CHAPTER 3

ANT COLONY OPTIMIZATION: PROGRESSIONS TO ROUTING

With the advancement of Internet and telecommunications, demand of fast, secure and accurate networks has also grown. Due to this, more and more complex network systems are being made. The complex networks are exposed to intricacies related to high cost, load balancing, congestion control, routing, etc. Therefore the challenge is to develop more sophisticated intelligent networks to resolve these problems. There are a number of upcoming intelligent approaches to elucidate such complex systems. Swarm intelligence, Evolutionary Algorithms (EA), Ant Colony Optimization (ACO) and their variants predominate the domain of nature-inspired metaheuristics. Ant Colony Optimization (ACO) is one of the very natural and accordant approaches to network routing. ACO can be considered as an adaption of swarm intelligence. ACO is the most prevalent technique among other swarm intelligence techniques. Swarm intelligence as defined in literature is “Artificial intelligence based on collective behavior of decentralized, self-organized system”. It portrays the collective behavior of insects and other organisms such as ants; honey bees etc. manifesting their states and actions. The interaction of thousands of autonomous swarm members through the environment is based on the principle of “stigmergy” which leads to complex and intelligent behavior exhibited by the insects through the interaction of thousands of autonomous swarm members. Examples in natural systems include ant colonies, bird flocking, animal herding, bacterial growth etc.

The gain in popularity of ants based routing algorithms in research can be contributed to the fact that they are more robust, adaptive, scalable and fault tolerant than other conventional routing algorithms. They are anticipated to perform better than conventional routing algorithms like Link-State and Distance-Vector where exchange of messages (control information) is concerned.

During the last two decades computer scientists have successfully attempted and transformed the models of ant colonies into optimization and other similar applications. Since its inception, ACO has been applied successfully to a number of combinatorial optimization problems such as:
Traveling salesman
Load balancing
Routing in telecommunication networks
Industrial applications
Machine learning
Bioinformatics
Quadratic Assignment problem: Assign $n$ activities to $n$ locations (campus and mall layout) etc.

This chapter surveys the customization of ACO in network routing. The application of ACO in network routing is the simulation of ants travelling, pheromone layering and maintaining data structures while travelling from source to destination

### 3.1 ANT COLONY OPTIMIZATION

ACO deals with the way ants find their food traversing shortest paths. Foraging behavior of ants lead them to discover optimum paths (in terms of distance) to reach their food. For doing this, they wander randomly to search food, on the traversed path laying a chemical substance called pheromone, and on finding their food they return back on the same path to their nest (as a consequence of stigmergy) reinforcing more pheromone which facilitate other ants to follow the same path. With the subsequent movement of other ants along the same path, the path gets further reinforced with pheromone and thus starts becoming the best (shortest path) for the ants. Pheromones evaporate faster on longer paths thus making those paths inconspicuous. Finally, the ants turn up at the shortest path. The procedure is depicted in figure 3.1 which is a very customary figure found in literature depicting the principle of ACO.
ACO procedure can thus be summarized as:

- An ant (agent) marks its return journey with pheromones (marks) after it finds its food. Marks serve as a shared memory for the agents. By inducing these marks, an agent provides knowledge of concern to other agents.
- If path traversed are longer, pheromones will evaporate faster (knowledge become outdated with time)
- Successive ants follow path on pheromone tracing that is existent on shorter paths. i.e. other agents perceive these marks to influence their behavior. Simultaneously, they may manipulate the marks to keep the knowledge updated.
- The shorter path will be reinforced further by the pheromones laid down by successive ants.
- Thus automatically shorter paths are identified by ants for travelling

Following characteristics of swarm intelligence make it suitable for routing in networks particularly when multiple paths are to be identified. Problem formulation in this thesis is governed by these characteristics.
Collective behavior forming a synchronized system which is capable of accomplishing difficult tasks in dynamic and varied environments. Further, the tasks are performed without any external guidance or control and without any central coordination.

Attaining an optimistic performance gained by collective action which ordinarily could not be achieved by individual efforts.

A natural model inspired by intelligent behavior of biological organisms befitting to distributed problem solving.

These characteristic gave the following directions for the work carried out:

- Design of a distributed and dynamic routing strategy.
- No centralized control.
- Use of software agents, referred as ‘Pert Ants’ to explore multiple apropos links leading to the identification of multiple lucrative paths between source-destination pairs.
- Efficient adaptation to dynamism and variations in traffic status and topologies while exploring the network.

