.

The hosts in the above figure communicate with each other using radio frequency waves in the absence of a fixed or central controller over the whole communication system.

An ad-hoc network uses no centralized administration. This is to be sure that the network will not collapse just because one of the mobile nodes moves out of transmitter range of the others. Nodes should be able to enter/leave the network as they wish. Because of the limited transmitter range of the nodes multiple hops may be needed to reach other nodes. Every node wishing to participate in an ad-hoc network must be willing to forward packets for other nodes. Thus every node acts both as a host and as a router. A node can be viewed as an abstract entity consisting of a router and a set of affiliated mobile hosts. A router is an entity which among other things runs a routing protocol. A mobile host is simply an IP-addressable host/entity in the traditional sense.

Ad-hoc networks are also capable of handling topology changes and malfunctions in nodes. It is fixed through network reconfiguration. For instance if a node leaves the network and causes link breakages, affected nodes can easily request new routes and the problem will be solved. This will slightly increase the delay but the network will still be operational. Wireless ad-hoc networks take advantage of the nature of the wireless communication medium. In other words in a wired network the physical cabling is done a priori restricting the connection topology of the nodes.

This restriction is not present in the wireless domain and provided that two nodes are within transmitter range of each other an instantaneous link between them may form.

1.6.2 Characteristics of MANET

Ad-hoc networks are often characterized by a dynamic topology due to the fact that nodes change their physical location by moving around. This favors routing protocols that dynamically discover routes over conventional routing algorithms like distant vector and link state. Another characteristic is that a host-node has very limited CPU capacity, storage capacity, battery power and bandwidth also referred to as a "thin client". This means that the power usage must be limited thus leading to a limited transmitter range.

The access media, the radio environment also has special characteristics that must be considered when designing protocols for ad-hoc networks. One example of this may be unidirectional links. These links arise when for example two nodes have different strength on their transmitters, allowing only one of the hosts to hear the other but can also arise from disturbances from the surroundings. Multi-hop in a radio environment may result in an overall transmit capacity gain and power gain, due to the squared relation between coverage and required output power. By using multi-hop, nodes can transmit the packets with a much lower output power.

1.6.3 Usage of MANET

There is no clear picture of what these kinds of networks will be used for. The suggestions vary from document sharing at conferences to infrastructure enhancements and military applications. In areas where no infrastructure such as the Internet is available an ad-hoc network could be used by a group of wireless mobile hosts. This can be the case in areas where a network infrastructure may be undesirable due to reasons such as cost or convenience. Examples of such situations include disaster recovery personnel or military troops in cases where the normal infrastructure is either unavailable or destroyed.

Other examples include business associates wishing to share files in an airport terminal or a class of students needing to interact during a lecture. If each mobile host wishing to communicate is equipped with a wireless local area network interface, the group of mobile hosts may form an ad-hoc network. Access to the Internet and access to resources in networks such as printers are features that probably also will be supported.

1.6.4 Benefits and Applications of MANET

MANET is useful in many situations where impromptu communication facilities are required. It is most useful for mobile host communication in a particular terrain where fixed infrastructure is unavailable. In disaster relief missions, construction sites, military maneuvers, instant conferencing for meetings or sensor networks a temporary communication network such as MANET would be most appropriate.

Recently it has even been commercialized for public forum communication. Electronic classrooms, instant conferencing for meetings and conventions, concerts and festivals are just a few commercial examples. There is also an on-going research that studies the feasibility of MANET usage in the area of Intelligent Transportation System (Varaiya, 1993). With such a broad scope of applications ad hoc networks are expected to operate over a wide range of coverage areas, node densities and node velocities. The benefits of MANET include:

• ***Unconstrained communication and connectivity***: Mobile devices can connect directly with each other without any hindrance or the need for fix communication infrastructure.

• ***Rapid and low cost of deployment compared to fix networks***: MANET only incurs cost for the mobile devices that is getting cheaper.

• ***Flexible and on-demand setup***: MANET can be instantly set up anywhere at any time.

1.6.5 Routing in MANET

Mobile Ad Hoc Networks (MANETs) have various advantages over a traditional wired networks, e.g. requiring no fixed infrastructure and allowing arbitrary movement and participation of mobile nodes. However, routing in MANETs faces many difficulties, e.g., frequent topology changes and a multiple access medium which is easily interfered by other radio signals. Many MANET routing protocols [10] have been proposed to handle these problems in the past. Unfortunately, they require a considerable amount of undesirable overhead especially in the unstable conditions like failures and mobility. Therefore, we aim at designing a MANET routing protocol which is adaptive against topology changes and packet collisions while having low overhead.

1.6.6 Routing Difficulties in MANET

1) Unpredictable dynamic changes in topology which is caused by the arbitrary movement and participation of the member nodes.

2) Decentralized control as each node has to maintain the route locally with only partial network information.

3) Limited bandwidth as the wireless channel uses a multiple access method, e.g., CSMA/CA, and each transmission interferes with each other.

4) Limited energy of the mobile node.

Due to these difficulties, a routing protocol should not create high control overhead due to limited bandwidth and energy despite the need of control mechanisms for handling the Network dynamics. Designing such routing protocols is a challenging task; therefore, one of the most active research activities on MANETs is on routing protocols.

1.6.7 USE OF ACO IN MANETS

Ant Colony Optimization (ACO) is a paradigm for designing meta-heuristic algorithms for combinatorial optimization problems. The first algorithm which can be classified within this framework was presented in 1991 [8, 9] and, since then, many diverse variants of the basic principle have been reported in the literature.

The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions. Ant Colony Optimization algorithms and their applications in Mobile Ad-hoc Networks have been studied for long. A couple of popular algorithms which aim to alleviate the problems in Mobile Ad-hoc Networks are the AntHocNet Algorithm and ARA (Ant colony-based Routing Algorithm). Both the routing algorithms extensively use the concept of pheromones and pheromone tables. AntHocNet, a more lucrative solution of the two, is a hybrid algorithm, which has both proactive and reactive features. Hybrid Algorithms, despite their better performance are not very popular due to the heavy overhead of the proactive algorithm (continuous Exchange of routing information, etc between neighbors).

1.6.8 Why ant colony optimization meta-heuristic suits to Mobile ad-hoc networks

The simple ant colony optimization meta-heuristic shown in the previous section illustrates different reasons why this kind of algorithms could perform well in mobile multi-hop ad-hoc networks. We will discuss various reasons by considering important properties of mobile ad-hoc networks.

Dynamic topology: This property is responsible for the bad performance of several routing algorithms in mobile multi-hop ad-hoc networks. The ant colony optimization meta-heuristic is based on agent systems and works with individual ants. This allows a high adaptation to the current topology of the network.

Local work: In contrast to other routing approaches, the ant colony optimization meta-heuristic is based only on local information, i.e. no routing tables or other information blocks have to be transmitted to neighbors or to all nodes of the network.

Link quality: It is possible to integrate the connection/link quality into the computation of the pheromone concentration, especially into the evaporation process. This will improve the decision process with respect to the link quality. It is here important to notice, that the approach has to be modified so that nodes can also manipulate the pheromone concentration independent of the ants, i.e. data packets, for this a node has to monitor the link quality.

Support for multi-path: Each node has a routing table with entries for all its neighbors, which contains also the pheromone concentration. The decision rule, to select the next node, is based on the pheromone concentration on the current node, which is provided for each possible link. Thus, the approach supports multi-path routing.

CHAPTER-2 LITERATURE REVIEW

In the forties and fifties of the twentieth century the French entomologist Pierre-Paul Grasse [34] observed that some species of termites react to what he called “significant stimuli”. He observed that the effects of these reactions can act as new significant stimuli for both the insect that produced them and for the other insects in the colony. Grasse used the term stigmergy to describe this particular type of communication in which the “workers are stimulated by the performance they have achieved”.

The two main characteristics of Stigmergy that differentiate it from other forms of communication are the following.

• Stigmergy is an indirect, non-symbolic form of communication mediated by the environment: insects exchange information by modifying their environment.

• Stigmergic information is local: it can only be accessed by those insects that visit the locus in which it was released (or its immediate neighborhood).

Examples of Stigmergy can be observed in colonies of ants. In many ant species ants walking to and from a food source deposit on the ground a substance called pheromone. Other ants perceive the presence of pheromone and tend to follow paths where pheromone concentration is higher. Through this mechanism, ants are able to transport food to their nest in a remarkably effective way.

Swarm Intelligence Algorithms in the early nineties of the last century have been suggested as representative of the ant colony algorithm and particle swarm algorithm has different characteristics in their application. Particle swarm algorithm is suitable for continuous optimization problems and ant colony algorithm is suitable for discrete optimization problems.

The ant colony algorithm has strong robustness as well as good distributed calculative mechanism and is easy to combine with other methods and the well performance has been shown on resolving the complex optimization problem. As a parallel searching algorithm ACO is more suitable for solving the optimization problems in complex environments. In ACO a set of software agents called artificial ants search for good solutions to a given optimization problem. To apply ACO the optimization problem is transformed into the problem of finding the best path on a weighted graph. The artificial ants incrementally build solutions by moving on the graph. The solution construction process is stochastic and is biased by a pheromone model that is a set of parameters associated with graph components (either nodes or edges) whose values are modified at runtime by the ants.

2.1 REVIEW OF ANT COLONY OPTIMIZATION ALGORITHM APPLICATIONS

Ant colony optimization algorithms have been applied to many combinatorial optimization problems.

