International Journal of Scientific Research Engineering & Technology (IJSRET) ISSN: 2278–0882
IEERET-2014 Conference Proceeding, 3-4 November, 2014

Network Routing Algorithm using Genetic Algorithm and Compare with Route Guidance Algorithm

Uma Agarwal[#1],
[#]Electronics Department, Gyan Vihar University
[1]uma01988@gmail.com

Vipin Gupta[#2]
Electronics Department, Gyan Vihar University
[2]vipinsbcet@gmail.com

Abstract— **This paper aims to develop a genetic algorithm to solve a network routing protocol problem. The algorithm has to find the shortest path between the source and destination nodes. In the literature, the routing problem is solved using search graph techniques to find the shortest path. Route guidance algorithm is use to find best shortest path in routing network, This is poised to minimize costs between the origin and destination nodes. Dijkstra's algorithm is one of popular techniques to solve this problem. The proposed algorithm was compared with the Dijkstra algorithm in order to find the best and shortest paths. Simulation results are carried out for both algorithms using MATLAB. The results affirmed the potential of**
the proposed genetic algorithm.

Keywords— **Search methods, Genetic Algorithms, Protocols, Routing, Route guidance system (RGS).**

I. Introduction

Dijkstra's algorithm, conceived by Dutch computer scientist Edsger Dijkstra in 1959 is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge path costs, producing a shortest path tree [2]. This algorithm is often used in routing. An equivalent algorithm is developed by Edward F. Moore in 1957. For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. It can also be used for finding costs of shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined. For example, if the vertices of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities. The shortest path first is widely used in network routing protocols, most notably OSPF (Open Shortest Path First). OSPF is a dynamic routing protocol. It is a link state routing protocol and is part of the interior gateway protocols group. OSPF keeps track of the complete network topology and all the nodes and connections within that network. The basic workings of the OSPF routing protocol are as follows:

A. Start up

When a router is turned on it sends Hello packets to all neighboring devices, and the router receives Hello packets in response. From here routing connections are synchronized with adjacent routers that agree to synchronize.

B. Update

Each router will send an update message called its “link state” 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.

C. Shortest path tree

Each router will calculate a mathematical data structure called “shortest path tree” that describes the shortest path to the destination address, this is where OSPF gets its name. It will try to open the shortest path first.

OSPF routing protocol is a very important protocol to consider when setting up routing instructions on the network. As OSPF gives the routers the ability to learn the most optimal (shortest) paths it can definitely speed up data transmission from source to destination. In the literature, Dijkstra's algorithm is often described as a greedy algorithm. The *Encyclopedia of Operations Research and Management Science* describes it as a "*node labeling greedy algorithm*" and a greedy algorithm is described as "*a heuristic algorithm that at every step selects the best choice available at the step without regard to future consequence*" [4].

Routing is a process of transferring packets from source node to destination node with minimum cost (external metrics associated with each routing interface). Cost factors may be the distance of a router (Round-trip-delay), network throughput of a link or link availability and reliability expressed as simple unit less numbers. 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 [9]. 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. In the literature, different approaches are applied to solve this problem as: Dijkstra's algorithm[2] dynamic programming technique, and emerged ants with genetic algorithm [3].

This paper is organized as follows. The literature work and the routing problem definition are presented in section I. Section II describes the basics of Dijkstra's algorithm. While section III; gives a brief description of the genetic algorithms as existed in the literature. The developed genetic algorithm to find the shortest path is introduced in Section IV. While in Section V Developed the Route Guidance Algorithm Simulation results are presented and discussed in Section VI. Finally, conclusion is drawn in Section VII.

II. DIJKSTRA'S ALGORITHM

The Dijkstra's algorithm calculates the shortest path between two points on a network using a graph made up of nodes and edges. It assigns to every node a cost value. Set it to zero four source node and infinity for all other nodes. The algorithm divides the nodes into two sets: tentative and permanent. It chooses nodes, makes them tentative, examines them, and if they pass the criteria, makes them permanent. The algorithm can be defined by the following steps [2], [9]:

1. Start with the source node: the root 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 node that was the last permanent node.

4. Assign a cumulative cost to each node and make it tentative.

5. Among the list of tentative nodes

a. Find the node with the smallest cumulative cost and mark it as permanent. A permanent node will not be checked ever again; its cost recorded now is final.

b. 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.

