# Routing in Dynamic Network using Ants and Genetic Algorithm

**Cauvery N K+ and Dr K V Viswanatha***
R V College of Engineering, Bangalore, India.

**Summary**
Routing in dynamic network is a challenging one, because the topology of the network is not fixed. This issue is addressed in this presentation using ant algorithm to explore the network using intelligent packets. The paths generated by ants are given as input to genetic algorithm. The genetic algorithm finds the set of optimal routes. The importance of using ant algorithm is to reduce the size of routing table. The significance of genetic algorithm is based on the principle evolution of routes rather than storing the precomputed routes.
**Key words :**
*Routing, ant algorithm, genetic algorithm, crossover, mutation. Each of this is addressed in the following section.*

## 1. Introduction

Routing is a process of transferring packets from source node to destination node with minimum cost. Hence routing algorithm has to acquire, organize and distribute information about network states. It should generate feasible routes between nodes and send traffic along the selected path and also achieve high performance. Routing in conjunction with congestion control and admission control defines the performance of the network [1]. Routing algorithm should have generic objective of routing strategy to be both dynamically reconfigurable and be based on locally available information. It should also satisfy user quality of service objectives. Some of the methods proposed in achieving these objectives are social insect metaphors and cognitive packet network. These two methods use the probabilistic routing table and allow the packets themselves to investigate and report network topology and performance. Dorigo M & Di Caro G [2], presents Antnet as an approach for routing in communication network. R Schoonderwoerd, Owen Holland, Janet Bruten and Leon Rothkrantz [3], in their paper describe achieving load balancing in telecommunication networks using ant algorithm. Tony White [4], in his technical report describes how the biologically inspired agents can be used to solve control and management problems in telecommunications.
The purpose of this work is to generate solution using ant algorithm (social inset metaphor) and optimize solution using genetic algorithms. Ant algorithm is a class of swarm intelligence. Swarm intelligence offers an alternative way of designing intelligent system, in which autonomy, emergence and distributed functioning replace control, preprogramming and centralization [5]. This approach emphasizes on distributed- ness, flexibility, robustness and direct or indirect communication among relatively simple agents. Genetic algorithm is one in which the population associated with each node co-evolve to solve the problem as whole.

## 2. Routing

Routing is a process of finding paths between nodes. There are mainly two types of routing policies viz., static routing and dynamic routing. In static routing, the routes between the nodes are precomputed based on certain factors and are stored in routing table [1]. All packets between any two nodes follow the same path. If topology of the network changes, then the path between two nodes may also change, and hence in dynamic routing policy, the routes are not stored but are generated when required. The new routes are generated based on the factors like traffic, link utilization etc which is aimed at having maximum performance.
Routing policy may be centralized or distributed. In the case of centralized routing, only centralized node, generates routes between any pair of nodes. In distributed routing, each node generates routes independently between pair of nodes as and when required. Other classification of routing policy is optimal routing (global routing) and shortest path routing (local routing). Some of the shortest path algorithms are distance vector algorithm and link state algorithm. Characteristics of routing policy are distributed-ness, stochastic and time varying, multiobjectve and multiconstraint.
During the process of route generation, two types of packets are used viz., routing packets (control packets) and data packets. Routing packets are given highest priority when compared to data packets. Each node in the network is of the type store and forward. The link performance may be measured in terms of bandwidth or link delay. The topology of the network may change due to growth in number of nodes, reconfiguration or failure of node. This change in topology should be reflected in the routing table, which in turn helps the routing protocol to generate optimal route for the current state of network. Some of the protocols are Resource Information Protocol (RIP), Interior gateway routing protocol (IGRP), Open source shortest path first (OSPF) and Border gateway

Manuscript received March 5, 2009
Manuscript revised March 20, 2009

protocol (BGP). RIP is a distance vector routing protocol, with hop count as metric for path selection. The change in network topology is reflected by broadcasting route updates. IGRP is also distance vector routing protocol for autonomous system. It supports multiple metrics for each route like bandwidth, delay, load and MTU (Maximum Transfer Unit). It also reflects the change in network by broadcasting route updates. OSPF is a link state routing protocol which uses shortest path first algorithm to compute low cost route to destination. Enhanced Interior Gateway Routing Protocol (EIGRP) is an enhanced distance vector routing protocol with optimization to minimize the effect of change in topology and efficient use of bandwidth and processing power at the router. It uses unequal cost load balancing.