A. Applications in Image Processing:

The thyroid nodule in the ultrasound images is usually blur and difficult to discern and analyze. A new algorithm which is based on the ant colony algorithm for extracting the edges of thyroid nodule in ultrasound images is presented. Taking the characters of ultrasound images into account the method of setting up the pheromone and the food-source is developed and the algorithm is elaborated. Compared with other image segmentation algorithms such as the Canny operator and the Sobel operator, ACO can achieve better segmentation results [35]. In [36], image recognition techniques of intelligent monitoring system were introduced and the image segmented by using ACO and rule-based inference was used to recognize intruders. After segmentation and getting the suspected light-area in the image the rule-based inference for intelligent recognition of the area was used. With the fuzzy C-means clustering (FCM) algorithm, it is difficult to determine the number of clusters in the research of medical image segmentation which is easy to get into a local optimum. To overcome these shortcomings a fuzzy C-means clustering algorithm is presented for image segmentation based on the ant colony algorithm in [37]. First, the overall robustness advantages of the ant colony algorithm are used to get the cluster centers and the number of the clusters of the image. Then the results are obtained as the initial cluster centers and the number of clusters. This algorithm overcomes the shortcomings of traditional ones and gets better results.

B. Applications in Scheduling Problems:

Several heuristic approaches have been proposed to solve the job-shop scheduling problem (JSSP). But most of them are limited to single objective and fail in real-world applications which naturally involve multiple objectives. [38] Presents an improved ant colony algorithm for solving multi-objective JSSP searching for near optimal solutions heuristically and optimizing multiple objectives simultaneously. By improving the transfer probability, pheromone update made and combining of cross optimum to make sure that this algorithm can accelerate convergence rate and avoid falling into local optima simultaneity. [39] Proposes a hybrid algorithm for job shop schedule. The ant colony algorithm which plunges the local situation easily is used as a global search algorithm. In addition it proposes taboo search algorithm based on a new neighborhood search method and the TS algorithms in the method are used as local search algorithm. Because the TS algorithms have the stronger local search ability and it can overcome the disadvantages of ant colony algorithms, it gets satisfied solutions for job shop scheduling.

C. Applications in Routing Problems:

Many of the existing proposed routing protocols could not give well stability and reliability and not fit in the needs for Ad hoc network. Because of the problems of great overhead and the lower stability in Ad hoc routing technology, an improved ant colony algorithm is proposed to study an ant-based Ad hoc routing protocol in [40]. Compared with the AODV(Ad hoc On-Demand Distance Vector) routing protocol which is a very mature strategy in Ad hoc study simulation results show that by bringing the node colony function into play the improved ant colony routing protocol can reduce the end--to--end delay and the routing overhead and increase the packet delivery rate. The network performances such as the stability and the efficiency are improved effectively. Most of the algorithms applied to the Quality of Service multicast routing problem are heuristic algorithms. Ant colony algorithm is a self-organized, novel heuristic algorithm based on ant colony system principle. Utilizing its capability of searching the shortest route proposed a Quality of Service multicast routing algorithm based on ant colony system to solve the delay and delay variates in constrained multicast routing problem. In order to support multicast efficient multicast routing is crucial, which is a process of establishing a tree rooted from the source node and containing all multicast destinations. In nature it is to construct a minimum cost tree namely Steiner tree. The research on distributed algorithm is relatively less. A novel distributed multicast routing algorithm based on ant colony algorithm is proposed in [41]. The algorithm tries to construct the minimum-cost tree by taking advantage of the local information and the pheromone left by previous ants in the case that the source node does not possess the whole network information. Combined the characteristic of multicast routing the algorithm is improved which accelerate the convergence speed and obtained better results.

D. Applications in TSP:

TSP is a classical NP-hard combinatorial optimization problem. Ant algorithm is a method for solving this problem. In order to adjust ant colony algorithm parameters hybrid intelligence technique based on ant colony swarm is introduced in [42]. The key parameters are taken as the attributes of ant instead of constant. The best ant individual is saved while the worst ant individual is abandoned via genetic operator. Parameter adaptation on dynamically occurs in parallel to the running of the hybrid algorithm. The value of recommendation trust is evaluated by probability average method in traditional recommendation trust model. But the method has low efficiency and is difficult to resist the unite cheat behavior. A searching trust path model based on ant colony algorithm is presented in [43], which is able to choice many better independence paths by a few circles. The algorithm is able to prevent unite cheat behavior in a certain extent.

E. Application in Vehicle Routing Problem:

The Vehicle Routing Problem (VRP) is one of the most challenging combinatorial optimization tasks. Defined more than 40 years ago, this problem consists in designing the optimal set of routes for fleet of vehicles in order to serve a given set of customers. The objective of the VRP is to deliver a set of customers with known demands on minimum-cost vehicle routes originating and terminating at a depot. For such problems it is often desirable to obtain approximate solutions, so they can be found fast enough and are sufficiently accurate for the purpose. Usually this task is accomplished by using various heuristic methods which rely on some insight into the problem nature. The VRP arises naturally as a central problem in the fields of transportation, distribution and logistics. The first ant system for VRP has been designed very recently by Bullnheimer who considered the most elementary version of the problem: CVRP [44]. For more complex versions of VRP, Gambardella, Taillard and Agazzi have developed a multiple ant colony system for VRPTW (MACS-VRPTW) which is organized with a hierarchy of artificial ant colonies designed to successively optimize a multiple objective function [45], the first colony minimizes the number of vehicles while the second colony minimizes the traveled distances. Cooperation between colonies is performed by exchanging information through pheromone updating. Compared with other approaches the ACO approach to solve the VRP is better as is shown in the experimental results.

Other Applications:

In addition to the above applications, ACO also can be applied to many other fields such as Robot Field, Power System, Fault Diagnosis, Data mining, Parameter optimization, System Identification, Clustering Analysis, Route Planning, Air combat Decision-making, Layout Optimization, Communication, water conservancy, Chemical, Architecture and traffic etc. Since the initial work of Dorigo and Maniezzo on ant system that is the first ACO system ACO is quickly developing in many fields. Many scholars have developed lots of models that were used to successfully solve a large number of complex combinatorial optimization problems.

2.2 CURRENT HOT TOPICS IN ACO:

A significant part of research on ACO is still concerned with applications as they have been presented in the previous section. However increasing attention is and will be given to even more challenging problems for example involve multiple objectives, dynamic modifications of the data, and the stochastic nature of the objective function and of the constraints. Other developments focus on the extension of the applicability of ACO algorithms from discrete to continuous optimization problems and to the study of parallel implementations of ACO algorithms.

A. Dynamic optimization problems

Dynamic problems are characterized by the fact that the search space changes during time. Hence while searching the conditions of the search the definition of the problem instance and thus the quality of the solutions already found may change. In such a situation it is crucial that the algorithm be able to adjust the search direction following the changes of the problem being solved. A paradigmatic example is routing in telecommunication networks [46]. ACO algorithms have also been applied to dynamic versions of the TSP, where either the distance between some pairs of cities changes [47], [48] or cities are dynamically added or removed from the set of cities to be visited.

B. Stochastic optimization problems

In stochastic optimization problems some variables have a stochastic nature. Apart from the network routing problems for which the main focus was put on their dynamic character the probabilistic traveling salesman problem (PTSP) was the first stochastic problem tackled by ACO algorithms. In the PTSP each city has a given probability of requiring a visit and the goal is to find an a priori tour of minimal expected length over all the cities with the strategy of visiting a random subset of cities in the same order as they appear in the a priori tour. The first ACO algorithm for this problem was proposed by Bianchi [49]. Further ACO algorithms for the PTSP have been proposed by Branke and Guntsch [50] and Gutjahr [51].

C. Multi-objective optimization

Multiple objectives can often be handled by ordering or weighting them according to their relative importance. In the two-colony ACS algorithm for the vehicle routing problem with time window constraints and in the MMAS for the bi-objective two-machine permutation flow shop problem [52], the multi-objective optimization problem is handled by ordering the objectives differently. Doerner [53] apply ACO to a bi-objective transportation problem and combine the objectives in a weighted sum. On the other hand if preferences or weights cannot be given a priori the goal is to find a set of non-dominated solutions that are optimal in the Pareto sense.

D. Parallel implementations

ACO algorithms lend themselves to be parallelized in the data or population domains. In particular any parallel models used in other population-based algorithms can be easily adapted to ACO. Two main strategies have been followed. In fine-grained parallelization very few individuals are assigned to single processors and information exchange among the processors is frequent. In coarse-grained approaches, on the contrary larger subpopulations are assigned to single processors and information exchange is rather rare. Research on parallel ACO algorithms has quickly shown that fine-grained parallelization results in a very significant communication overhead. Therefore the focus has mostly turned to coarse-grained parallelization schemes, where p colonies run parallel on p processors [54]–[55].

E. Continuous optimization

Recently ACO algorithms have been applied to continuous optimization. When an algorithm designed for combinatorial optimization is used to tackle a continuous problem the simplest approach would be to divide the domain of each variable into a set of intervals. However when the domain of the variables is large and the required accuracy is high, this approach is not viable. For this reason, ACO algorithms have been developed which are specifically designed for continuous and mixed continuous-discrete variables [56]. Research in this direction is currently ongoing.

2.3 OTHER ANT- INSPIRED ALGORITHMS:

The source of inspiration of ACO is the path marking behavior that some ant species exhibit when foraging. Nonetheless this behavior is not the only behavior of ants that has inspired computer scientists. Here are, in a very concise way some other examples of algorithms that are inspired by ants. The common trait of all these techniques is that they make use of stigmergic variables that is variables associated with the environment that hold the information that artificial ants share and exploit.

A. Other algorithms inspired by foraging and path marking

Apart from ACO a few other approaches take inspiration from the path marking behavior of ants. Two algorithms have been proposed for graph exploration: Edge Ant Walk [57] and Vertex Ant Walk [58]. In these algorithms ants mark with pheromone the edges they visit to coordinate graph exploration. Contrary to ACO in these algorithms the pheromones direct the ants toward unexplored areas of the search space. In fact their goal is to cover the graph is to visit all the nodes without knowing the graph topology.