If the algorithm is applied to the network in Figure 1 to calculate the shortest path between the source node a(1) and the destination node b(5); the shortest path will be 1-3-6-5 with cost 20.

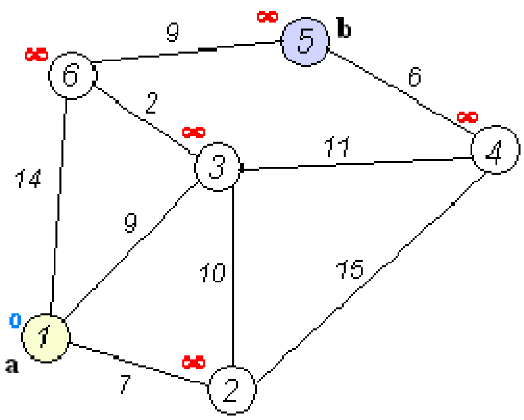

Fig. 1 - Network topology

III. GENETIC ALGORITHMS

Genetic algorithms (GAs) are global search and optimization techniques modeled from natural selection, genetic and evolution. The GA simulates this process through coding and special operators. The underlying principles of GAs were first published by [1]. Excellent reference on GAs and their applications is found in [5]. A genetic algorithm maintains a population of candidate solutions, where each candidate solution is usually coded as binary string called a chromosome. The best choice of coding has been shown to be a binary coding [3]. A set of chromosomes forms a population, which is evaluated and ranked by fitness evaluation function. The fitness evaluation function play a critical role in GAs because it provides information how good each candidate. The initial population is usually generated at random. The evolution from one generation to the next one involves mainly three steps: fitness evaluation, selection and reproduction [10].

First, the current population is evaluated using the fitness evolution function and then ranked based on their fitness. A new generation is created with the goal of improving the fitness. Simple GA uses three operators with probabilistic rules: reproduction, crossover and mutation. First selective reproduction is applied to the current population so that the string makes a number of copies proportional to their own fitness. This results in an intermediate population.

Second, GA select "parents" from the current population with a bias that better chromosome are likely to be selected. This is accomplished by the fitness value or ranking of a chromosome.

Third, GA reproduces "children" (new strings) from selected parents using crossover and/or mutation operators. Crossover is basically consists in a random exchange of bits between two strings of the intermediate population. Finally, the mutation operator alters randomly some bits of the new strings. This algorithm terminates when an acceptable solution is found, when convergence criteria are met or when a predetermined limit number of iteration is reached. The main features of GAs are that they can explore the search space in parallel and don't need the optimized function to be differentiable or have any smooth properties. The precision of the solution obtained depends on the number of bits used to code a particular variable (length of chromosome) and a sufficient number of iterations.

Genetic Algorithm:

Evaluation of Fitness

$$F = \begin{cases} \frac{1}{\sum_{i=1}^{N-1} C_i(g_i, g_{i+1})} & ; Feasible\ path \\ 0 & ; Infeasible\ path \end{cases} \qquad (1)$$

And algorithm is,

1.	**Initialize** the start and destination points
2.	**Generate** randomly the initial population using via nodes in each chromosome
3.	**While NOT (**convergence condition**) DO**
4.	**Evaluate** the fitness for each chromosome in current population using equation (1)
5.	**Rank** the population using the fitness values
6.	**Eliminate** the lowest fitness chromosome
7.	**Duplicate** the highest fitness chromosome
8.	**Apply** randomly crossover process between

	current parents using the given probability, while keeping the start and end nodes without change in the population
9.	**Apply** the mutation process with the given probability
10.	**Generate** the new population
11.	**END**
12.	Output the best individual found
13.	**END**

IV. ROUTE GUIDANCE SYSTEM

In this section, path planning algorithm for route guidance system has been proposed. Then, some basic problem formulation and properties have been discussed. Finally, route guidance system algorithm has been introduced to find dynamic shortest path using the proposed path planning algorithm. The proposed algorithm can also be used for shortest path and finding costs in the urban road network. A route guidance system (RGS) will calculate the shortest path from current vehicle position to destination locations, so that the traveling cost or time for drivers will be minimized. We have used shortest path computation (SPC) to calculate the nearest distance of the destinations with the support of vehicles in the route guidance system. The updated information of the vehicle will be used by the system to find the shortest path from current vehicle position to destination zone for drivers based on current conditions. However, one of the requirements of intelligent transportation system dynamic is real and current information about travel time for a continuous path that can be acquired by several detectors such as magnet sensors, video cameras, GPS, GSM (global system for mobile communication) and also other network traffic sensors on the routes to transport [4], [11].

