

# Role of Genetic Algorithm in Routing for Large Network

\*Mr. Kuldeep Kumar,

Computer Programmer, Krishi Vigyan Kendra,  
CCS Haryana Agriculture University, Hisar.  
Haryana, India  
verma1.kuldeep@gmail.com

#Mr. Ramkala,

Teaching Associate, Department of Computer Science & Engineering,  
Guru Jambheshwar University of Science and Technology Hisar.  
Haryana, India  
ramkalapunia@gmail.com

**Abstract: Routing Algorithm is that part of the network layer that is responsible for deciding the path for the incoming packets on which they should be transmitted. There are number of routing algorithms. We have to choose the best algorithm that produces the optimal path for each incoming packets on the large network. For optimal path, routing algorithm uses many routing protocol because the performance and reliability of Internet depend on the operation of the underlying routing protocol. IP routing protocols compute paths based on the network topology and configuration parameters, without regard to the current traffic load on the routers and links. Here we describe the Genetic Algorithm for routing optimization. Genetic algorithms (GA) are search algorithms based on the principles of natural selection and genetics. The genetic algorithm is used to solve a network routing protocol. The algorithm has to find the shortest path between the source and destination nodes. Simulation results are carried out using MATLAB.**

*Key words: Routing Algorithm, Routing Protocol, Genetic Algorithm, Search Methods.*

## 1. INTRODUCTION

In most networks, packets will require multiple hops to travel from one location to another location. Routing procedure is used to travel packets from one location to another. Routing is a process of transferring packets from source to destination with minimum cost. Cost factors may be the distance of router, network throughput of a link or link availability and reliability. In a large network routing is an issue if the source and destination are not on the same network. 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.[1] The algorithms that choose the routes and data structure is the major issue of network layer design. The routing algorithms are the part of the network layer and are used to specify the best optimal path on a large network. The route followed by the packets may change every time when a packet arrives every time or it may remain same just as followed by the first packet. Regardless of whether routes are chosen independently for each packet or only new connections are established, there are certain properties that are desirable in routing algorithm: correctness, simplicity, robustness, stability, fairness and optimality. Correctness and simplicity are not the major issues in routing but robustness and stability are major issues. Once a packet comes on the network how much our network and routing algorithm adopted by the network is robust. During the period of transmission of packets there will be hardware and software failures of all kinds on the network. The routing algorithm should be able to cope with changes in the topology and traffic. Stability is also an important goal for the routing algorithm. The optimality of routing algorithms means that minimizing the packet delay and maximizing total network throughput.[2]

Routing algorithms can be grouped in two categories first is adaptive and the second is non-adaptive. Adaptive routing algorithm changes their routing decisions to reflect changes in the topology or traffic load on the links. In contrast, Non-adaptive routing algorithms do not base their routing decisions on measurements or estimates of the current traffic and topology of the network. There are number of routing algorithms under the categories adaptive and non-adaptive. These are used to find the path between sources and destinations. Most of these algorithms work on the cost to find the routs so each path evaluated in terms of cost. The most notably OSPF (Open Shortest Path First) protocol is generally used to find the shortest route between the source and destination. Open Shortest Path First is a dynamic protocol. It is a link state protocol and is part of the interior gateway protocol group. This protocol support routing based on type of service and to be able to route real-time traffic one way and other traffic a different way. OSPF support three kinds of connections and networks. First is Point-to-Point links between exactly two routers. Second is Multi-access networks with broadcasting and the

third is Multi-access networks without broadcasting. OSPF works by abstracting the collection of actual networks, routers and links in to a directed graph in which each edge is assigned a cost i.e. distance, delay etc. It computes the shortest path based on the weight assigned to the each edges. [2][3] The basic working of OSPF routing protocol are as follows:

1. When a router boots, it sends HELLO packets to all of its point-to-point links and the router receives the HELLO packets in response by which each router learns who its neighbors are.
2. In the second step each router periodically floods LINK STATE UPDATE message to each of its adjacent routers to describe its database to all other devices. So that all the routers connected together have an up to date description of each topology that is connected to each router or a router can see whether an incoming LINK STATE UPDATE is older or newer than what it currently has. Routers also send these messages when a link goes up or down or its cost changes.
3. At the last step each router will calculate a mathematical data structure called “Shortest Path Tree” that describes the shortest path to the destination address. Using flooding, each router informs all the other routers in its area of its neighbors and costs. This information helps each router to construct the graph for its area and compute the shortest path. [2]

## 2. SHORTEST PATH ALGORITHM

The most common algorithm used for shortest path algorithm is Dijkstra’s algorithm that calculates the shortest path between two points on a network using graph made up of vertices and edges. A cost value is assigned to every node i.e. zero is set to source vertex and infinity for all other vertices. The algorithm divides the vertices into two sets: tentative and permanent.[2] This algorithm chooses vertices, makes them tentative, examines them and if they pass the criteria, makes them permanent.[1] The steps are describes as follows:

1. Start with the source or root node of the tree.
2. Assign a cost of 0 to this node and make it the first permanent node.
3. Examine each neighbor node of the source node that was the last permanent node.
4. Assign a cumulative cost to each node and make it tentative.
- 5 From the list of tentative nodes:
  - i). Find the node with smallest cumulative cost and make it permanent. A permanent node will be not checked ever again; its cost is recorded as final.
  - ii). If a node can be reached from more than one direction, select the direction with the shortest cumulative cost.
6. Repeat steps 3 to 5 until every node becomes permanent.