Real Ant Vs Artificial Ant

The deviation between the characteristics of the real ants and the artificial ants as modeled for various experiments is:

- Real ants move asynchronously in the environment. Artificial ants move in a synchronized way, that, is in every iteration artificial ants move from source node to destination node and while coming back they follow the same path.
- Real ants while moving, deposit pheromone on the ground. Artificial ants imitate the behavior by depositing artificial pheromone in terms of numeric value.
- The foraging behavior of real ants is based on the fact that the ants moving on a shorter path will complete the path first as compared to longer paths, so shorter paths will have higher value of pheromone. Artificial ants evaluate a path according to some metrics. Depending on metrics, artificial ants calculate the value of pheromone that is to be deposited by ants during their return path.
Although most of the ACO approaches use the basic problem solving techniques of biological ants, but some variations are done according to the requirement of the problem and solution. When using artificial ants, some of the biological traits of the ants may not be used and some other traits may be added to have a better solution using heuristic approaches.

### 3.2 ACO CHALLENGES IN NETWORK ROUTING

As the Internet is growing at an exponential speed, so network routing has also become NP-hard problem which cannot be solved in polynomial time. A number of network routing algorithms based on ACO approaches have been proposed in literature. Distinct algorithms have been summarized in this chapter along with their capabilities and limitations.

While working with ACO in network routing, the major challenges to be considered are:

- How the routing information will be handled?
- What will be the routing overhead?
- How the issues of adaptation and stagnation will be handled?

#### 3.2.1 CONVENTIONAL ROUTING VS ACO

**Routing Process**

In traditional routing algorithms such as RIP or OSPF, nodes in the network depend upon the routing information provided by all the immediate neighbors of the node. This process is regulated by all the nodes which lead to the formation of complete routing table. Routing table in RIP is dependent on the distances between the nodes, while in OSPF; the routing table is formulated using Link-State information circulated by all the nodes in a network. The Link-State information specifies the neighbors of a node along with the delays on the links to those neighboring nodes. This information is then used by all the nodes to create labeled graph of the network on which Dijkstra’s algorithm is applied to obtain the shortest paths.

In ACO, the paths are explored in parallel and independently. As an ant arrives at a node, it updates the pheromone value corresponding to the source node in the path. Therefore all the
entries of a pheromone table at a node can be modified independently and in this manner the complete pheromone table/routing table is formed independently.

**Routing Overhead**

In traditional routing, the transmission of routing vector is done by every node to its every neighbor in case of RIP. In case of OSPF, a Link-State-Packet is transmitted by all the nodes to all the other nodes in the network using flooding. The overhead can be very large in RIP as complete routing vector needs to be transmitted to every neighbor. In case of OSPF, though the size of LSP is much smaller, yet due to flooding, duplicate copies of same LSP may be transmitted by various nodes via different paths, which results in increase of superfluous packets leading to overheads and congestion.

Routing in ACO is achieved by transmitting ants, which generally have a very small size as compared to LSP or a complete routing table. Multiplicity of ants is taken care of in nearly all the ACO based routing schemes. Therefore the overhead in ACO is generally much lesser than traditional routing.

**Handling Dynamism**

In case of dynamic networks, the changes to be made in LSP or large routing tables may incur large overheads for routing and may lead to slower speed of the network. In ACO, the changes to be made for dynamic network may be handled comparatively easily by piggybacking ants in data packets with more frequent transmission as the ant’s size is comparatively smaller.

**Stagnation Problem**

However, ACO approach also suffers from the problem of stagnation. Stagnation is the stage when all the ants choose only one path. As ants confine themselves to a single path, they will deposit more and more pheromone on that path and the path will become optimum very soon. This may lead to congestions and therefore the network might come to a standstill. Another problem is that of local optima. If most of the ants start following a non optimum path, then that
Chapter 3: Ant Colony Optimization: Progressions to Routing

path will be considered as optimum path. The actual optimum paths may not be exploited by the ants.

This problem is not encountered in conventional routing algorithms.

3.3 VARIOUS ACO ALGORITHMS IN ROUTING

The first definition of the ACO metaheuristic was given in 1999 by Dorigo, Marco et al. [Dor99].

Before the origin of ACO, AntNet and ABC algorithms were introduced in literature. These algorithms along with some other algorithms were used in various applications. This section initially provides an overview of the two main approaches in applying ACO in network routing, i.e. AntNet, and ABC.