Other information requirement is the graphical network with equipment hardware, software and communication between simulated model and real traffic network through design and defined protocol data routing system. We have applied the roles of path planning algorithm in route guidance system as follows:

- Transferring information acquired through the sensors;
- Receiving route request from vehicles;
- The shortest path computation and sending it to vehicles drivers.

Route guidance system Algorithm (RGSA)

Proposed algorithm (RGSA) can also be used for finding shortest path (i.e. lowest cost) from origin node (i.e. node 1) to destination node (i.e. node 16) in a road network graph. The input of the algorithm consists of distances directed between nodes as r and a source data from origin and destination paths as s, d in the network graph G. If the distance between two nodes can be defined as path cost, the total cost of a network graph is the sum of all route distances between nodes in a graph.

1.	Input G (V, E) is a data file which composed a set of 'N' nodes and set of 'M' directed edges. PARAMETERS:
2.	R(s; d), a nonnegative number stands for the cost where "s" is start node and "d" is last node. i, j , k; loop index, G(1, i) is array of vertexes source; G(2, i) is array of vertexes destination. G(3, i) is array of edge distance(or cost); W(i) is array of node weights table for each node and P(i) is array of shortest path from the origin to final node. OUPUT: W(k) is a cost data table for all nodes. P(k), a
3.	shortest path data table for a graph. INITIALIZATION: //W(k) array elements are filled with the

4.	maximum value, except
	// the last node which is set to 0.All nodes from last to first
	// nodes are examined for the routes connected nodes.
	// For each edge do operation in two steps as follows:
	set W[1... n-1] = 999 , W(n) = 0 , P(i) = 0 ;
	
	BEGIN
	// step 1: Node weight computation

5.	For all nodes //for each node and edge pick the costs in W(k).
	for j = first to last edges // j is set to the destination node
	if (G(2 , j) = i) // k is set to the source node
	W(k) = W(i) + G(3 , j);
	end if
	end for
	end for
	
	// step 2: Shortest Path computation
	for i = first to last edges
	while (the origin (k) is the same in graph, G)
	if (G(3 , i) = W(k) - W(j))
	P(k) = G(2 , i);
	else
	i = i + 1;
	k = G(1 , i);
	end if
	end while
	end for
	END

This algorithm is expressed based on node weight. Indeed, in this study, a "node weight" is defined as the shortest distance from each node to last network node. If "n" is the last node number, "n-1" and "k" are two node numbers connected to "node n" in the network. And also, regarding Equation (1) and node weight definition, the procedure is presented as follows: All nodes weight in the road graph are calculated in two steps as follows [5], [7]:

Step 1.

Distance between node n-1 and node n is set to the weight of node n-1 and also distance between node k and node n is set to the weight of node k (expect for first and last nodes, each node have two input and one output routes or at least one input and one output route).

Step 2.

Similarly, the procedure continues to calculate weights of all next nodes until the first node weights is calculated. Finally, the weight of the first node will be the minimum distance paths between first points to last nodes in the network. Now, consider calculated node weights in RGSA algorithm as the input data file, the shortest path computation procedure is presented as follows [6]:

i. Network route is started from first node (node 1), then amount of each distance routes connecting the first node minus the weight of first node. If the result number is equal to weights of the each node, then the second node is connected to first node and it will be priority route of network or next optimal route.

ii. Procedure will continue till the route connected to the last node then the shortest path network is determined.

iii. This algorithm is also provide the second shortest path as well.