Routing process uses a data structure called routing table at each node to store all the nodes which are at one hop distance from it (neighbor node). It also stores the other nodes (hop count more than one) along with the number of hops to reach that node, followed by the neighbor node through which it can be reached. Router decides which neighbor to choose from routing table to reach specific destination.

## 3. Ant algorithm

Ant algorithm uses the concept of Swarm Intelligence (SI). It is the local interaction of many simple agents to achieve a global goal. SI is based on social insect metaphor for solving different types of problems. Insects like ants, bees and termites live in colonies. Every single insect in a social insect colony seems to have its own agenda. The integration of all individual activities does not have any supervisor. In a social insect colony, a worker usually does not perform all tasks, but rather specializes in a set of tasks. SI is emerged with collective intelligence of groups of simple agents [2]. Ant algorithm along with routing table and data packets, use an intelligent packet called ants to find path between nodes in the network. Ant packets use the concept of stigmergy (Indirect communication) to explore network [6]. As ants move form one node to another, they deposit a chemical called pheromone. Other ants which follow the same path increase the concentration of pheromone. If there exists multiple paths to reach a specific node, the decision to choose next node is based on the probability value. This probability value is influenced by previously available pheromone value. The pheromone will enforce the fastest (shortest) route to the destination. The path building is bidirectional process. On successful reach of destination, the ants travel back to source by strengthening the path between source and destination. The process of food searching in ant is used in numerous applications in the real world such as industry, design, vehicle routing, network and gaming to name a few. It is also used in solving traveling salesmen problem and quadratic assignment problem. The different types of ant algorithms are antnet, ant based control, mobile agent based routing, ant algorithm for mobile ad-hoc network and termite. This work uses antnet algorithm.

Each node in the network periodically launches forward ant towards destination. It explores the network to reach the destination. Forward ant collects the information of network like node identification, time stamp at which the specific node was visited etc. This information is stored in the stack of forward ant. On visiting a node it deposits pheromone to indicate the path taken. It takes a probabilistic decision in case of multiple paths. Forward ants travel with the same priority as that of data packet; hence they suffer the same network delays. Forward ant on successfully reaching destination, gets converted to backward ant. The stack of forward ant is copied to the queue of backward ant. Backward ant follows the same path as forward ant to reach source. It updates the routing table at each node, based on the information collected by forward ant (queue). Backward ants are given higher priority than forward ant. Backward ants are killed once they reach source.

## 4. Genetic Algorithms

Genetic Algorithm (GA) is a programming technique that depicts the biological evolution as the problem solving technique. GA works on the search space called population [7]. Each element in the population is called as chromosome. GA begins with randomly selecting set of feasible solution from population. Each chromosome is a solution by itself. Each chromosome is evaluated for fitness and this fitness defines the quality of solution. GA uses adaptive heuristic search technique which finds the set of best solution from the population. New offsprings are generated /evolved from the chromosomes using operators like selection, crossover and mutation. Most fit chromosomes are moved to next generation. The weaker candidates get less chance for moving to next generation. This is because GA is based on the principle of Darwin theory of evolution, which states that the “survival is the best”. This process repeats until the chromosomes have best fit solution to the given problem [8]. The summary is that the average fitness of the population increases at each iteration, so by repeating the process for many iterations, better results are discovered.

GA has been widely studied and experimented on many fields of engineering. GA provides alternative methods for solving problems which are difficult to solve using traditional methods. GA can be applied for nonlinear programming like traveling salesman problem, minimum spanning tree, scheduling problem and many others. Using a GA for difficult scheduling problems enables relatively arbitrary constraints and objectives to be incorporated painlessly into a single optimization method.

### 4.1 Strengths of Genetic algorithm

- The most important feature of genetic algorithms is that they are parallel in nature. They explore solution space in multiple directions at once. GA is well suited for solving problems where the solution space is huge and time taken to search exhaustively is very high.
- They perform well in problems with complex fitness. If the function is discontinuous, noisy, changes over time or has many local optima, then GA gives better results [9].
- GA has ability to solve problems with no previous knowledge (blind).

The performance of GA is based on efficient representation, evaluation of fitness function and other parameters like size of population, rate of crossover and mutation and the strength of selection.