3.3.1 ANTNET [Di 98], [Di 97]

Di Caro A. and M. Dorigo designed AntNet algorithm for Asymmetric Packet Switched networks for optimizing the performance of overall network instead of focusing on shortest or minimal path. It is an algorithm for adaptive best-effort routing in IP networks also. Design of AntNet algorithms is based on Ant Colony optimization (ACO) framework. AntNet uses two types of ants (exploration agents), which are forward (FRANTs) and backward ants (BKANTs). FRANTs are launched at a regular time intervals from source node to a random destination. FRANTs use normal queues to experience the true network conditions. FRANTs collect information about trip time to each node, nodes traversed, and traffic existing at each node. The FRANTs travel in an asynchronous and concurrent manner with the data traffic.

As a FRANT reaches its destination, it copies its structure into BKANT and dies. BKANT then travels in backward direction towards the source node via exactly same path as travelled by FRANT but in reverse order. The reason for using two types of ant agents in AntNet is that the FRANTs are basically used for collecting information such as the node ids in the path and trip times. They are not used for any updates in the routing table at the nodes. The BKANTs receive
this information from the FRANTs and accordingly updates the routing table at the nodes on their way back. BKANTS use priority queues to quickly circulate the information to the nodes.

The nodes in the network maintain various data structures such as $M_k$, a statistical model containing information about end to end delays, $T_k$, a pheromone table, and the routing table to be updated by BKANTS. The updation in the pheromone and routing tables are done depending on the quality of the path identified by the FRANTS.

The data structure $M_k$, which is a statistical model, is actually a vector for storing the mean and variances calculated on the basis of the delays measured by the ant agents and to have a good estimation of the best trip time. The routing table stored at each node includes entries for each neighbor as well as each destination of the node. For example, if a network consists of $n$ nodes and a specific node is having $m$ neighbors, then the routing table at that node will have $n-1$ rows and $m$ columns. The values stored in the table corresponding to each neighbor ($n$) and each destination ($d$) are the probability ($P_{dn}$) of going through this link for the destination $d$. The rows of the table represent the probabilities of reaching to the destination via various neighbors.

The entries of the routing table are probabilities, and must sum to 1 for each row. Dual purpose is served by these probabilities:

(i) The exploration agents of the network use them to decide the next hop to a destination, which is a random selection among all candidates based on the routing table probabilities for a specific destination.

(ii) The data packets deterministically select the path with the highest probability for the next hop.

The sequential procedure followed in AntNet can be specified as:

i. In regular time intervals, forward ants are launched by each network node to all destinations.

ii. Based on the current routing table entries, the ants explore their path randomly towards the destination.
iii. While moving from node to node stack is created and maintained by the forward ants in which it pushes the identity of the current node being traversed and the trip time experienced on reaching that node.

iv. When the forward ant reaches the destination, the backward ant inherits the stack.

v. The backward ant follows the reverse path from the destination to the source by popping the entries from the stack and hopping to the next node according to that entry.

vi. The trip times associated with a node in that popped entry is used by the backward ant to update the node tables.

Probability function for an ant to choose an edge is using a combination of both pheromone concentration and heuristic function.

\[ P'_{nd} = \frac{\tau_{nd} + \alpha \cdot l_n}{1 + \alpha \cdot (|N_k| - 1)} \]

Where \( \tau_{nd} \) is the pheromone concentration, \( l_n \) is the normalized heuristic function. The values of \( l_n \in [0,1] \) proportional to the status of the local link queues of the link connecting the node \( k \) to its neighbor \( n \).

The value of \( \alpha \in [0, 1] \) weighs the relative importance of the heuristic correction with respect to the pheromone values stored in the pheromone matrix.

Parametric Statistical Model \( M_k \) is a vector of \( (n-1) \) data structures stored at each node storing sample mean \( \mu_d \), variance of the travelling time \( \sigma_d^2 \) to reach destination \( d \) from the current node, while \( W_d \) is the best travelling time to \( d \) over the window of the last \( w \) observations with reference to destination \( d \).

\[ M_k(\mu_d, \sigma_d^2, W_d) \quad r \in (0,1] \]