V. EXPERIMENTAL RESULT

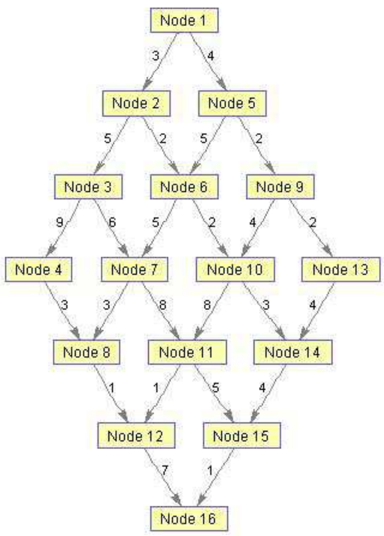

Fig. 2 - When no congestion occur in network

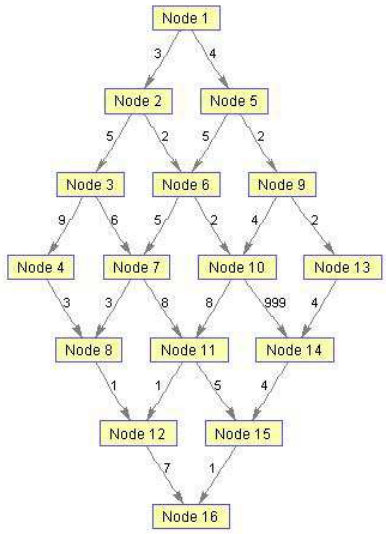

Fig. 3 -When congestion occur in network

Case I

(When no congestion occur in routing network)

Dijkstra algo gives following result

Cost = 15

Path = 1 2 6 10 14 15 16

Pred = 0 1 2 3 1 2 6 7 5 6 10 8 9 10 14 15

Elapsed time is 0.854542 seconds.

GA gives

Cost =15

Path = 1 2 6 10 14 15 16

Time elapsed is 0.000145 seconds.

RGSA algo gives

Cost =15

Path = 1 2 6 10 14 15 16

Time Elapsed is 0.003557 seconds.

Case II

(When congestion occur in network from node 10 to 14)

Dijkestra gives

Cost = 17

Path = 1 5 9 13 14 15 16

Pred = 0 1 2 3 1 2 6 7 5 6 10 8 9 13 14 15

Elapsed time is 1.285112 seconds.

GA gives

Cost = 17

Path = 1 5 9 13 14 15 16

Elapsed time is 0.000290 seconds.

RGSA gives

Cost = 17

Path = 1 5 9 13 14 15 16

Elapsed time is 0.003448 seconds.

So we can say that genetic algorithm gives more accurate result in minimum time.

VI. CONCLUSIONS

This paper includes a new Route Guidance System Algorithm for network routing protocol system to find the shortest path in the routing network to conduct the network from origin to their destination. In experimental case, average path cost gaps (AvePCGap) of GA method are less than the average path cost gaps of Dijkstra algorithm & RGSA and has better performance results. The proposed algorithm is simulated to find the shortest path for 16 nodes routing network for congestion free routing or routing with congestion.

REFERENCES

[1] N. Selvanathan and W. J. Tee, *"A 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.

[2] M. Sniedovich, *"Dijkstra's algorithm revisited: the dynamic programming connexion"*. Journal of Control and Cybernetics **35** (3): 599–620, 2006.

[3] N. K. Cauvery and K. V. Viswanatha, *"Routing in Dynamic Network using Ants and Genetic Algorithm"*, Int. J. of Computer Science and Network Security, Vol. 9 No.3, pp.194-200, March 2009.

[4] B. A. Forouzan, *Data Communications and Networking*, Fourth Edition McGraw Hill, 2007.

[5] D. E. Goldberg, *"Genetic Algorithms in Search, Optimization, and Machine Learning"*, Addison-Wesley Publishing Company, 1989.

[6] Edsger Wybe Dijkstra, *"A Note on Two Problems in Connection with Graphs. journal of Numerical Mathematics"*. vol. 1, pp. 269–271. (1959)

[7] Vasirani, Matteo and Ossowski, Sascha, *"A market-inspired approach to reservation-based urban road traffic management, AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems"*, pp. 617–624. Budapest, Hungary(2009)

[8] Zi Zhang Jian-Min Xu, *"A dynamic route guidance arithmetic based on reinforcement learning, Machine Learning and Cybernetics"*. Proceedings of 2005 International Conference on Vol. 6 , pp. 3607–3611.China (2005)

[9] Schulz, Frank and Wagner, Dorothea and Weihe, Karsten, *"Dijkstra algorithm on-line: an empirical case study from public railroad transport. J. Exp. Algorithmics"*. Vol. 5 , pp. 12. New York, NY, USA (2000)

[10] Zou Liang, Xu Jian-Min, Zhu Lingxiang, *"Application of genetic algorithm in dynamic route guidance system"*. Journal of Transportation Systems Engineering and Information Technology, Vol. 7, pp. 45–48.Chine(2007)

[11] Shi, X. and Hu, J. M. and Xu, Y. S. and Song, J. Y., *"A simulation study on agent-network based route guidance system. In: 8th IEEE Intelligent Transportation Systems Conference (ITSC)"*, pp. 263–268. Vienna, Austria (2005)