It is found that 55% of packets are lost when they visit same node more than once. In order to reduce this, the ants are made to explore network independently of routing table information. By doing this the dependency on routing table can be eliminated. The aim of this work is to evolve solution to packet switched routing problem.

## 5. Problem definition

The network under consideration is represented as G = (V, E), a connected graph with N nodes. The metric of optimization is cost of path between the nodes. The total cost is the sum of cost of individual hops. The goal is to find the path with minimum total cost between source node $V_s$ and destination $V_d$, where $V_s$ and $V_d$ belong to V.
This paper presents the efficient on-demand, source initiated routing algorithm using ant algorithm and genetic algorithm. It is implemented as two modules viz., Network generation or initialization of routing tables using ant algorithm and generation of optimal path using genetic algorithm. Finally data is sent along the generated path.

### 5.1 Initialization of routing table

This module is used to generate all possible paths from a given node to all other nodes in the network. Initially, 'n' random paths are considered (chromosome). This 'n' defines the population size. These chromosomes act as population of first generation. Forward ant is created for each chromosome. It is allowed to explore the network and find paths to all the nodes in the network. Forward ant chooses the next node to visit, based on the gene value at that position (allae) and the pheromone concentration of the link.

### 5.2 Processing at each node

On arrival of ant, It is required to find the type of ant (forward ant or backward ant). If forward ant, it explores path to destination and if backward ant retrace path to source by updating routing tables at intermediate nodes. Following statements describe the functions of a node for each type of ant.

```
if (forward ant)
{ Get the next node based on the value
          of gene position
  if (the link is available and no loop
      caused) then
  {
     • Update forward ant with network status (stack)
     • Send forward ant to the next node
   }
  else if   (no such link exist)
  {
     • Create backward ant and load contents of
     forward ant to backward ant (queue).
     • send Backward ant towards source along the
     same path as forward ant
  }
}
if backward ant
{
  if current node is source node
  {
     • Store path and kill backward ant
     • Update routing table
  }
 else
{
  • Forward backward ant on to link available on
      queue
  • Update routing table
}
if next node is not available
        Kill backward ant
}
```

### 5.3 Generating optimal paths

This phase deals with finding the optimal path using genetic algorithm. The input to this module is the set of paths generated one for each backward ant. Each path is called as chromosome.
Once the source node receives 'm' (say 10- population size) number of backward ant,

1. Calculate the fitness of each of the chromosome.

The fitness of the chromosome is evaluated based on the equation Eqn 1.

Fitness = no of hops * 10 – total cost of path
-Eqn.1.

Number of hops defines the number of intermediate nodes visited along the path from source to destination and total cost is the sum of cost of individual links in the path.

2. Select best two chromosome s as parents (Roulette wheel method)
3. Perform crossover with probability 0.6
4. Perform mutation with probability 0.01 (Insertion)
5. Place children in the population and eliminate the worst chromosome
6. If termination condition is not attained then repeat the steps 1 to 6 else (termination condition is reached )
   {
   - Store the paths for duration t seconds
   - send data to the destination along the path

   }
7. Refresh the path after duration of t seconds.

In dynamic network, the status of the network changes with time. In order to depict the current status of the network accurately, forward ants are launched at every t seconds and optimal routes are recomputed.

### 5.3.1 Selection of parents

This section presents different methods for selecting parents for next generation. Following are some of the selection methods.

**Elitist selection**- In this selection most fit chromosomes are guaranteed to be selected for next generation.

**Rank selection**- In rank selection each individual is selected based on its rank.

**Roulette wheel selection** – In roulette wheel selection, the individual is selected based on the relative fitness with its competitors. This is similar to dividing the wheel into a number of slices. More fit chromosomes get larger slice. For selecting the chromosome for next generation, the wheel is spun. Once the wheel stops, the individual corresponding to the slice on which it lands goes to next generation. As higher fit chromosomes have larger slice, it ensures that most fit chromosomes have higher chance of going to next generation [7]. Some of the other selection methods are scaling selection, tournament selection, rank selection etc. Current work is based on roulette wheel selection.