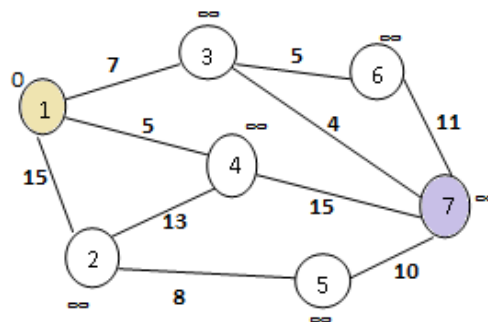


Figure 1: Network topology

By applying Dijkstra’s algorithm to Fig. 1 to calculate the shortest path between node 1 and the destination node 7; the shortest path will be 1-3-7 with cost 11.

## 3. GENETIC ALGORITHM

Genetic algorithms (GA) are search algorithms based on the principles of natural selection and genetics. The most important idea that stands beyond the initial creation of GAs is the aim of developing a system as robust and as adaptable to the environment as the natural systems. GA operates on a population of potential solutions applying the principle of the survival of the fittest to produce better and better approximations to the solution of the problem that GA is trying to solve. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness value in the problem domain and breeding them together using the Operators borrowed from the genetic process performed in the nature, i.e. crossover and mutation. This process leads to the evolution of the populations of individuals

that are better adapted to their environment than the individuals that they were created from, just as it happens in natural adaptation. [4]

Figure 1 illustrates the operation of a general genetic algorithm. The operation starts from an initial population of randomly generated individuals. Then the population is evolved for a number of generations and the qualities of the individuals are gradually improved. During each generation, three basic genetic operators are sequentially applied to each individual with certain probabilities, *i.e.*, selection, crossover, and mutation. First, a number of best-fit individuals are selected based on a user-defined fitness function. The remaining individuals are discarded. Next, a number of individuals are selected and paired with each other.

Each individual pair produces one offspring by partially exchanging their genes around one or more randomly selected crossing points. At the end, a certain number of individuals are selected and the mutation operations are applied, *i.e.*, a randomly selected gene of an individual abruptly changes its value[4].

### 3.1 STRUCTURE OF GENETIC ALGORITHM

GA evolves a population of initial individuals to a population of high quality individuals, where each individual represents a solution of the problem to be solved. Each individual is called chromosome, and is composed of a certain number of genes that in general case does not have to be fixed. The quality of each rule is measured by a fitness function which is quantitative representation of each rule's adoptions to the environment, *i.e.* the problem to be solved. The procedure starts from an initial population is evolved for a number of generations while the qualities of increasing the fitness value as the measure of quality. During each generation, three basic genetic operators are sequentially applied to each individual with certain probabilities, *i.e.* selection, crossover and mutation [8].

Crossover consisting of exchanging of the genes between two chromosomes performed in a certain way, while mutation consists of random changing of a value of a randomly chosen gene of a chromosome. Both crossover and mutation are performed with a certain possibility, called crossover/mutation rate. The algorithm flow is presented in Fig 2