Reinforcement is a function of the goodness where

\[ r \equiv r(T_k, M_k) \]
The new FRANTs are influenced by the updation done by BKANT in the routing table, which in turn is dependent on the path goodness as calculated by earlier FRANTs. When BKANT updates the routing table, it increases the probability $P_{df}$ and decreases the probabilities $P_{dn}$ corresponding to all other neighbours. This increase or decrease in the probabilities is done on the basis of a reinforcement value $r$, which is calculated for the new trip time on the basis of current routing table and local traffic statistics. The value of $r$ lies between 0 and 1 and tells about the goodness of the newly experienced delay. The pheromone table is updated by increasing the probability $P_{df}$ and decreasing the probabilities $P_{dn}$. The values are updated in a manner so that sum of all the probabilities still sum to 1.

$$P_{df} = P_{df} + r \times (1 - P_{df}) \ldots \ldots \ldots \ldots \ldots (3.1)$$

$$P_{dn} = P_{dn} - rP_{dn} \ldots \ldots \ldots \ldots \ldots (3.2)$$

However, by using both FRANT and BKANT, the routing overhead in AntNet Algorithm is increased.

### 3.3.2 ANT BASED CONTROL [Sch97]

Ant Based Control (ABC) algorithm was the first algorithm inspired by the behavior of ant colonies in network routing. The approach was designed by Schoonderwoerd et al. for routing in telephone networks for load balancing. The principle on which this technique works comprises of mobile routing agents called ants. These agents traverse through the network and explore it randomly. The routing tables are updated on the basis of current state of the network. Each node in the network comprises of capacity $C$ which shows the number of calls that can be accommodated, probability of being a destination and probabilistic routing table.

ABC uses only one type of ant i.e. the forward ant. These ants are generated from every node to a randomly chosen destination at regular intervals. When travelling from one node to another, they select the next node according to the values in the pheromone table. Upon arriving at a node, the forward ant updates the routing table entries immediately against their source node. In other words the pheromone value corresponding to its previous node is incremented.
It is important to note that ants moving from source node can actually influence only those ants travelling in reverse direction or for which the source node is the destination. In simple words, it can be said that the ants travelling from source $S$ to destination $D$ may influence other ants that are travelling from any node in the network towards $S$ and ants travelling from source $S$ to destination $D$ may get influenced by other ants that are travelling from $D$ to any other node in the network.

The entries corresponding to the node from which the ant has arrived are updated according to the following in equation (3.3).

$$P = \frac{(P_{old} + \Delta P)}{(1+\Delta P)} \quad \ldots \ldots \ldots (3.3)$$

where $P$ is the updated probability according to the probability increment $\Delta P$. Other entries corresponding to the node are decremented according to equation (3.4).

$$P = \frac{(P_{old})}{(1+\Delta P)} \quad \ldots \ldots \ldots (3.4)$$

For identifying shorter paths, avoiding visiting heavily congested nodes, and avoiding stagnation, three methods have been introduced, which include aging, delaying of ants and noise.

Shorter paths can be identified by reducing the value of $\Delta P$ gradually depending on the age of the ant. Age of the ant depends on the length of the path it travels. Longer the path more will be the age of the ant and therefore the ants which are travelling via shorter paths will have more impact on the routing table updates as compared to the ants which travels through longer paths.

The equation (3.5) below indicates a proposal for calculation of $\Delta P$.

$$\Delta P = \frac{((d/age) + c)}{(1+\Delta P)} \quad \ldots \ldots \ldots (3.5)$$

where $c$ and $d$ are constants.

For avoiding the ants to travel through congested nodes, a delay may be added to the ants so that lesser number of ants will be travelling to their neighbors via this path. This delay will prevent other ants to travel via this path and the new ants will be able to find alternate paths avoiding
congestion. The delay will also increase the age of ants again resulting in a lesser pheromone deposit and therefore lesser probability of selecting the congested path.

Stagnation can be avoided by adding noise. The basic purpose of adding noise is to have diversity in finding the paths. As in case of ants it is not guaranteed that always the shortest path will be found. It may be the case that if the initial ants select a longer path, the other ants may also follow that path. As more and more pheromone is deposited on that path, that path may yield the final path, which obviously is incorrect.

Adding noise will enable the ants to select a path in a random fashion and without considering the influence of the pheromone table. By adding noise, more diverse paths can be found, therefore even if a bad path has been selected by earlier ants, the better paths may still be identified by the newer ants and in this manner stagnation can be avoided.

Both ABC and AntNet algorithms have proved to be very useful in the field of network routing. A number of ramifications have been proposed by various authors in literature in these algorithms. Before discussing these ramifications, the comparison of both the approaches is essential.