B. Algorithms inspired by brood sorting

Brood sorting is an activity that can be observed in many ant species. These ants compactly cluster their smaller eggs and micro larvae at the center of the nest brood area and the largest larvae at the periphery of the brood cluster. Deneubourg [59] have proposed a model of this phenomenon in which an ant picks up and drops an item according to the number of similar surrounding items. Recently Handl [60] described an improved version of Lumer and Faieta's algorithm and compared its performance to other standard clustering techniques such as k-means. One of the salient features of this ant-based algorithm is its ability to propose a "natural" number of clusters.

C. Algorithms inspired by division of labor

In ant colonies individual workers tend to specialize on specific tasks in their lifetime. However ants can adapt their behavior to the circumstances: a soldier ant can become a forager, a nurse ant a guard and so on. This combination of specialization and flexibility is a desirable feature for multi-agent optimization and control, especially in task or resource allocation problems that require continuous adaptation to changing conditions. Many approaches inspired by division of labor in real ant colonies are based on a threshold model developed by Robinson [61] in which workers with low response thresholds respond to lower levels of stimuli than do workers with high response thresholds. Such a response-threshold model has been applied to the problem of choosing a paint booth for trucks coming out of an assembly line in a truck factory [62].

D. Algorithms inspired by co-operative transport

The behavior of ant colonies has also inspired research in robotics in particular for the design of distributed control algorithms for groups of robots [63]. An example of a task that has been used as a benchmark for ant algorithms applied to distributed robotics problems is co-operative box pushing [64]. Another example of application of ant algorithms is the one to the related problem of pulling an object. This has been achieved [65] within the Swarm-bots project (www.swarm-bots.org), a project dedicated to the study of ant algorithms for autonomous robotics applications.

CHAPTER-3 RELATED WORK

Mobile wireless devices are rapidly gaining popularity due to recent improvements in the portability and power of these products. There is a growing need for communication protocols which allow users of these devices to communicate over wireless links. Due to the NP complex nature of ad-hoc routing problem, it is pretty hard to get a perfect solution or even a better solution without increasing the bandwidth or packet overhead. All those problems led the researchers to work with Adaptive protocols for ad-hoc routing.

Along with the development of bionics and agent, ant colony algorithm and agent is applied to network's routing gradually. The first ACO routing algorithm was designed for wired networks like as AntNet, which is a proactive routing protocol, mainly using ants to discover and maintain route, but it doesn't adapt the networks that constantly change, so doesn't apply to Ad Hoc networks [11]. The ACO routing algorithms were designed for mobile Ad hoc networks such as Ant-AODV [12] and AntHocNet [13,14] all use mobile agents like ants to carry on big scope to scan the networks, make use of ant agents to collect the information about routing. Among them, the Ant-AODV combines ACO proactive characteristics with AODV reactive characteristics, to improve the performance of routing search delay and to adapt the constantly change of network's topology; AntHocNet is a hybrid algorithm, which combines reactive path setup with proactive path probing, maintenance and improvement.

Several protocols have been proposed for the routing problem in MANETs. These protocols are usually categorized according to their design approaches as proactive, reactive or hybrid [15]. A proactive (table-driven) protocol is one in which the routing information is maintained for all the paths from source to destination, whether in use or not. Protocols such as Destination Sequenced Distance Vector routing DSDV [16], Global State Routing (GSR) [17], Fish-eye State Routing (FSR) [18], Wireless routing protocol (WRP) [19] fall under proactive category. In a reactive (on-demand) protocol only routes currently in use are maintained and they are created only when needed. Ad hoc On-Demand Distance Vector (AODV) [20], Dynamic Source Routing (DSR) [21], Temporally-Ordered Routing Algorithm (TORA) [22] are examples of reactive routing protocols. Hybrid protocols, attempt to, fuse proactive and reactive routing to achieve better performance. Zone routing Protocols (ZPR) [23], Virtual Backbone routing (VBR) [24] are categorized as hybrid protocols.

The first attempt at using ACO in communication was done by Schoonderwoerd [25] on the Ant Based Control (ABC) algorithm for a wired circuit switched network. They were able to optimize performance in the network by balancing the load in the network. This work was followed by the AntNet algorithm introduced by Di Caro and Dorigo [26] for adaptive routing in packet switching networks.

Camara and Loureiro [27] describe a novel routing protocol called Global Positioning System ANT-Like Routing Algorithm (GPSAL) which uses a Global Positioning System and mobile software agents modeled on ants for routing. Ants are used to collect and disseminate information about the location of nodes in the MANET while the GPS provides the physical location of a destination node. This algorithm focuses on position based routing and so does not consider the concentration of pheromone on any paths.

Gunes [28] were the first to propose a routing algorithm based on the idea of AntNet for MANETs. Their algorithm, Ant-colony based Routing Algorithm (ARA) relies on a backward and forward ant and consists of 3 phases: route discovery, route maintenance and route failure handling. The forward ant acquires information about the network from source to destination while the backward ant updates the routing information collected by the forward ant for each node from destination to source, establishing the path. ARA reduces routing overhead but it is not scalable and cannot detect cycles.

The Accelerated Ants-Routing algorithm discussed by Matsuo and Mori [29] uses ant like agents that go through the network randomly, without a specific destination, updating pheromone entries pointing to their source (the source of the agents is represented as destination in the entries). It is an enhancement of the ARA and a modification of the Ants-Routing algorithm [30]. The Accelerated Ants-Routing algorithm is proposed to accelerate the converging speed of the routing table to obtain the optimum routing paths.

Ahmed [31] studied the performance issues related to routing in MANETs. The author develops a routing algorithm based on ACO and Klein-rock’s delay analysis [32] and in particular focuses on the end-to-end delay in MANETs. The author combines mobility modeling with queuing network analysis to evaluate end-to-end delay. Based on the experiments, the author is able to determine that the algorithm improves system performance of a network and that there is no significant effect on delay among different mobility models considered. The performance of this algorithm is yet to be compared to other routing algorithms.

Ant Routing Algorithm for Mobile Ad hoc net-work (ARAMA) [33], is a combination of on-demand and table driven algorithms. The focus of this paper is on optimizing different Quality of Service (QoS) parameters, other than number of hops. Such parameters include energy, delay, battery power, mobility etc. The paper proposes a path grading enforcement function that can be modified to include these QoS parameters. One of the important attributes of this algorithm is that the lifetime of the ad hoc nodes have been extended by using a fair distribution of energy across the network.

CHAPTER-4 PROPOSED ALGORITHM

4.1 Motivation

Proposed algorithm is a routing protocol for mobile ad hoc networks inspired by the foraging behavior of ants. It uses the principles of ACO routing to develop a suitable problem solution. It uses ant agents.

4.2 Proposed algorithm

The ants explore the paths of the network in a restricted broadcast manner in search of routes from a source to a destination and establish the path information that is acquired by them. These agents create a bias at each node for its neighbors by leaving a pheromone amount from its source. Data packets are stochastically transmitted towards nodes with higher pheromone concentration along the path to the destination.

Data structure maintained:

1. Routing Table
    - 1.1 Routing table distance
    - 1.2 Routing table from
2. Pheromone Table
3. Neighbor Table
4. Node-Visited Table
5. Network
6. Node Travelled
7. Probability Matrix
8. Path length
9. Source Node

10. Destination Node

Routing Table:

Each node in the network has a routing table whose size is the degree of the node times all the nodes in the network. That is, if the number of nodes is N and the degree of node v_i is d_i then the size of the routing table is N×d_i . In other words we can say that routing table is a matrix of the size equal to number of nodes in the network. Routing table has two sub tables.

Routing Table Distance:

In this table each entry represents the length of shortest path if exist between the nodes represented by rows and columns of the routing table. For example if there exist a path between nodes i and j then the entry corresponding to the i^{th} row and j^{th} column is that distance.

Routing Table From:

This table is helpful to calculate the path between the nodes. Each entry in this table tells that the path exists between the two nodes is via this node. For example let suppose the path between the nodes i and j is via node k then the corresponding entry gives the value as node k.

Pheromone Table:

Paths are implicitly defined by the pheromone tables which are kept locally at each node. An entry τ_{ij} in the pheromone table at node contains a value indicating the estimated goodness of going from i to j. More the goodness value more is the chances of selecting that path. Initially all the entries in this table are same and equal to 1.

Neighbor Table:

This table tells about the neighbors of a given node. Because in MANET nodes have mobility means they don't have any fixed location, their position changes continuously. So here it is more important to find that which nodes are connected at a particular time. It have entry 1 for neighbor which is connected, 0 for the same node and -1for the nodes that are not connected means 1 represent the neighbor and -1 not a neighbor.

Node-visited Table:

This table tells about the nodes which are already visited by the agents. This plays an important role in preventing the agents from going in loops. This means that once a node is visited by the agent it is marked so that it should not be visited again by the same agent. Visited nodes are marked as 1. So when an agent moves from one node to another then firstly it checks that whether that node is visited or not, if not visited then consider it else leave it and search for the next feasible node.

Network:

This tells about the network structure. In MANET nodes are not static means their position remains changing. For this purpose we can't take the static structure of the network, that why we have taken it as dynamic. We have used a random number generator for this purpose.

Node-Travelled:

This is the data structure which is used for storing the nodes which are travelled by the agent while its journey from source node to the destination node. This gives the final path from source to destination. While the agent is moving from one node to another node during its journey these visited nodes are stored and maintained in this data structure.

Probability Matrix:

Each node is associated with some probability which tells about its goodness value. More the probability more its goodness value hence more are the chances of selecting this node as next node. In our case the probability is a function of pheromone and visibility. Out of the feasible nodes the node having the highest probability is selected as the next node.

Path Length:

This tells about the length of the path travelled by the agent. This length can be function of number of hopes or time elapsed or distance between the nodes.

Source Node:

This tells about the source node in the network from where the agent will start its journey.

Destination Node:

This node tells about the destination node in the network where the agent has to reach. Once the destination node is found the agent stops.