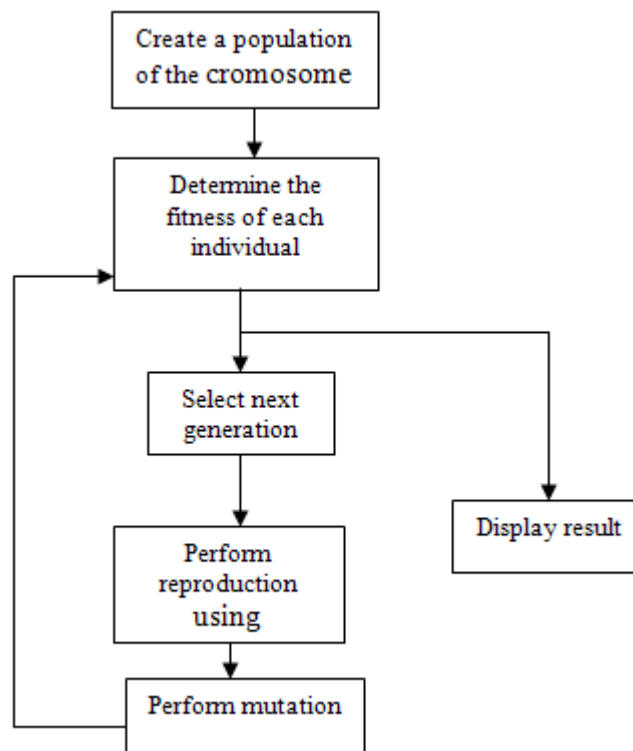


Figure 2: Genetic Algorithm Process

## 4. PROPOSED ALGORITHM

Now a connected Network with N nodes is considered. The cost of path between the nodes is taken for optimization factor and the total cost is the sum of the individual hop in the path. The main objective is to find the minimum total cost between source node and the destination node. This paper presents a Genetic Algorithm to find the shortest path that is more effective and simple. There may be any path from source node to destination node is a feasible solution but the optimal solution is the shortest one. When the algorithm starts a

set of random population of strings is generated which represents feasible or unfeasible solution. Unfeasible solutions are those strings that cannot reach the destination. And, feasible solutions are those strings that can reach the target node.[7] Unfeasible solutions has assigned lowest fitness value i.e. 0. Now consider the network topology in fig 3 with 10 nodes and 16 edges with their corresponding weight.

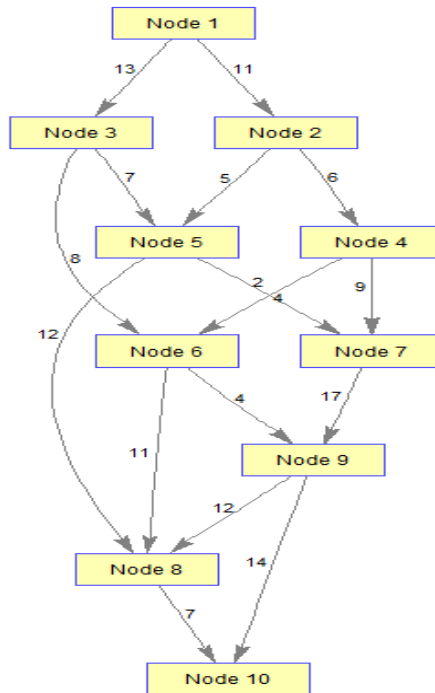


Figure 3: Network for simulation

In Fig. 3 each node is assigned a number that is used to encode the path as a string expressed by the order on numbers. The source node is node 1 and destination node 10. In this figure there are number of feasible path. For example, 1-2-5-8-10 is a feasible path with source node 1 and destination node 10 and it might be optimal or not. The Chromosome is represented by a string of bits and these bits are in binary form. After that crossover and mutation operators are applied these strings of bits. During the crossover, a string that has an efficient fitness is randomly selected as a parent. If the second parent contains the common number the first one; both strings exchange the part of their strings following the common number. If not another string is selected as the second parent and same procedure is followed. Now using the Network diagram in figure 3 two chromosomes are specified:

**Chrom 1:** 1-2-4-7-9-10

**Chrom 2:** 1-3-5-7-9-8-10

Now apply crossover operation on these chromosomes at node 7. After applying this operation the chromosomes are:

**Chrom 3:** 1-2-4-7-9-8-10

**Chrom 4:** 1-3-5-7-9-10

The important question is how to makes the binary coding so possible and easy to be used in a fixed length chromosome definition. So, the path is encoded using binary numbers where each node in a chromosome is encoded by 4 bits binary number with their respective linked nodes as given in Table 1.

Node	Binary Code	Linked Nodes
1	0001	2,3
2	0010	4,5
3	0011	5,6
4	0100	6,7
5	0101	7,8
6	0110	8,9
7	0111	9
8	1000	10
9	1001	8,10
10	1010	Destination

Table 1: Binary Coding of the Network vertices

Each chromosome used in the used genetic algorithm represent a possible solution of the optimization problem i.e. each chromosome represents a path which consists of a set of vertices complete the feasible solution i.e. from the source node to destination node.[1] The maximum length of the chromosome is the number of node multiplied by the number of bits used by a particular node. Each vertex is represented by 4 bits as shown in Table 1. So the total numbers of bits in a chromosome are 10\*4=40 bits. The first gene in a chromosome represent node 1 i.e. source node which is coded in 4 binary bits, the next gene is node 2 and so on. The initial population of chromosomes can be randomly generated such that each chromosome has random genes and source and destination nodes are fixed in the population. After the populations generated the fitness function is the next question. The fitness function of a chromosome measures the objective cost function. So the cost of the path used by the different nodes is used to calculate the fitness. The main objective to increase the fitness value is that an efficient path can be chosen.[7] The fitness function can be defined as below:

$$F(X) = \frac{X}{\sum X} + \frac{n}{\sum n} \dots\dots\dots$$

Where X is the number of the path in the list of possible paths,  $\sum n$  is the number of nodes in the network;  $\sum X$  is the sum of all the paths.