### 3.3.3 DIFFERENCES AND SIMILARITIES IN ABC AND ANTNET APPROACHES

#### Differences

- AntNet is used to improve the overall performance of the network instead of focusing on shortest path whereas ABC is used for load balancing in telecommunication network.
- In ABC ants update pheromone trails on node by node basis, while in AntNet, the updation is carried out by backward ants when going back to the source node. Therefore in ABC approach, ants do not require to travel in reverse direction and go back to their source node.
- In AntNet, selected set of ants (namely backward ants) have the privilege to deposit more or less pheromone depending on the problem, which can prove to be very useful for different types of applications. But the use of backward ants increases the network traffic.
in return trip. As ABC does not use backward ants, network traffic is less in this approach.

- In AntNet, local traffic models are used to score the ant traveling time, while ABC does not use such types of models and therefore is faster than AntNet.
- AntNet performs learning from local queue information and ant’s own memory to improve the decision taken by the ant and balance the pheromone updates accordingly. ABC does not use any such concept.
- In ABC approach cycles can be formed as it does not use the information contained in its sub-paths, while AntNet uses such type of information and therefore cycles are not formed.

**Similarities**

- Both approaches do not put any limit on the amount of pheromone they deposit on each link/path.
- There is no limit on the number of ants to be used.
- Both techniques use uniform distribution of probability in the initial stage which does not reflect the state of network.

### 3.4 EXTENSION OF ANTNET AND ABC ALGORITHMS

This section discusses some of the ramifications of the ABC and AntNet approaches.

#### 3.4.1 ALGORITHMS FOR WIRED CONNECTIONLESS NETWORKS

In this section the work related to the application of AntNet, ABC, and, other ant colony ideas, to routing in wired connectionless networks such as the Internet has been discussed.

Subramanian, Devika et al. considered general cost-asymmetric networks and provided analysis of two different algorithms, viz. ABC and uniform ants. Though routing table is updated in the reverse direction of their motion in both algorithms, yet the difference between them lies in the fact that uniform ants move in the network blindly, without relying on pheromone. The major advantage of using uniform ants is that using simple unbiased exploration adaption to any change
in the network can be easily identified, especially failures. Uniform ants are able to find all multipaths with equal probabilities. The authors show that both the approaches exhibit similar results for small networks and are similar to simple link-state and distance-vector algorithms. But when the network size increases, the efficiency of uniform ant decreases very rapidly. The crux of this research is that ACO proves to be a better technique to find multipath instead of uniform ant algorithm [Sub97].

Van der Put, Roland proposed an algorithm called ABC-backward which is a combination of basic ABC and forward-backward updating strategy of AntNet. It has been proved in this paper that backward ant mechanism provides better result than basic ABC scheme [Van98] [Van99].

Heusse, Martin et al. proposed a cooperative asymmetric forward (CAF) technique for routing in packet-switching networks. CAF works similar to ABC except that a CAF ant updates routing table using the cost in reverse direction as stored by the data packets. CAF ants can be used for both symmetric and asymmetric networks [Heu98].

This approach may not be very useful if data packets do not travel on both sides on a path. Another problem with this approach was that it needs to maintain a reverse routing table.

Oida, Kazumasa, and Akira Kataoka proposed DCY-AntNet, NFB-Ants algorithms. The algorithms improved an earlier version of AntNet given by Di Caro A. and M. Dorigo. In the earlier AntNet, a problem of stagnation was a major issue. The use of an evaporation mechanism by the authors of this paper will allow exploration at any time. The authors modified pheromone table updating rules to avoid the locking behavior. Their algorithms, DCY-AntNet and NFB-Ants, proved to be much better than AntNet under such condition where new paths needed to be identified under conditions such as congestion, network failure, etc. [Oid01].

Doi, S. and Yamamura.M, proposed BNetL approach to solve stagnation problem addressed by Oida, Kazumasa, and Akira Kataoka, but by using AntNet-FA, which is actually lock-free. Their algorithm showed at par performance with AntNet-FA [Yam02] [Doi02].

Baran, Benjamin, and Ruben Sosa have proposed a number of modifications to AntNet-FA. In the algorithm they have proposed (i) more pheromone deposit on shorter paths instead of
initializing starting pheromone using equal probability (ii) The edges or link which are prone to failures are initialized to zero pheromone values (iii) uniform ant is used by using random decision policy similar to the policy given by Subramanian, Devika et al. to avoid the stagnation effect (iv) in the proposed approach regular ants use greedy deterministic decisions but the disadvantage is that it reduces the exploration and increases the chances of trapping of ants and data in loops. (v) To reduce routing overhead, the ants used are only four times the number of edges, but this can downturn the receptivity of the algorithm and it may not be manageable in a distributed way [Bar00].