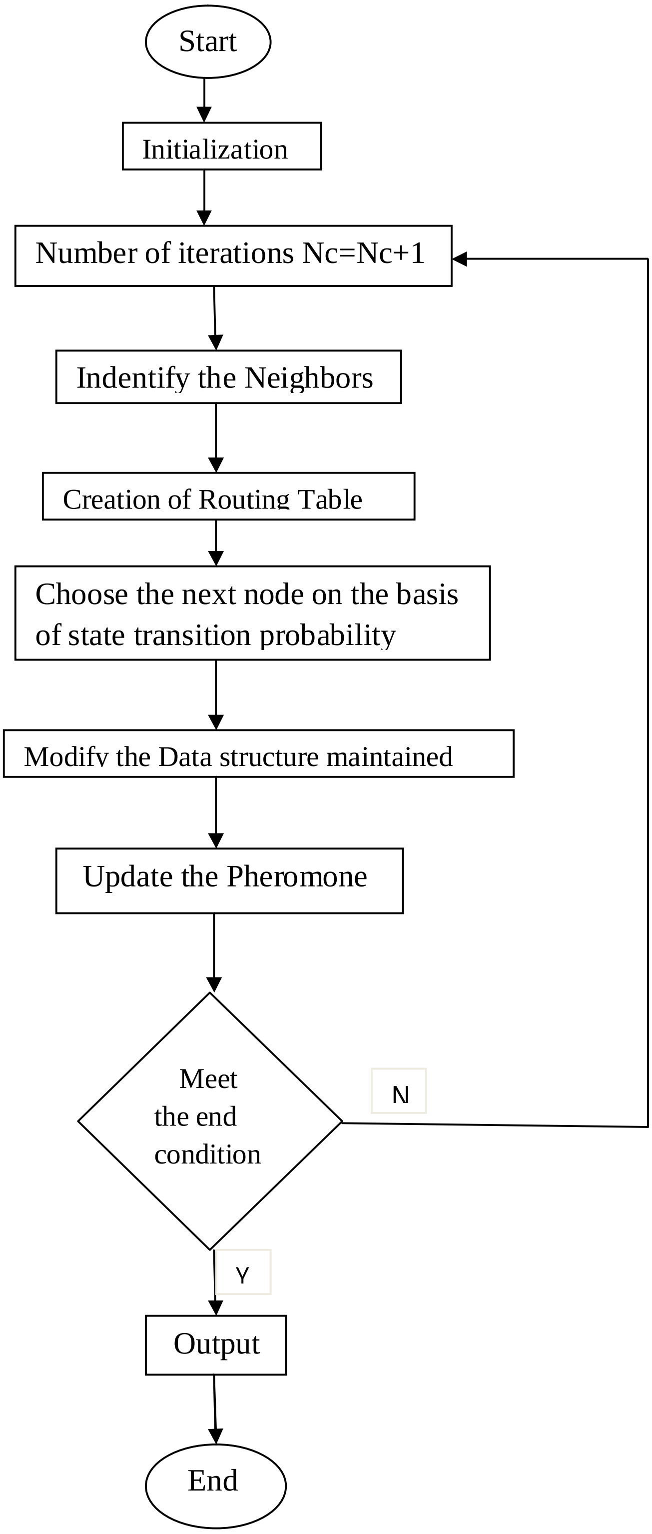

Figure: 4.1 Flow-chart of the Proposed Algorithm

Assuming that ***s*** is the source node and ***d*** is the destination node this algorithm will give the route from source to destination. The proposed algorithm can be performed as the following steps:

i. Start.

ii. Initialization.

iii. Set NC=0. (Loop counter.)

iv. Identify the Neighbors.

v. Routing table formation.

vi. Repeat until next node equal to destination node

    a. Find the probability of the all neighbor nodes of current node.

    b. Select the next node with the maximum probability.

    c. Insert this node in Node travelled and Make it as current node.

vii. Calculate the path length.

viii. Update the pheromone table.

ix. Repeat until NC=NC_{max}.

x. Stop.

Given the definitions of the preceding section, the algorithm is simply stated as follows:

In the initialization step we initialize the data structure required with their initial values. For example: the pheromone table is initialized with value 1 at each entry. This is the initial value of pheromone. This is done because initially all the paths have equal probability to be get selected. Diagonals of the pheromone matrix are always equals to 0 because this is the entry for node with itself. Network table is initialized with random values which are the distances between the nodes, for this purpose we use a random number generator. We also initialize the source node and the destination node in the network. Node visited table is initially empty because in the starting no node is visited. As agent proceeds the nodes which are visited by the agent are marked as visited in this table. The probability matrix is also initialized with 0 values at the starting, after the probability of each node is calculated its values are updated. The path length is also initialized with its initial value of 0. Its final value is the total path length of the tour travelled by the agent from going source to destination. Node travelled table is also initialized by 0 initially. When nodes are travelled by the agent, these are inserted in to the table one by one. This is one dimension table which returns the final path of the ant from source to destination at the end of algorithm means the intermediate nodes which are travelled by the agent to reach to the destination. NC is the number of iterations we want to make. NC is initialized with 0 and having its maximum value NC_{max}. This means the total number of iterations is NC_{max} which is inputted by the user.

After initialization the next step is to identify the neighbors. As the nodes in the MANET are not static, they change their physical location by moving around. This makes task more difficult, because the node which is neighbor of some node now, after some time it no longer be the neighbor to that node. For achieving this, we have used a random number generator which generates the random numbers. After that on these numbers we have applied a threshold. The nodes which satisfy this threshold values are to be considered up and other are considered as down. In the neighbor table up nodes are represented by 1 and down are represented by -1. Now by looking the entries in the neighbor table we can say find the neighbors of any given node. If that entry is 1 means the two nodes are neighbors and if the entry is -1 then they are not neighbors. Point to note here is that the diagonal entries of the neighbor table are 0's because this represent the relationship of the node with itself.

After knowing the neighbors the next step is the formation of routing table. Each node has its own routing table which has the information about the nodes which can be reached by this node. Initially node has no information about its neighborhood. Each node sends its own information to its neighbors and also gets their local information. By doing so it got to know about its surrounding. For this purpose node maintains a routing table, which have the entries of the nodes which can be reachable via this node. With this information it also has the detail of the intermediate node via which that particular node is reached. The mutual exchange of information between the nodes is performed after a particular time interval and the old values are updated with the new ones. That's why in our algorithm we have used two sub-routing table. One namely routing table distance have the entries of the nodes which can be reached by the given node with it cost. Here we are considering the cost as the distance between the nodes. The other one namely routing table from tells about the intermediate nodes by the help of which the required node is reached.

Once the routing table is maintained, next step is to find the Next hope means the node with the maximum probability. Make the source node as current node. Mark it as visited. Insert the source node in the node travelled table. Now calculate the probability for all the neighbors of the current node using equation-(1).

$$P_{ij}=\frac{(\tau_{ij})^{\alpha} * (\eta_{ij})^{\beta}}{\sum_k (\tau_{ik})^{\alpha} * (\eta_{ik})^{\beta}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(1)}$$

P_{ij}= Transition probability from node i to node j.

η_{ij}= Visibility i.e. $1/d_{ij}$. d_{ij} = distance between the node i and j.

τ_{ij}= Pheromone on the edge between the nodes I and j.

α, β= controlling parameters that control the relative importance of pheromone versus visibility.

Here the probability is the function of the pheromone and the visibility. We have two separate parameters for controlling the pheromone and visibility. By using these two parameters we can control the contribution of each in calculating the probability. Parameters can be chosen according to the desired need. Probability can be calculated as the goodness of the particular node to the sum of goodness of all the neighbor nodes of the current node. With the node having highest probability we also have put some extra conditions like that this node should be a neighbor node and should not be visited earlier. If all the conditions are true then only select this node as next node else choose another node in same manner.
After selecting the next node we have to update the data structure maintained, Such as make the next node as current node. Mark it as visited. Insert this node to node travelled table. Now in similar fashion find the next node until we get the destination node. After reaching to the destination node from the node travelled table find the path from source to destination. Path is the sequence of nodes travelled starting from source with intermediate nodes to the destination node. After that, find the path length, i.e. the total distance from source to destination.

Next step is to update the pheromone. The pheromone is updated using the equation-(2) given below.

$$\tau_{ij=(1-\rho)}\tau_{ij+}\,\Delta\tau_{ij} \qquad \text{………………………………..…… (2)}$$

$\Delta\tau_{ij}$ = 1/Path-length: if the edge between i and j is in the path.

Otherwise $\Delta\tau_{ij}$ =0

We know that, the pheromone evaporates with time. Hence the pheromone concentration goes on decreasing as the time passes. But by the same time if any ant visits any edge during this time then the concentration of the pheromone on that particular edge (path) increases. This increase in concentration is an indication that the particular edge has the more probability of being chosen in the solution as compared to the others. So by keeping this property in mind, we have updated the pheromone accordingly. First of all we have decreased the pheromone concentration on every edge according to the pheromone evaporation rate. Here ρ is the pheromone evaporation rate. The equation-(2) uses the evaporation rate to decrease the pheromone concentration on every edge. While finding a route to the destination node if any ant visits any edge in the network, the concentration of pheromone increases on that particular edge, making it more favorable than the others. This increases in concentration is shown in equation-(2) by the $\Delta\tau_{ij}$. By using the equation-(2), we have increased the pheromone concentration on the edges which are used by the ant in their final path to the destination. As from the above equation it is clear that the pheromone increase is done only on the edges which are in the final path. If any edge is used by the ants in their final path then only the pheromone concentration on that edge is increased and if any edge is not used by the ant then on that edge no increment is done i.e. $\Delta\tau_{ij}$ is equal to zero for these.