### 5.3.2 Crossover

Crossover or recombination operator combines sub parts of two parent chromosomes and produces offspring that contains some part of both the parent genetic material. Crossover is mainly of two types namely single point crossover and multipoint crossover. In single point crossover, there is one cross over site and in multipoint crossover there is more than one crossover site. In single point crossover, one offspring consists of part before crossover site of parent 1 and part after crossover site of parent 2, another offspring consists of part before crossover site of parent 2 and part after crossover site of parent 1. Following shows the example for single point crossover. The parents are paths from source 1 to destination 6 and the crossover site is 4.

Parent 1:1 2 3 4 | 5 6
Parent 2:1 5 4 2 | 3 6
Offspring 1: 1 2 3 4 3 6
Offspring 2: 1 5 4 2 5 6

Though this method is simple, it has some problems like formation of cycles when used for routing. From the previous example, considering the first, offspring 1: 1 2 3 4 3 6, has cycle formed 1 2 3 4 3 which is not desirable. Hence it is required to use some of the advanced multipoint crossover techniques to eliminate cycle. Some of the advanced multipoint crossover techniques are Partially Mapped Crossover (PMX), Cycle crossover (CX) and Order crossover (OX) [7][8]. This paper deals with PMX method. In PMX two crossover sites are picked up randomly along the string. The strings between the crossover sites are exchanged position by position, other elements are determined by ordering information, which is partially determined by each of its parents. PMX is designed to preserve absolute positions from both parents.

Example: Parent 1 :1 2 3 | 4 5 | 6 7
Parent 2 :1 5 4 | 6 2 | 3 7
Offspring 1 :1 5 3 6 2 4 7
Offspring 2 :1 2 6 4 5 3 7

In the above example two crossover points at position 3 and position 5 are considered. The contents between the two crossover sites of parents are swapped and placed in offspring. Remaining elements are mapped accordingly. Once the offspring is generated it should be validated. Validation is done by checking the chromosome (offspring) with all possible routes. If offspring belongs to all possible routes then its fitness is computed and sent to next operation. If the offspring does not belong to all possible route set, then it is dropped.

### 5.3.3 Mutation

It may be possible that crossover operation may produce degenerate population. In order to undo this, mutation operation is performed. Mutation operation can be inversion, insertion, reciprocal exchange or others. In case of inversion two random points are selected and the string between them is reversed. In case of insertion a node is inserted at random position in the string. In reciprocal exchange, nodes at two random positions are exchanged

[10]. This paper uses insertion. This is because a node along the optimal path may be eliminated through crossover. By using insertion, it can be brought back. Once mutation is completed, the offspring generated by mutation have to be validated with the same technique used in crossover.

Once the optimal solution is generated using GA, data is transmitted along that path. There may be change in topology of network as some nodes may join the network or some nodes may leave the network or some nodes may fail. The change in network may be detected by the algorithm at regular intervals, ΔT seconds. Under these circumstances the optimal path may no more be the shortest. Hence the network has to be refreshed at every ΔT seconds and new routes may be generated .

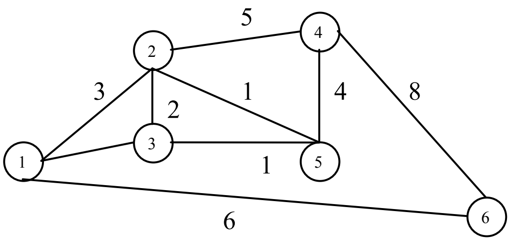


Fig 1.0 Sample topology

Generate 20 random chromosomes

| Chromosome | Delay | Fitness |
|---|---|---|
| 1 6 4 5 3 2 | 21 | 29 |
| 1 3 2 5 4 6 | 22 | 28 |
| 1 2 4 5 3 6 | 13 | 27 |
| 1 2 4 5 3 6 | 13 | 27 |
| 1 2 5 4 3 6 | 8 | 22 |
| 1 2 3 6 4 5 | 5 | 15 |
| 1 2 4 3 6 5 | 8 | 12 |
| 1 3 5 6 4 2 | 8 | 12 |
| 1 2 4 3 5 6 | 8 | 12 |
| 1 2 4 3 6 5 | 8 | 12 |
| 1 2 6 4 5 3 | 3 | 7 |
| 1 2 6 4 3 5 | 3 | 7 |
| 1 6 4 3 2 5 | 14 | 6 |
| 1 6 4 3 5 2 | 14 | 6 |
| 1 6 3 4 5 2 | 6 | 4 |
| 1 6 5 2 3 4 | 6 | 4 |
| 1 6 3 2 5 4 | 6 | 4 |
| 1 3 6 5 4 2 | 7 | 3 |
| 1 3 6 4 2 5 | 7 | 3 |
| 1 4 5 6 3 2 | 0 | 0 |