Yang, Y., et al. studied the implementation of AntNet on a real network i.e. on 5-node LAN of Windows-based machines using the TCP/IP protocol. The algorithm was implemented on application layer not on network layer so that time for implementation can be shortened. They recognized that with constant reinforcements performance of the system gets low and with adaptive reinforcements better performance may be achieved [Yan02].

Doi, Shigeo, and Masayuki Yamamura proposed a loop-free AntNet routing algorithm. Two algorithms namely original AntNet-FA and the loop-free AntNet algorithms have been approved on different network topologies such as hierarchical, scale-free, Internet-like topologies. The paper analyzed that different topologies have a momentous impact on the performance of the two algorithms [Doi04].

Liang, Suiliong et al. have studied and compared the performance of AntNet and GA-agent based on a distributed genetic algorithm architecture, in a dynamic scenario having 56-node network. On this network AntNet proved to be the best routing algorithm. The performance of GA –agent algorithm which does not require prior global information is found to be moderate as compared to AntNet algorithm with and without global information [Lia02], [Lia06].

Verstraete, Vincent, et al. implemented AntNet on a real physical network of 5 routers and 2 hosts. The authors modified and extended the AntNet algorithm so that it may work properly on a real physical network. AntNet’s performance was compared with OSPF on the basis of throughput and failures. AntNet was found to be much better than OSPF in terms of throughput
in all the testing environments. But AntNet recovers to failures slower than OSPF as it does not have any built-in failure recovery mechanism [Ver06].

Dhillon, Santpal Singh, and Piet Van Mieghem reviewed the characteristics of the AntNet algorithm. The authors have compared the AntNet with Dijkstra’s centralized shortest-path algorithm. It has been identified by the authors that AntNet performance is at par with Dijkstra. But when traffic load is variable adaptation of AntNet is better as compared to Dijkstra’s [Dhi07].

### 3.4.2 ALGORITHMS FOR WIRED CONNECTION-ORIENTED NETWORKS

The section reviews the relevant work concerning to the application of ACO ideas to wired connection-oriented networks such as telephone networks and IP networks using virtual circuits is taken into consideration.

Di Caro A. and M. Dorigo proposed a novel method AntNet-FairShare (AntNet-FS) for optimum routing and flow control in virtual circuit networks. In this method for every flow of traffic a virtual circuit is created and bandwidth is also reserved for each circuit. The bandwidth is the maximum bandwidth that can be given to each circuit according to optimum distribution among the users. In AntNet-FS, the usual proactive ant generation is done on the basis of demand. Then whenever a new demand comes, a new forward ant is generated so that is able to find one or more path for the session. In this process the forward ant travels like a normal ant, except when more than one equally good alternative exist at a node, the ant is replicated to all those paths and sent to those paths. The first ant reaching at the destination then goes back to the source and again creates a virtual circuit. Further ants reaching at destination goes back only if their trip time is comparable to that of the first ant and then creates a virtual circuit. Another thing to be considered is that the path should be sufficiently disjoint from those of the circuits allocated so far and the bandwidth should not exceed the reserved bandwidth for that session. The bandwidth is dynamically allocated whenever each session is created and/or destroyed and the data sets are adjusted accordingly [Di 98].

Tony, White et al., proposed various schemes for routing on the basis of ACO, swarm intelligence and genetic algorithms. Their proposal is very similar to the AntNet-FS, but it makes
use of pheromone updating formulas which are derived from Ant System given by Dorigo, Marco et al. [Dor96]. In particular, they used the concept of pheromone evaporation to sustain path exploration. They considered both scenarios and conducted experiments on small networks. The proposed algorithm is able to find shortest path for given conditions. The performance of ASGA routing algorithm has been increased with the adoption of the genetic algorithm [Ton98], [Whi98], [Whi99].

Di Caro et al. proposed a variant of AntNet based on priority queues depending on which forward ants travel. The backward ants estimates the trip times and perform other tasks such as updating the local traffic statistics and deposit the pheromone (probability) accordingly. The routing information in this technique is more accurate as compared to AntNet. The results found using this approach are better as compared to the AntNet [Di04].

Bonabeau E. et al. extended the ABC approach with the idea of using smart ants. In this work, ants update the probability values for all visited nodes. Pheromone update is done for source node as well as for intermediate nodes by the smart ants. It has been identified that although smart ants are more complex yet better performance is achieved in case of smart ants as compared to ABC ants. Also total number of ants required is less as compared to ABC ants and number of dropped calls is also less [Bon98].