For example, let the entry corresponding to the i^{th} row and j^{th} column in the pheromone table means the pheromone value on the edge between the nodes i and j is updated using the equation-(2). Then first of all the pheromone concentration on the edge between the nodes i and j is decreased using the evaporation rate constant. Now after decreasing the pheromone concentration a check is made whether the edge joining the nodes i and j is in the final path of the ant to the destination. If it is so, then its pheromone concentration on this edge is increased using $\Delta\tau_{ij}$ and if this edge is not in the final path to the destination then the value of $\Delta\tau_{ij}$ is zero i.e. no increment is done.

The experimental results are as follows:

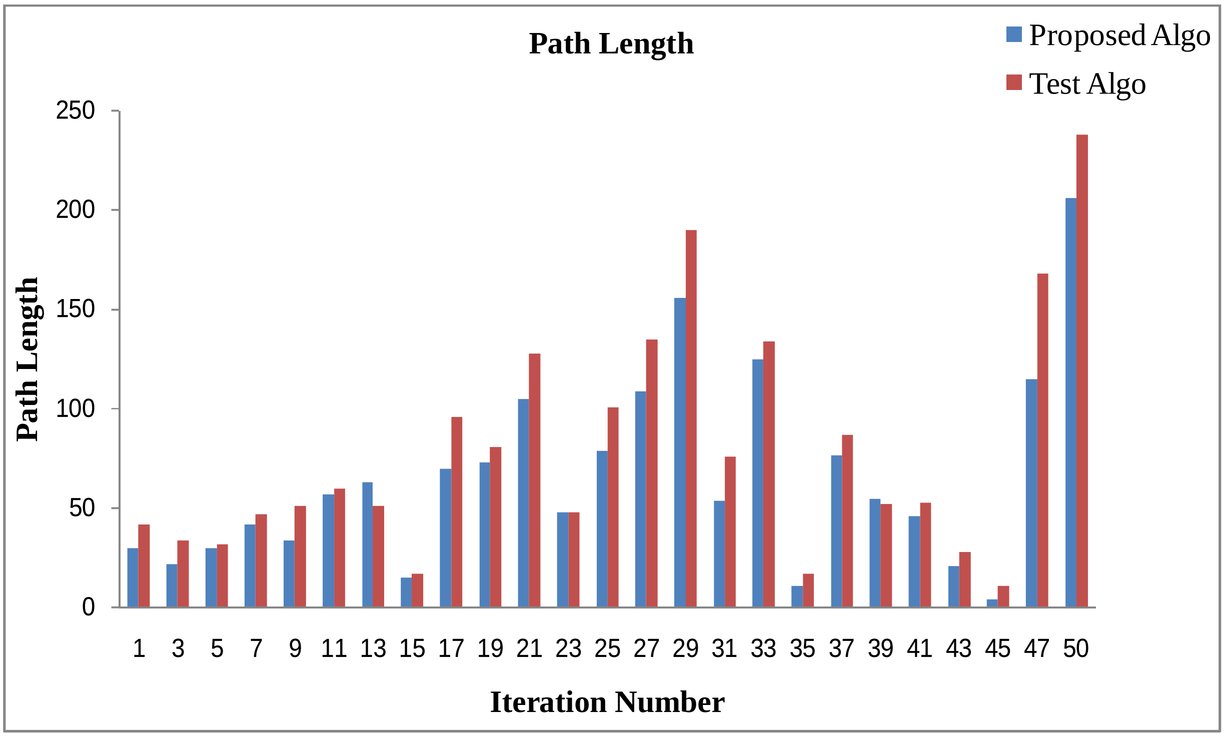

Figure 5.1 Graph showing the Path Length Vs Iteration number

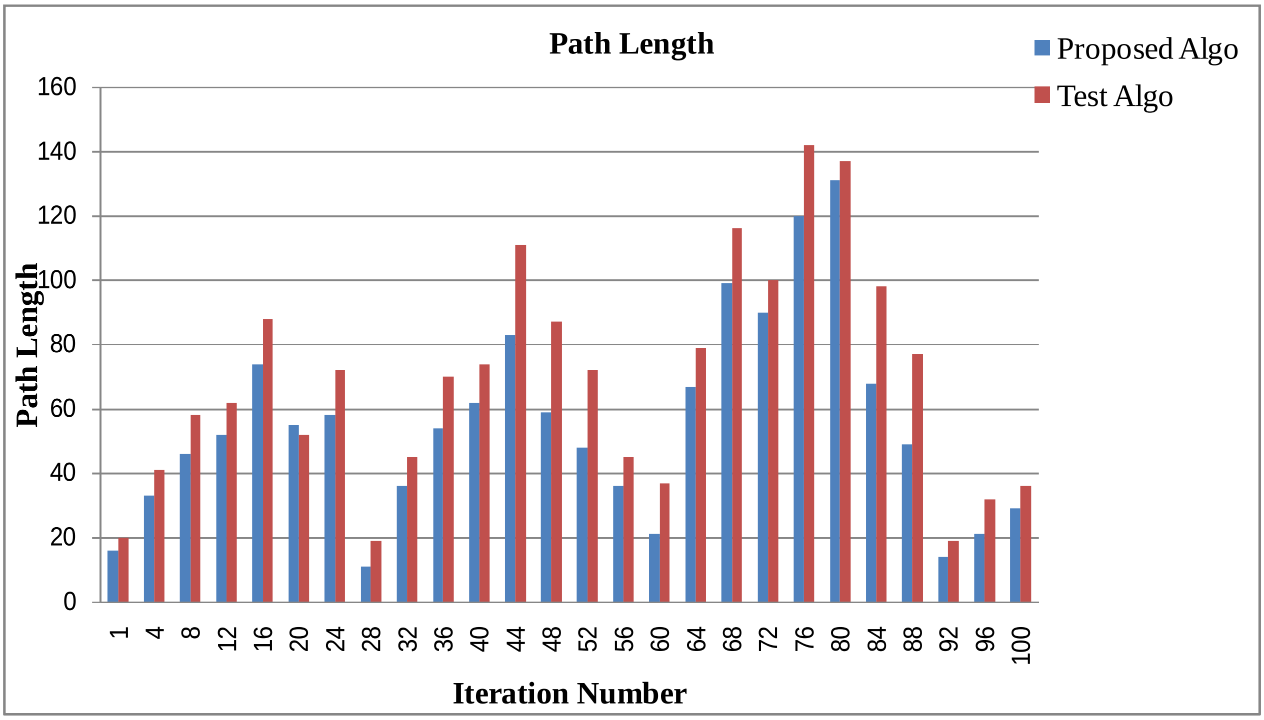

Figure 5.2 Graph showing the Path Length Vs Iteration number

Iteration Number	1	8	16	24	32	40	48	56	64	72	80
Proposed Algorithm	16	46	74	58	36	62	59	36	67	90	131
Test Algorithm	20	58	88	72	45	74	87	45	79	100	137

Table 5.1 Path lengths of the two algorithms at diff. Iteration Numbers

The above two graphs show the comparison between the proposed algorithm and the test algorithm on the basis of length of the route calculated from source node to destination node. The horizontal axis in the graph represents the iteration number and the vertical axis shows the length of the route calculated from source node to the destination node. From both graphs it is clear that the proposed algorithm gives the better results than the test algorithm. The proposed algorithm gives the path of shorter length as compared to the path given by test algorithm.

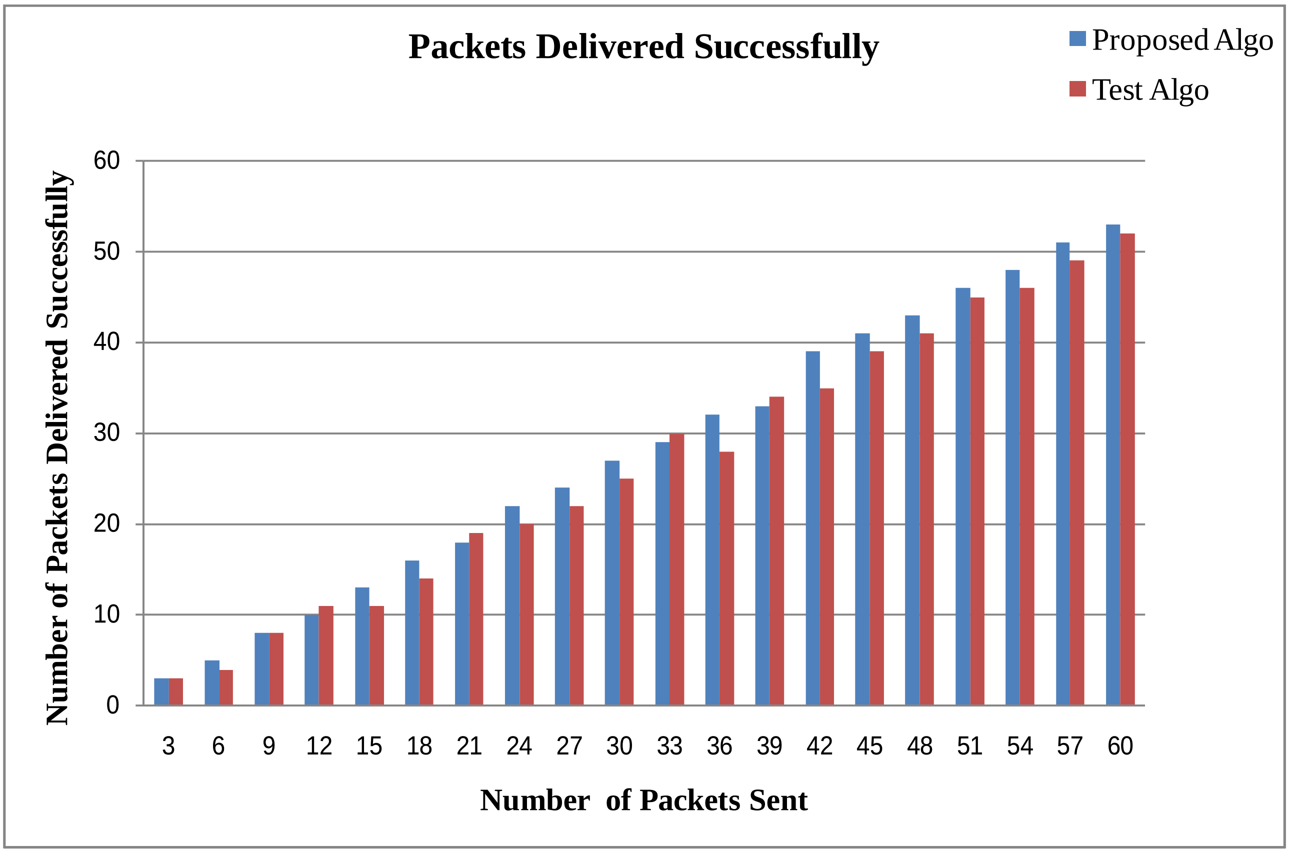

Figure 5.3 Graph showing the Packets delivered successfully Vs Packets Sent

Number of Packets sent	1	6	12	18	24	30	36	42	48	54	60
Proposed Algorithm	1	5	10	16	22	27	32	39	43	48	53
Test Algorithm	1	4	11	14	20	25	28	35	41	46	52