********************************
GENERATION 1
********************************

| Chromosome | Nodes Visited | Delay | Fitness |
|---|---|---|---|
| 1 6 4 5 3 2 | 6 | 21 | 29 |
| 1 3 2 5 4 6 | 6 | 22 | 28 |
| 1 2 4 5 3 6 | 5 | 13 | 27 |
| 1 2 4 5 3 6 | 5 | 13 | 27 |
| 1 2 5 4 3 6 | 4 | 8 | 22 |
| 1 2 3 6 4 5 | 3 | 5 | 15 |
| 1 2 4 3 6 5 | 3 | 8 | 12 |
| 1 3 5 6 4 2 | 3 | 8 | 12 |
| 1 2 4 3 5 6 | 3 | 8 | 12 |
| 1 2 4 3 6 5 | 3 | 8 | 12 |

## 6. Results

Current work is tested on the network consisting of 6 nodes. The topology of the network is shown in Fig 1.0. Initially 20 random chromosomes are generated, out of which best 10 is considered for generation -1. Crossover and mutation functions are applied to generate next generation chromosomes. At each generation the chromosomes are validated and best fit chromosomes are sent to next generation. It is found that the fitness value increases at each generation.

********************************GENERA
TION 2
********************************

| Chromosome | Nodes Visited | Delay | Fitness |
|---|---|---|---|
| 1 2 3 5 4 6 | 6 | 18 | 32 |
| 1 2 4 5 3 6 | 5 | 13 | 27 |
| 1 2 4 5 3 6 | 5 | 13 | 27 |
| 1 2 4 3 6 5 | 3 | 8 | 12 |
| 1 2 4 3 5 6 | 3 | 8 | 12 |
| 1 2 4 3 6 5 | 3 | 8 | 12 |
| 1 2 6 5 3 4 | 2 | 3 | 7 |
| 1 6 2 4 3 5 | 2 | 6 | 4 |
| 1 4 6 5 3 2 | 1 | 0 | 0 |
| 1 5 3 6 4 2 | 1 | 0 | 0 |

********************************
GENERATION 3
********************************

| Chromosome | Nodes Visited | Delay | Fitness |
|---|---|---|---|
| 1 2 3 5 4 6 | 6 | 18 | 32 |
| 1 6 4 5 3 2 | 6 | 21 | 29 |
| 1 2 4 5 3 6 | 5 | 13 | 27 |
| 1 2 4 5 6 3 | 4 | 12 | 18 |

```
1 2 4 6 3 5   4     16      14
1 2 4 3 5 6   3      8      12
1 2 4 3 6 5   3      8      12
1 3 5 6 4 2   3      8      12
1 2 6 5 3 4   2      3       7
1 6 2 4 3 5   2      6       4

*******************************
GENERATION   4
*******************************
-------------------------------
Chromosome  Nodes
           Visited Delay Fitness
-------------------------------
1 2 3 5 4 6  6      18     32
1 6 4 5 3 2  6      21     29
1 2 4 5 3 6  5      13     27
1 2 4 5 3 6  5      13     27
1 2 4 5 6 3  4      12     18
1 2 4 6 5 3  4      16     14
1 2 6 5 3 4  2       3      7
1 6 2 4 3 5  2       6      4
1 6 2 4 3 5  2       6      4
1 5 3 6 4 2  1       0      0


*******************************
GENERATION 5
*******************************
-------------------------------
Chromosome  Nodes
           Visited Delay Fitness
-------------------------------
1 2 3 5 4 6  6      18      32
1 6 4 5 3 2  6      21      29
1 2 4 5 3 6  5      13      27
1 2 4 5 3 6  5      13      27
1 2 4 5 6 3  4      12      18
1 2 4 6 5 3  4      16      14
1 2 6 5 3 4  2       3       7
1 2 6 4 3 5  2       3       7
1 6 2 4 3 5  2       6       4
1 4 5 3 6 2  1       0       0


*******************************
GENERATION  6
*******************************
-------------------------------
Chromosome  Nodes
           Visited Delay Fitness
-------------------------------
1 2 3 5 4 6   6   18      32
1 6 4 5 3 2   6   21      29
1 3 2 5 4 6   6   22      28
1 2 4 5 3 6   5   13      27
1 2 5 4 3 6   4    8      22
1 2 4 5 6 3   4   12      18
1 2 3 6 4 5   3    5      15
1 2 4 6 5 3   4   16      14
1 2 4 3 6 5   3    8      12
1 3 5 6 4 2   3    8      12

-------------------------------
```