Sandalidis, Harilaos G. et al. studied the ABC and found that the results of ABC are quite good for routing. Then the same authors tried to improve ABC using the concept of anti-pheromone quite similar to the repulsive pheromone used by ants in nature to block unfavorable paths. They have found that a sampled route using anti-pheromone is not good compared to other available routes. In most of the other ACO implementations, after being sampled, the selection probability of a route is always increased. The performance of the algorithm has been compared to that of ABC and has shown a slightly better performance [San01] [San04].

Multiple ant colony approach (MACO) was proposed firstly by Verela and Sinclair. The proposed approach was based on problems found in virtual wave length path routing and wavelength allocations. The proposed strategy of wavelength path routing is to allocate the least wavelengths for each link by distributing the wavelength requirements evenly over all the links.
Chapter 3: Ant Colony Optimization: Progressions to Routing

The concept of pheromone attraction and repulsion is used in this type of routing. The pheromone attraction works in a similar manner as in case of ACO, but the pheromone repulsion is used to repel the different wavelength so that they can be distributed over different links. A probability function similar to the ACO applications is generated; however this function is created using the different degrees of pheromone attraction and repulsion. In this technique the ants of different colonies have been represented by different colours. The ants travel in the same manner as in case of AntNet. The ants check the pheromone deposit by the ants of same colony. If the pheromone deposit of the same colony is high, then the ants are attracted to that path. However, the pheromone deposit of the ants of different colony is high, the ants are repelled. Obviously, after sometime ants of different colours will choose their own path. In this manner multiple optimum paths can be identified [var99].

Sim, Kwang Mong, and Weng Hong Sun have proposed the Multiple Ant Colony Approach (MACO) based on multiple colonies for load-balancing in connection-oriented networks. Each colony is represented with different colour and lays different type of pheromone. An ant from a particular colony selects paths having high values of pheromone that is laid by ants of the same colony from which ants are generated and get fend off by paths having high concentration of pheromone laid by ants of different colony. It has been discovered by Sim and Sun that by applying MACO approach, stagnation reduces and adaptivity increases. It increases the probability of selecting new and better paths. It was proposed to achieve load balancing in circuit-switched networks [Sim03].

3.4.3 ALGORITHMS FOR NETWORKS PROVIDING QUALITY-OF-SERVICE

In this section literature related to application of AntNet for QoS in wired networks has been reviewed.

Oida, Kazumasa, and Masatoshi Sekido proposed an Ant based routing system which is an improvement of AntNet so that it can support best effort and QoS routing with reservations of resources and admission control as well. A Weighted Fair Queueing algorithm is used which distributes the capacity between best-effort and QoS traffic at the nodes. The QoS measures studied are bandwidth and hop count. The hop count should be less than a pre-specified value
and bandwidth should be among a pre-defined level. An ant is generated from the source using AntNet scheme under given constraints. Links having more bandwidth are preferred for choosing the next hop. After finding a feasible path, it is proclaimed back to the source node. Source node stores the information in an up-to-date table. If a path is available within constraints, an ant is sent to explore it and reserve the necessary resources. If the path does not exist, then session is expired [Oid99], [Oid00].

Michalareas, T., and L. Sacks combined the characteristics of AntNet and ABC algorithms to delineate a new algorithm using multiple swarms for routing in multi-constrained QoS environment. The algorithm uses end-to-end delay and bandwidth as constraints. For each constraint a different swarm of ants is used which adopts a multi colony approach. Ants which deal with delay work according to AntNet approach and others dealing with bandwidth a resource monitor have been introduced to calculate the average spare bandwidth available at the links. In this paper, the bandwidth estimate is reduced to a delay estimate. Simulation experiments show that Multi-Swarm has performance comparable to OSPF [Mic01].

3.5 OBSERVATIONS AND MOTIVATION:

Above delineated literature review of application of ACO approach (AntNet, ABC and other ant colony ideas) to routing in wired connectionless and connection oriented networks and QoS based routing led to the following elucidation which helped in the inception of problem formulation for the work carried out in the thesis.