Table 5.2 Number of Packets delivered successfully Vs number of Packets sent

The above graph shows the comparison between the proposed algorithm and the Test algorithm on the basis of Number of packets sent and the Number of packets reached to the destination successfully. The horizontal axis in the graph represents the number of packets sent and the vertical axis shows the number of packets reached to the destination successfully. From the graph it is clear that the proposed algorithm outperform the test algorithm in most of the cases in the successful transmission.

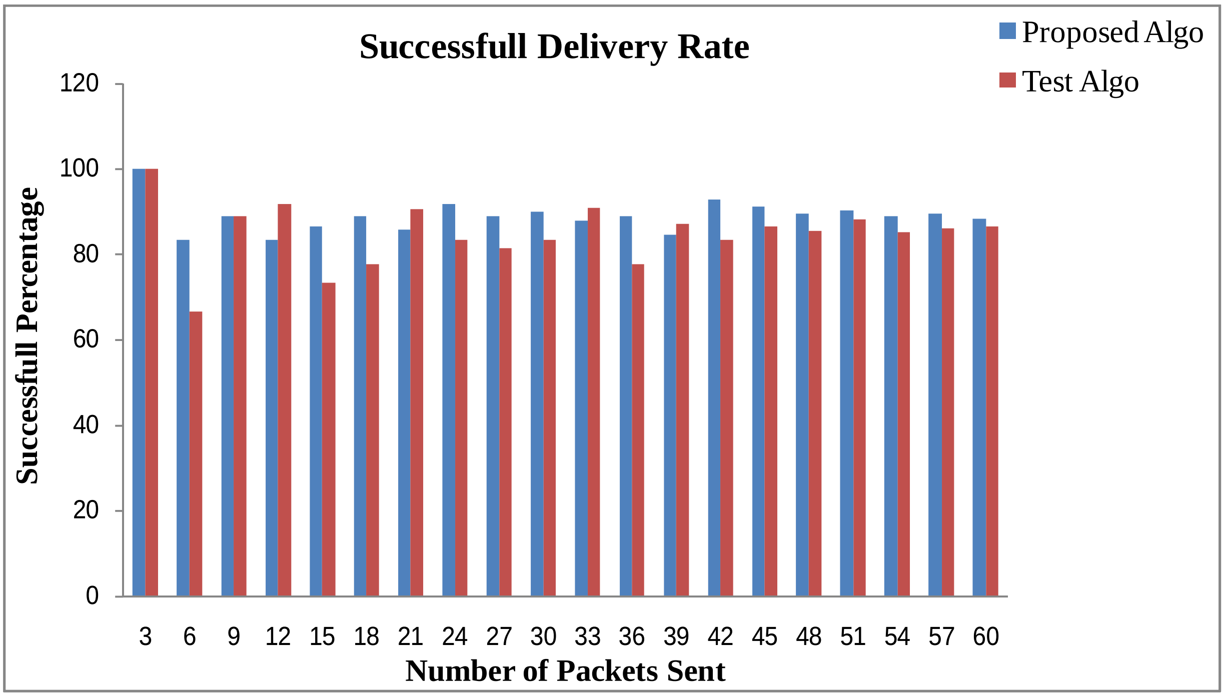

Figure 5.4 Graph showing the Percentage of packets delivered Successful

Number of Packets sent	1	6	12	18	24	30	36	42	48	54	60
Proposed Algorithm	100%	83.3%	83.3%	88.8%	91.6%	90%	88.7%	92.8%	89.5%	89.4%	88.3%
Test Algorithm	100%	66.6%	91.6%	77.7%	83.3%	84.3%	77.7%	84.4%	85.4%	85.9%	86.6%

Table 5.3 Percentage of Packets delivered successfully Vs number of Packets sent

The above graph shows the comparison between the proposed algorithm and the test algorithm on the basis of successful percentage delivery i.e. the number of packets reached successfully to the destination for every hundred packets. The vertical axis in the graph represents the successful delivery rate and the horizontal axis shows the number of packets sent. The percentage decreases as the number of packets sent increased. This may be due to some other network characteristics for example collision or congestion. Still our proposed algorithm have more successful rate as compared to the test algorithm.

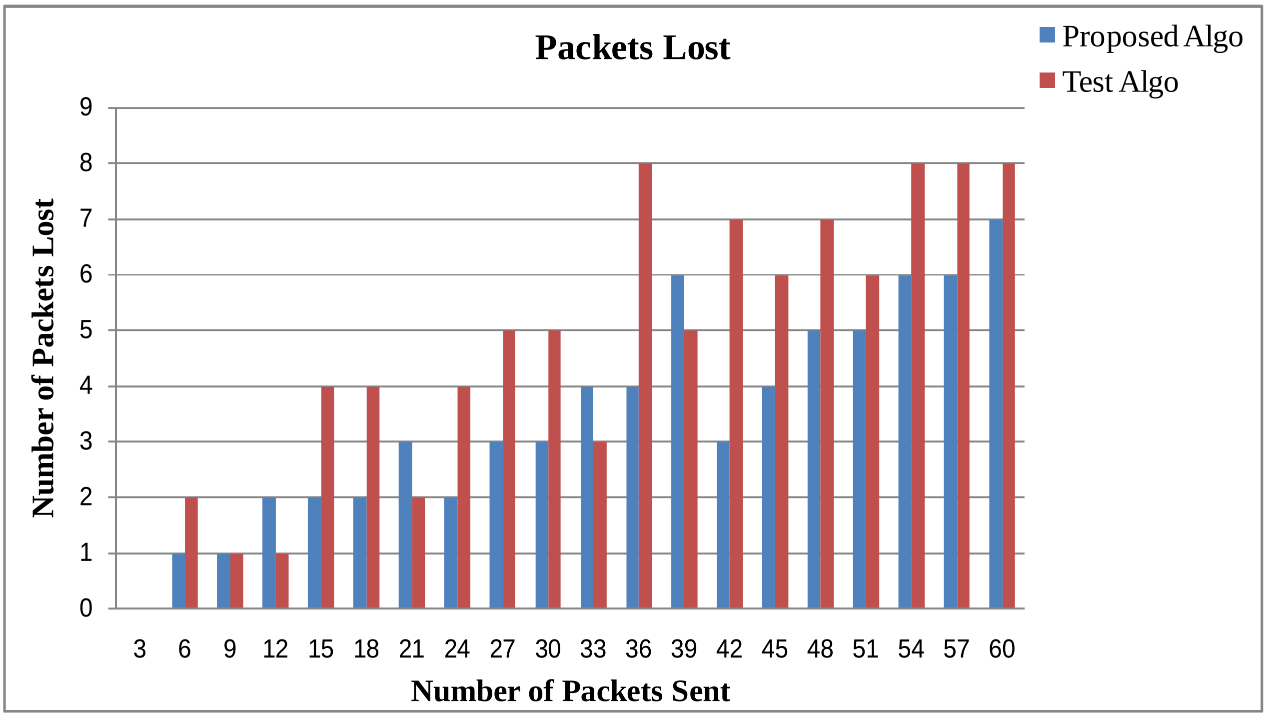

Figure 5.5 Graph showing the number of Packets lost Vs Number of Packets Sent

Number of Packets sent	1	6	12	18	24	30	36	42	48	54	60
Proposed Algorithm	0	1	2	2	2	3	4	3	5	6	7
Test Algorithm	0	2	1	4	4	5	6	7	7	8	6

Table 5.4 Number of Packets lost Vs number of Packets sent

The above graph shows the comparison between the proposed algorithm and the test algorithm on the basis of Number of packets lost during the transmission. The vertical axis in the graph represents the number of packets lost during the transmission and the horizontal axis shows the number of packets sent. The graph shows the packets that are lost or that are not delivered successfully. The packet lost can be because of the collision, congestion or channel overflow. Form the graph we can say that as we increases the number of packets sent the number of packets lost also increases. But our algorithm has fewer packets lost during the transmission.