After the path to all nodes from source node 1 is computed. The set of paths to a specific node will be displayed. Let the destination node is node 3. Following is the set of paths from node 1 to node 3.

```
--------------------------------------------------------
Routes to the destination 3
-------------------------------
Source Destination Delay Route
 ------------------------------
   1    3    5       1 2 3
   1    3    7       1 3
   1    3    13      1 2 4 5 3
   1    3    19      1 6 4 5 3
 ------------------------------
```

## 7. Conclusion

In this work ant algorithm and genetic algorithm are used for routing in packet switched data networks. Ant algorithm, is found to reduce the size of routing table. Genetic algorithm cannot use global information of the network. Hence, the combination of these two algorithms, which makes the packets to explore the network independently, helps in finding path between pair of nodes effectively. The proposed algorithm creates initial population, forwards forward ant, access fitness, generate new population using genetic operators and update routing table.

## 8. Future Enhancements

It is suggested to improve the current work by enhancing it support for load balancing. It can also be improved for using better crossover and mutation techniques and different probabilities. The other place where the improvement can be done is to extend this protocol for an environment where multiple applications are running.

### Acknowledgement

The authors gratefully acknowledge the support of Dr S C Sharma, Principal, R V College of Engineering, Bangalore and Professor B I Khodanpur, Head Department of Computer Science and Engineering, R V College of Engineering, Bangalore for their invaluable support and encouragement.

## References


[1] Andrew S Tannenbaum, "Computer Networks", 4th Edition, Prentice-Hall of India

[2] Dorigo M & Di Caro G, "Antnet: A Distributed stigmergetic control for communication network", Journal of Artificial Intelligence Research, pp 317-365, 1998

[3] Schoonderwoerd R , Holland O Bruten J, "Ant like agents for load balancing in Telecommunication Networks", Hewlelt-Packard Laboratories, Bristol-England, 1997

[4] Tony White, "Routing with Swarm Intelligence" SCE Technical Report SCE-97-15, September 1997.

[5] Dorigo M & Gambardella L, "Ant colony system: A Cooperative learning approach to the travelling salesman problem", IEEE Transaction on Evolutionary Computation, Vol. 1, N1, pp53-66

[6] Liang S , Zincir Heywood A N, Heywood M I, "The effect of Routing under local information using a Social insect Metaphor", IEEE International Congress of Evolutionary Computation, May 2002, pp 1438-1443

[7] David E Goldberg, "Genetic Algorithms in search, optimization and machine learning", Pearson Education.

[8] Ed Laurence Davis, "Hand book of Genetic Algorithms".

[9] K Vijayalakshmi and S Radhakrishnan, "Dynamic Routing to Multiple Destinations in IP Networkin using Hybrid Genetic Algorithm (DRHGA), International Journal of Information Technology, Vol 4, No 1, PP 45-52.

[10] Aluizio F. R. Araújo, Cícero Garrozi and André R.G.A. Leitão Maury M. Gouvêa Jr, "Multicast Routing Using Genetic Algorithm Seen as a Permutation Problem", Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06)


**Cauvery NK** has completed her ME from Bangalore university in the year 1999 in the field of Computer Science and Engineering. She has an academic experience of 10 years in R V College of Engineering, Bangalore. She has four publications in International Conference. She is a member of CSI and ISTE. Her area of research includes Swarm Intelligence routing and Genetic Algorithms.

**K. V. Viswanatha** has completed his Ph D from IISC, Bangalore in the year 1975 in the field of Electrical Communication Engineering. He has worked in several private sector enterprises in the area of Software development for about 18 years. He has been working in academics for about 15 years. He has to his credit two International and one National publication. His area of research includes Algorithms and Network Security