After finding the value of F(X) for each possible path then the fitness for each path is calculated as give by the following function:

$$\text{Fitness} = \frac{F(X)}{\text{Max. } F(X)} \dots\dots\dots$$

Where Max F(X) is the maximum F(X) in the list so that one of the path which has the maximum F(X) will be given Fitness = 1.

### 5. SIMULATION RESULT

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. These factors make MATLAB an excellent tool for teaching and research [5]. It has a very powerful string manipulation commands that help to convert easily the numeric variables into string of bits and vice-versa. Figure 3 is used for simulation work i.e. to find shortest path. In this network diagram the source node is 1 and the destination node is 10 with 16 edges. The initial population consist of 9 chromosomes 40 bits each. The genetic algorithm used the probability of 0.8 and the mutation rate equal to 0.02. The simulation result after 70 generation is given by:

- 1 2 2 5 5 5 8 8 10 10
- 1 2 2 5 5 5 8 8 10 10
- 1 2 2 5 5 5 8 8 10 10
- 1 2 2 5 4 5 8 8 10 10
- 1 2 2 5 5 5 8 8 10 10
- 1 2 2 3 5 5 8 8 10 10
- 1 2 2 5 5 5 8 8 10 10
- 1 2 2 5 5 5 8 8 10 10

In the above results the 4<sup>th</sup> and 7<sup>th</sup> chromosomes are different due to mutation process. The part of the repeated numbers is cut off and the final result will be the optimal or shortest individuals i.e. 1-2-5-7-9-10 with the cost of 35. Fig. 4 describes the population generation and show that after 70 generation the best value is obtained. The obtained results declared to be true in respect of the genetic algorithm where the convergence was guaranteed to obtain the optimal path in each case.

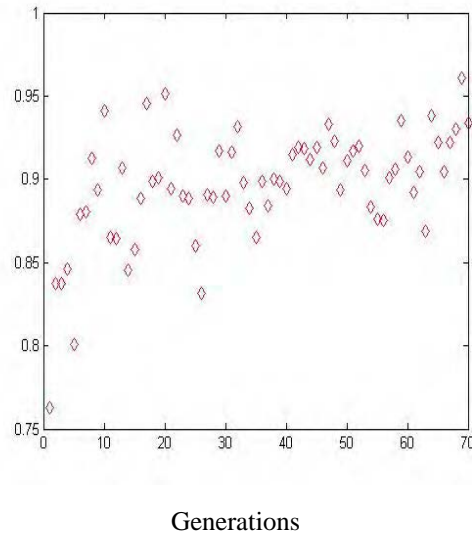


Figure 4: Population generations

## 6. CONCLUSION

Genetic Algorithm is the efficient and simple method for finding the shortest path in a large and dynamic network. The chromosome length depends on the number of nodes in the network. Although the MATLAB environment also provide a default algorithm to find the shortest path and that algorithm is Dijkstra's algorithm. GA is also produce the same result as the Dijkstra's algorithm. In future work, GA can be more improved by decreasing the length of the chromosome used for the vertices in the network to reduce the complexity.

## REFERENCES

- [1] Gihan Nagib and Wahied G.Ali, "Network Routing Protocol Using Genetic Algorithms", International Journal of Electrical & Computer Sciences, Vol:10 No:02, pp. 40-44, April, 2010.
- [2] Computer Networks (3rd Edition) by Andrew S. Tanenbaum, David J. Wetherall.
- [3] N.Selvanathan and W.J. Tee, "Genetic Algorithm Solution to Solve the Shortest Path Problem OSPF and MPLS", Malaysian Journal of Computer Science, Vol. 16 No. 1, pp. 58-67, 2003.
- [4] D.E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley Publishing Company, book: ISBN: 0201157675, 1989.
- [5] Finding shortest path in Network using MATLAB, Available online at <http://hubpages.com/hub/Shortest-Path-Routing--Finding-Shortest-Path-in-Network-using-MATLAB>
- [6] Edward William; "Genetic Programming Lab Toolset version 3.0". *IEEE*; 2008.
- [7] Yousra Ahmed Fadil, "Routing using Genetic Algorithm for Large Networks", Diyala Journal of Engineering Sciences, Vol 03, No. 02, pp. 53-70, Dec.,2010.
- [8] H. Pohlheim; "Genetic and Evolutionary Algorithms: Principles; Methods and algorithms" <http://www.geatbx.com/docu/index.html>, January 2005).

## AUTHORS PROFILE

1. Kuldeep Kumar, Computer Programmer, CCS Haryana Agriculture University, Hisar (Hry.)
2. Ramkala , Teaching Associate, Dept. of CSE, Guru Jambheshwar University S & T, Hisar (Hry.)