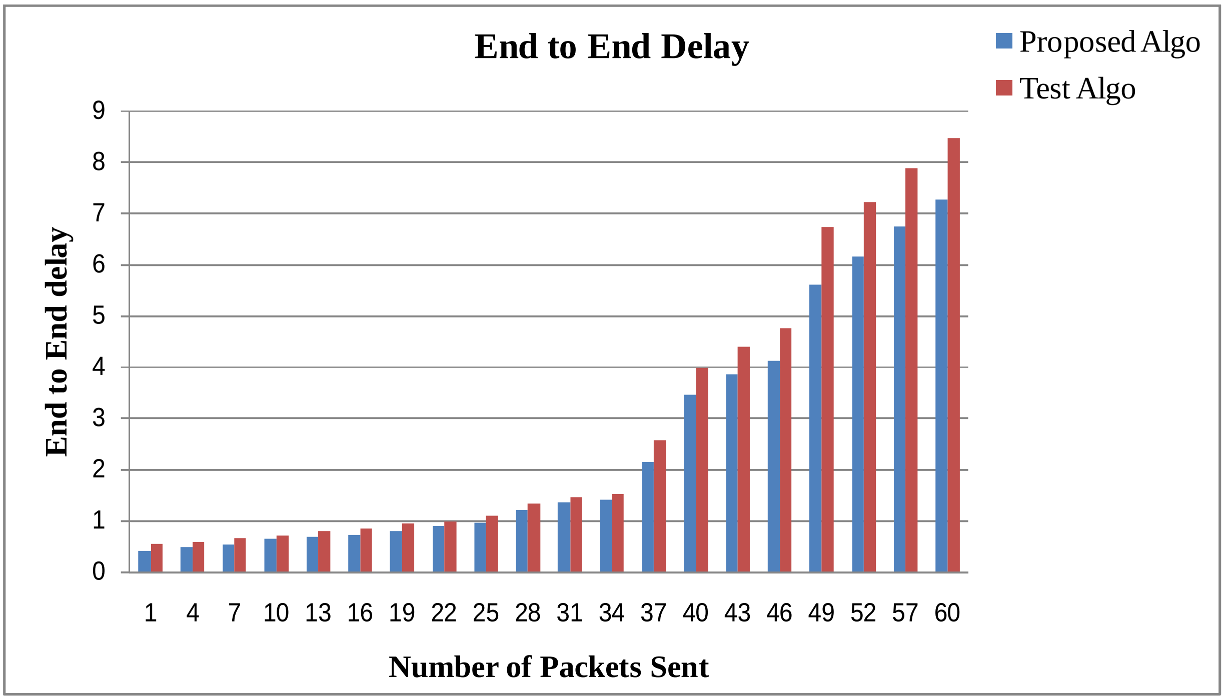

Figure 5.6 Graph showing the End to End Delay Vs Number of Packets Sent

Number of Packets sent	1	4	10	16	25	31	37	43	49	57	60
Proposed Algorithm	0.42	0.496	0.654	0.735	0.968	1.374	2.157	3.865	5.615	6.763	7.284
Test Algorithm	0.56	0.594	0.712	0.861	1.112	1.466	2.584	4.408	6.749	7.895	8.471

Table 5.5 End to end delay (secs) between packets Vs number of Packets sent

The above graph shows the comparison between the proposed algorithm and the test algorithm on the basis of end to end delay. The vertical axis in the graph represents the end to end delay in seconds and the horizontal axis shows the number of packets sent. From the graph it is clear the delay increases in both cases as the number of packets sent increases. But our proposed algorithm has less delay as compared to the test algorithm.

Figure 5.7 Graph showing the Execution Time Vs Number of Iterations

Number of Iterations	1	4	10	16	25	31	37	43	49	57	60
Proposed Algorithm	0.342	0.484	0.612	0.668	0.735	0.812	0.968	1.194	1.372	1.532	1.675
Test Algorithm	0.432	0.521	0.694	0.861	0.956	1.0	1.12	1.347	1.538	1.881	1.987

Table 5.6 Execution time of the algorithm Vs Number of Iterations

The above graph shows the comparison between the proposed algorithm and the test algorithm on the basis of execution time of algorithm. The vertical axis in the graph represents the execution time in seconds and the horizontal axis shows the number of iterations. From the graph it is clear has less execution time as compared to the test algorithm.

CHAPTER 6 CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In this thesis, we have suggested a new routing algorithm for optimizing the route from source node to destination node in Mobile Ad-hoc Networks. We have used Ant Colony Optimization technique because the characteristics of it suites the MANET routing requirements. Work done by Wan-Jun Yu, Guo-Ming Zuo and Qianq-Qian Li for Routing in Mobile Ad-hoc Networks is our test algorithm. The algorithm given by them is based on the probabilistic approach and the probability is the function of Pheromone alone. Our proposed algorithm is also based on probabilistic approach and but in our case the probability is the function of the pheromone and the visibility. By taking visibility in to the consideration along with pheromone value we are selecting the next node which is close to the current node in the distance, as the visibility is the function of distance. By doing so, we are selecting the node having less distance to the current node but having comparative low value of pheromone over the node having high value of pheromone but having more distance to the current node. Hence our algorithm gives the route to the destination with smaller path length. Because of the smaller path length, the end to end delay between the packets is also less. As we are taking both pheromone and visibility in the probability selection, there are more chances of selecting the right path to the destination. Because of this, our algorithm has better packet delivery rate means packets delivered to the destination successfully is more in our case. We have compared the results on the basis of parameters like path length, number of packets delivered successfully to the destination, successful packet delivery rate, end to end delay and execution time of algorithm for the performance evaluation and it is found that in each case our proposed algorithm gives the better results than the test algorithm.

6.2 Future Work

Because of mobility of nodes in Ad-hoc networks, the related-nodes (Neighboring Nodes) of any given node will be spread to other place in network. Routing Table Update interval and the number of Routing Tables can affect the Density and the Diffusion Radius of related-nodes. Our future work is to find an optimal routing algorithm with nodes moving at variable speed in these types of networks. Also our focus will be on improving the slow convergence speed of the ACO.

References

[1] E.Bonabeau, M. Dorigo, and G. Theraulaz, "Swarm intelligence: from natural to Artificial intelligence," *Oxford University Press, 1999,* ISBN 0-19-513158-4.

[2] M. Dorigo, V. Maniezzo, and A. Colorni, "Positive feedback as a search strategy," *Dipartimento di Elettronica, Politecnico di Milano, Italy,* Tech. Rep. 91-016, 1991.

[3] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant System: Optimization by a colony of Cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics—Part B,* vol. 26, no. 1, pp. 29–41, 1996.

[4] T. St¨ utzle and H.H. Hoos, "MAX–MIN Ant System," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889–914, 2000.

[5] M. Dorigo and L.M. Gambardella, "Ant colonies for the traveling salesman problem," *BioSystems, vol. 43, no. 2, pp. 73–81, 1997.*

[6] "Ant Colony System: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[7] L.M. Gambardella and M. Dorigo, "Solving symmetric and asymmetric TSPs by ant colonies," in *Proc. 1996 IEEE International Conference on Evolutionary Computation (ICEC'96),* T. Baeck et al., *Eds. IEEE Press, Piscataway, NJ*, pp. 622–627, 1996.

[8] Gunes, M. Sorges, U. Bouazizi I., "ARA-the ant-colony based routing algorithm for MANETs," *Parallel Processing Workshops, 2002. Proceedings. International Conference* on 18-21 Aug. 2002 Page(s):79-85.

[9] V.D. Park, M.S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," *Proc. IEEE INFOCOM – The Conference on Computer Communications, Kobe, Japan*, , pp. 7–11, April 1997.

[10] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications, 1999*.

[11] Zheng Xiang quan, Guo Wei, Ge Lijia and Liu Renting, "A Cross-Layer Design and Ant-Colony Optimization Based Load-Balancing Routing Protocol for Ad Hoc Network (CALRA)," [J]. *Chinese journal of electronics 2006.7 (7):* 1199-1208.

[12] S Marwaha, C K Tham and D Srinivasan, "A Novel Routing Protocol using Mobile Agents and Reactive Route Discovery for Ad-hoc Wireless Networks[C]," *Toward Network Superiority Proceedings of IEEE International Conference on Networks 2002(ICON 2002),* August 27-30, 2002:311-316.

[13] Di Caro, G., Ducatelle, F., Gambardella, L., "AntHocNet: an adaptive nature inspired algorithm for routing in mobile ad hoc networks," *European Transactions on Telecommunications, Special Issue on Self Organization in Mobile Networking 16(5)* (2005) 443–455.

[14] Di Caro, G., Ducatelle, F., Gambardella, L., "Swarm intelligence for routing in mobile ad hoc networks," *Proceedings of the 2005 IEEE Swarm Intelligence Symposium (SIS).* (2005):76-83

[15] C. K. Toh, "Ad Hoc Mobile Wireless Networks: Protocols and Systems," *Prentice Hall*, December 2001.

[16] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *ACM SIG-COMM'94 Conference on Communications Architectures, Protocols and Applications*, volume 24, pages 234–244, New York, NY, USA, 1994.

[17] T. Chen and M. Gerla, "Global state routing: A new routing scheme for ad-hoc wireless networks," *Proceedings of the IEEE international conference on communications (ICC),* pages 171–175, Atlanta, GA, June 1998.

[18] G. Pei, M. Gerla, and T. Chen, "Fish eye state routing in mobile ad hoc networks," *Proceedings of the 20th IEEE international conference on distributed computing systems (ICDCS) workshop on wireless networks and mobile Computing,* pages 71–78, Taipei,Taiwan, April 2000.

[19] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. Mobile Networks and Applications, 1(2):183–197, 1996.

[20] C. Perkins and E. M. Royer, "Ad–hoc on-demand distance vector routing," *Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, LA, February 1999.

[21] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *T. Imielinski and H. Korth, editors, Mobile Computing*, volume 353, chapter 5, pages 153–181. Kluwer Academic Publishers, Boston, 1996.

[22] Qiang Liu, Hua Wang, Jingming Kuang, Zheng Wang and Zhiming Bi, "A TORA-Based Multi-Path Routing Algorithm for Mobile Ad Hoc Networks," *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*

[23] Z. J. Haas, "A new Routing Protocol for Reconfigurable Wireless Networks," *Proceedings of IEEE International Conference on Universal Personal Communication*, pages 1–11, Atlanta, GA, October 1997.

[24] B. Liang and Z. Haas, "Hybrid routing in ad hoc networks with a dynamic virtual backbone," *IEEE Transactions on Wireless Communications*, pages 1392–1405, June 2006.