<table>
<thead>
<tr>
<th>Cognition</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Van98] [Van99] reveals that capabilities of ABC and AntNet combined may yield better performance.</td>
<td>Provides encouragement to design a multi path strategy that exploits the routing proficiency of both ABC and AntNet.</td>
</tr>
<tr>
<td>[Sub97] Unbiased exploration of network (move in network blindly without relying on pheromone) facilitates failure identification as well as identification of all multi paths between</td>
<td>An ant based strategy which is flexible and adaptive to network traffic conditions in the sense that under low load conditions it explores all or maximum paths exploring all functional</td>
</tr>
<tr>
<td>Cognition</td>
<td>Motivation</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>source-destination pairs in a network. On the other hand, it also increases routing overheads. This is indicative of the fact that for multipath identification, a strategy which adapts to both unbiased as well as selective multipath identification procedure under varying status and requirements of traffic and its flows will result in enhanced performance of the network.</td>
<td>links and under high load conditions, becomes selective according to optimum paths may be a reasonable solution in the direction of improving performance of a network when multi path considerations are made.</td>
</tr>
<tr>
<td>Pheromone evaporation to control stagnation is another evinced aspect in [Oid01]. Most of the variants of ACO implementations have accommodated this aspect using age field in ant structure. It is also observed that age is perceived in terms of hop count in many proposed strategies.</td>
<td>A rationale that emerged as a consequence is to adopt a mechanism to control stagnation. This motivated to develop a strategy to regulate the amount of pheromone deposited on paths in such a way so as to avoid stagnation which implements aging inherently in path exploration procedure with consequential updations rather than associating it in ant structure. Old value of Link utilization is updated using current load and bandwidth status. Further, hop count can be used as an indicator for path selection criteria rather than associating it with age.</td>
</tr>
<tr>
<td>Ants using greedy deterministic decisions instead of random proportional ones, reduces exploration but increases the chances of loops [Bar00].</td>
<td>This inference motivates to develop a greedy deterministic exploration approach with avoidance of loops.</td>
</tr>
<tr>
<td>Adaptive reinforcement gives better performance. Constant reinforcement leads to slow but dependable performance [Yan02].</td>
<td>Adopt constant reinforcement adaptive parameters, providing simple and fast computation to give dependable performance.</td>
</tr>
</tbody>
</table>
### Cognition

<table>
<thead>
<tr>
<th>Cognition</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Dio04] tested loop free AntNet on varying topologies and identified that topological characteristic have a significant impact on relative performance of algorithm</td>
<td>Functionality of strategy should be independent of topological characteristics yet giving stable performance.</td>
</tr>
<tr>
<td>[Ver06] Compared AntNet with OSPF on real network on the basis of throughput and failures and AntNet was found to be much better than OSPF in terms of throughput, but recovery from failures was slower in AntNet as compared to OSPF. However, insertion of simple mechanism in AntNet overcame this problem. It is pointed out here that original AntNet was developed for single path identification and OSPF as employed in the Internet also works on single path strategy.</td>
<td>ACO based strategy with multiple path identification can overcome the problem of recovery from failures being more robust.</td>
</tr>
<tr>
<td>With variable traffic load, adaption of AntNet is better as compared to Dijkstra’s centralized shortest path algorithm [Dhi07]</td>
<td>A multipath variant of AntNet will give better performance under variable traffic load.</td>
</tr>
<tr>
<td>Concept of anti-Pheromone to block unfavorable paths is used in [San01], [San04]</td>
<td>Ant based strategy should proceed in a way where unfavorable paths are in evitable.</td>
</tr>
</tbody>
</table>

On the basis of above literature review and inferences drawn from the cognition of work carried out by different researchers, it is identified that ACO has been proved to be a very effective technique for finding paths in a computer networks. In this thesis, an attempt has been made to exploit the capabilities of Ant based approaches to routing.

### 3.6 SUMMARY

Traditional routing may have a number of shortcomings such as network failure, slow response time, large overhead, congestion problem and high cost. ACO has been identified to have an edge over traditional routing w.r.t. these shortcomings. ACO is a technique based on real ants
replicating their work in a network. Under ACO, a number of approaches have been proposed, but the most suggestive researches were in the direction of AntNet and ABC approach. Survey and comparison of ABC and AntNet and their extensions in applying ACO in routing has been outlines in this chapter. It may be concluded that a hybrid approach using the efficiencies of both ABC and AntNet can yield better results in the direction of routing optimization. The survey has been categorized under review of ACO based approaches for wired connectionless as well as wired connection-oriented networks. A brief review of literature related to application of AntNet for QoS based routing is also outlined. The chapter concludes with the cognition of the reviews and the motivations derived from them.