[25] R. Schoonderwoerd, O. E. Holland, J. L. Bruten, and L. J. M. Rothkrantz, "Ant-Based Load Balanc-ing in Telecommunications Networks," *Journal of Adaptive Behavior*, 5(2):169–207, 1996.

[26] G. D. Caro and M. Dorigo, "AntNet: Distributed stigmergetic control for communication networks," *Journal of Artificial Intelligence Research*, 9:317– 365, December 1998.

[27] D. Camara and A. Loureiro, "A GPS/ant–like routing algorithm for ad hoc networks," *Proceedings of Conference on Wireless Communications and Networking*, volume 3, pages 1232–1236, Chicago, USA, September 2000.

[28] M. Gunes, U.Sorges, and I. Bouazzi, "ARA – the ant-colony based routing algorithm for MANETs," *Proceedings of the international conference on parallel processing workshops (ICPPW'02)*, pages 79–85, Vancouver, B.C., August 2002.

[29] H. Matsuo and K. Mori, "Accelerated Ants Routing in Dynamic Networks," *Proceedings of Intl. Conf. On Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 333–339, Nagoya Institute of Technology, Japan, August 2001.

[30] D. Subramanian, P. Druschel, and J. Chen, "Ants and reinforcement learning : A case study in routing in dynamic network," *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 832–839, Nagoya, Aichi, Japan, 1997.

[31] T. H. Ahmed, "Simulation of Mobility and Routing in Ad Hoc Networks using Ant Colony Algorithms," *International Conference on Information Technology: Coding and Computing*, volume II, pages 698–703, Las Vegas, Nevada, USA, April 2005. IEEE Computer Society.

[32] B. R. Haverkort, "Performance of Computer Communication Systems: A Model-Based Approach," *Wiley and Sons, Canada*, 1998.

[33] O. Hussein and T. Saadawi, "Ant routing algorithm for mobile ad hoc networks (arama)," *Proceedings of the 22nd IEEE International Performance, Computing, and Communications Conference*, pages 281–290, Phoenix, Arizona, USA, April 2003.

[34] P. P. Grasse, Les Insectes Dans Leur Univers. Paris, France: Ed. Du Palais de la decouverte, 1946.

[35] ZHU Ling, SHI Xinling, LIU Yajie, "Edge Extraction of Thyroid Nodule in Ultrasound Images Based on Improved Ant Colony Algorithm," *Computer Engineering. 2006*, 32(24): 178-179,239.

[36] GU Lingkang, LIN Hongji, "Monitoring System Based on Ant Colony Algorithm," *Journal of System Simulation. 2006*,18:369-373.

[37] YANG Licai, ZHAO Lina, WU Xiaoqing, "Medical image segmentation of fuzzy C-means clustering based on the ant colony algorithm," *Journal Of Shandong University (Engineering Science) 2007*,37(3):51-54.

[38] WANG Lihong, NI Zhiwei, GAO Yazhuo, “An Improved Ant Colony Algorithm for Multi-Objective Job-Shop Scheduling Problem,” *Computer Technology And Development. 2008*,18(10):49-52.

[39] SONG Xiaoyu, WANG Dan, “Research on Hybrid Ant Colony Algorithm for Job Shop Schedule,” *Computer Engineering.2007*,33(4):218-219,222.

[40] Feng Yong, Liao Ruihua, Rao Nini, “The Routing Protocol Based on Improved Ant Colony Algorithm for Ad hoe Networks,” *Journal of Electronics & Information Technology.2008*,30(10):2472-2475.

[41] Liu Yanpeng, Wu Mingguang and Qian Jixin, “A novel distributed multicast routing algorithm based on ant colony algorithm,” *JOURNAL OF CIRCUITS AND SYSTEMS. 2008*,13(5):112-116,144.

[42] Li Dengfei and Zhu Qunxiong, “Hybrid Intelligence Algorithm Based on Ant Colony Swarm and Its Application in TSP,” *JOURNAL OF QINGDAO UNIVERSITY(Natural Science Edition).2007*,20(l):58-62.

[43] Gao Chengshi, Wang Jianzheng and Zhang Dong, “Searching trust path model based on ant colony algorithm,” *Computer Engineering and Applications.2007*,43(15): 131-133.

[44] Bullnheimer et al., “Applying the ant system to the vehicle routing problem.” *2nd Metaheuristic International Conference, 1997.*

[45] Luca Maria Gambardella et al., “MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows. New Ideas in Optimization,” McGraw-Hill, 1999.

[46] G. Di Caro, F. Ducatelle, and L. M. Gambardella, “AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks,” *European Transactions on elecommunications,* vol. 16, no. 5, pp. 443–455, 2005.

[47] M. Guntsch and M. Middendorf, “Pheromone modification strategies for ant algorithms applied to dynamic TSP,” in *Applications of Evo-lutionary Computing: Proceedings of EvoWorkshops 2001*, ser. LNCS, E. J. W. Boers et al., Eds., vol. 2037. Springer Verlag, 2001, pp. 213–222.

[48] C. J. Eyckelhof and M. Snoek, “Ant systems for a dynamic TSP: Ants caught in a traffic jam,” in *Ant Algorithms – Proceedings of ANTS 2002 – Third International Workshop, ser. LNCS,* M. Dorigo et al., Eds., vol. 2463. Springer Verlag, 2002, pp. 88–99.

[49] L. Bianchi, L. M. Gambardella, and M. Dorigo, “An ant colony optimization approach to the probabilistic traveling salesman problem,” in *Proceedings of PPSN-VII, ser. LNCS, J. J. Merelo et al., Eds.*, vol. 2439. Springer Verlag, 2002, pp. 883–892.

[50] J. Branke and M. Guntsch, “New ideas for applying ant colony optimization to the probabilistic TSP,” in *Applications of Evolutionary Computing, Proceedings of EvoWorkshops 2003, ser. LNCS, G. R. Raidl et al., Eds.*, vol. 2611. Springer Verlag, 2003, pp. 165–175.

[51] W. J. Gutjahr, “A converging ACO algorithm for stochastic combinatorial optimization,” in *Proc. SAGA 2003, ser. LNCS, A. Albrecht and T. Steinh ¨ofl, Eds.,* vol. 2827. Berlin, Germany: Springer Verlag, 2003, pp. 10–25.

[52] V. T’kindt, N. Monmarch´e, F. Tercinet, and D. La¨ugt, “ An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem,” *European Journal of Operational Research,* vol. 142, no. 2, pp. 250–257, 2002.

[53] K. Doerner, R. F. Hartl, and M. Reimann, “Competants for problem solving – the case of full truckload transportation,” *Central European Journal for Operations Research and Economics*, vol. 11, no. 2, pp. 115–141, 2003.

[54] E.-G. Talbi, O. Roux, C. Fonlupt, and D. Robillard, “Parallel ant colonies for combinatorial optimization problems,” in *Parallel and Distributed Processing, 11 IPPS/SPDP’99 Workshops, ser. LNCS, J. Rolim et al., Eds.*, vol. 1586, 1999, pp. 239–247.

[55] M. Manfrin, M. Birattari, T. St¨utzle, and M. Dorigo, “Parallel ant colony optimization for the traveling salesman problem,” in *Proceed-ings of ANTS 2006, ser. LNCS, M. Dorigo et al., Eds.,* vol. 4150. Springer Verlag, 2006, pp. 224–234.

[56] K. Socha, “ACO for continuous and mixed-variable optimization,” in *Ant Colony Optimization and Swarm Intelligence, 4th International Workshop, ANTS 2004*, ser. LNCS, M. Dorigo et al., Eds., vol. 3172. Springer Verlag, 2004, pp. 25–36.

[57] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein, “Smell as a computational resource – a lesson we can learn from the ant,” in *Proceedings of the Fourth Israeli Symposium on Theory of Computing and Systems (ISTCS-99)*, 1996, pp. 219–230.

[58] “Efficient graph search by a smell-oriented vertex process,” *Annals of Mathematics and Artificial Intelligence,* vol. 24, pp. 211–223, 1998.

[59] J.-L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chr´etien, “The dynamics of collective sorting: Robot-like ants and ant-like robots,” in *Proceedings of the First International Conference* on Simulation of Adaptive Behavior: From Animals to Animats, J.-A. Meyer and S. W. Wilson, Eds. MIT Press, Cambridge, MA, 1991, pp. 356–363.

[60] J. Handl, J. Knowles, and M. Dorigo, “Ant-based clustering and topographic mapping,” *Artificial Life*, vol. 12, no. 1, pp. 35–62, 2006.

[61] G. E. Robinson, “Regulation of division of labor in insect societies,” *Ann. Rev. Entomol.*, vol. 37, pp. 637–665, 1992.

[62] M. Campos, E. Bonabeau, G. Theraulaz, and J.-L. Deneubourg, “Dynamic scheduling and division of labor in social insects,” *Adaptive Behavior*, vol. 8, no. 3, pp. 83–96, 2000.

[63] A. Martinoli, M. Yamamoto, and F. Mondada, “On the modelling of bio-inspired collective experiments with real robots,” in *Proceedings of the Fourth European Conference on Artificial Life, P. Husbands and I. Harvey, Eds. MIT Press, Cambridge, MA, 1997*.

[64] C. R. Kube and H. Zhang, “Collective robotics: From social insects to robots,” *Adaptive Behavior,* vol. 2, pp. 189–218, 1994.

[65] M. Dorigo, V. Trianni, E. S¸ ahin, R. Grob, T. H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, and L. M. Gambardella, “Evolving self-organizing behaviors for a Swarm-bot,” *Autonomous Robots,* vol. 17, no. 2–3, pp. 223–245, 2004.

[66] WAN-JUN YU, GUO-MING ZUO, QIANQ-QIAN LI, “ANT COLONY OPTIMIZATION FOR ROUTING IN MOBILE AD HOC NETWORKS” *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics,* Kunming, 12-15 July 2008