Journal of Theoretical and Applied Information Technology
15th December 2011. Vol. 34 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

BIONOMIC ALGORITHM FOR SHORTEST PATH FIRST

AZAM KHAN, SAIRA BEG, FARAZ AHSAN, SAJJAD MOHSIN

COMSATS Institute of Information Technology, Islamabad, Pakistan

Email: {(azam_khan, saira.beg, fahsan, smohsin)@comsats.edu.pk}

ABSTRACT

In this paper we propose the idea of using Bionomic Algorithm (BA) for the shortest path finding problem. SPF is a very important issue associated with many other tasks like network routing and graph/map analysis etc. Different techniques have been used for solving the SPF problem, some of them being Dijkstra Algorithm, Floyd Algorithm, Genetic Algorithm, Tabu Search etc. The choice of algorithm for any specific problem depends upon the complex tradeoffs between complexity, performance etc. Our aim is to use type 3 heuristics algorithm for SPF. Simulations are carried out for BA algorithm and Dijkstra algorithm using MATLAB.

Keywords: *Shortest Path Algorithm (SPF), Bionomic Algorithm (BA).*

1. INTRODUCTION

Shortest Path Finding (SPF) is an important problem in graph theory and it has many applications in communications, transportation, and electronics. For solving the SPF problem, there are multiple algorithms such as Floyd [1], Dijkstra [1], Genetic Algorithm (GA) [2-3], Tabu Search (TS) [4], Ant Colony Optimization (ACO) [5] etc. Dijkstra Algorithm is widely used because of its simplicity. It is based on bellman optimization theory. It finds the shortest path from the source node to all other nodes in the network, for which it requires global topological knowledge of the network (list of all nodes, interconnections, and costs). It computes the shortest path as a function of distance, cost, traffic etc. One major drawback of Dijkstra algorithm is that it consumes more processing time (computation cost) if network is large [2].

[3] discusses the contribution of Genetic Algorithm (GA) in SPF. One major concern when applying GA for solving such problem is how to encode a path in a graph into chromosomes. Here, priority based encoding scheme is used. GA is difficult to implement and also its running time is greater than that of Dijkstra algorithm and it increases with an increase in the size of the network. But, GA provides multiple optimum candidate solutions with optimized paths (as opposed to Dijkstra algorithm which gives only one solution, though the best one, if it becomes unavailable somehow, we don't have any second option and we get stuck right there); this makes GA a promising algorithm which can be used with other algorithms in different applications [2]. GA belongs to type 2 heuristics which mainly focus on heuristic guidance and these are commonly known as meta-heuristics [6].

Type 2(heuristics based algorithms), when used for a specific problem, cannot give good results because they cannot be generalized. Bionomic Algorithm (BA) and Scatter Search (SS) are type 3 heuristics which focus on mathematical programming contribution. In this paper we have used Bionomic Algorithm (BA) for solving the SPF problem. Our aim is to give an alternative solution for this classical problem. Section 2 discusses the SPF problem and existing solutions' overview.Section 3 explains Dijkstra algorithm for the shortest path finding. Section 4 presents BA algorithm and its applications. Section 5 contains our proposed algorithm with respect to the shortest path finding problem and section 6 presents the simulation results and conclusion.

2. SHORTEST PATH FINDING (SPF)

SPF is basically a problem of finding the shortest path between any two vertices Vi and Vj on an undirected graph G, where E is an edge (a weighted arc) between any two nodes on the graph G. Weight Wij of an edge is a non-negative value that represents the cost of a particular edge between the nodes Vi and Vj. Path between two nodes can be represented either by a sequence of arcs (*ai*, *aj*) … (*an*, *ak*) or by a sequence of nodes (vi, va …, vn,

vk) [3]. Shortest Path Finding problem may be considered as finding out the sequence of nodes (from a source node to a destination node) with the least cumulative weight.
[1] discusses the Floyd and Dijkstra algorithms.Choice of algorithm for specific problem is based on the complex tradeoffs between complexity, performance and cost etc. SPF is an important problem, and has many applications in different areas like Robotics, communication networks and specially in routing etc.

The application of SPF in routing is presented in [4-5, 9]. [4] discusses the utilization of Tabu Search (TS) for finding shortest path in packet network. In their simulation, they used 5, 8 node network having 8, 12 links respectively, with random cost for each link. Simulation results show that this algorithm provides a quality solution with nominal computational cost. [5] is about a hybrid algorithm for routing which combines Ant Colony Optimization (ACO) and Tabu Search (TS) with the major aim of findingthe shortest path including the blind alley. In routing searching problem, the moving destination has been limited unlike the Traveling Salesman Problem (TSP). So, there is a chance that Ant agent can be trapped in blind alley. To overcome this problem, they have combined Tabu Search and ACO. In their experiment they used 8 different kinds of starting and destination points having 56 routes in total. Results show a significant advantage over Dijkstra algorithm. [9] Represents Genetic Algorithm (GA) for finding shortest path in dynamic network. They compared their results withDijkstra algorithm, and affirmed the potential ofproposed algorithm. [10] presents the solution of finding shortest path for road maps. Shortest path in this problem is calculated as smallest distance between two cities on a map. Algorithm's efficiency is tested for road map having 125 cities and results show a potential utilizing GA in such problems.

3. DIJKSTRA'S ALGORITHM

Dijkstra proposed an algorithm for shortest path finding in 1959, known as Dijkstra's Algorithm [13]. His proposed algorithm could be used for finding the shortest path from a single node (source) to one or multiple destination nodes on a graph with non-negative edge weights (costs). Dijkstra's Algorithm assigns cost values to every node which is the current lowest distance to that particular node from the source node. Initially, it assigns zero to source node and infinity to all other nodes. Source node is initially set to be a permanent node and its neighbors are set to be tentative nodes; costs of tentative nodes have a capacity to be improved later on while cost of a permanent node cannot be changed. Dijkstra's algorithm improves the distance (costs) of all the nodes in the network step by step. Let's consider the case of a small network of 6 nodes with path costs as shown in figure 1.

Let's consider node-1 to be the source node and node-5 to be the destination node.

a) Dijkstra's Algorithm starts with assigning zero value to source node (node-1 in this case) and making it a permanent node. It then assigns an infinity value to all other nodes.
b) It then examines all the nodes which are immediate neighbors to the node which was set as permanent in the last. For our considered example (figure 1), it examines nodes 2, 3 and 6 which are neighbors to node 1.
c) It assigns a cost to each of these nodes and sets them as tentative nodes. Like in case of the network under consideration, it will assign costs 7, 9 and 14 to nodes 2, 3 and 6 respectively.
d) It then selects the node with the smallest cumulative cost among all the tentative nodes and sets it as a permanent node. Like in case of the network we have considered here, it will select node 2 as a permanent node because it has the smallest cumulative cost among all the tentative nodes.
e) It then checks all the nodes in the network which can be reached from the current permanent node without revisiting any node in the current path. For example, nodes 3 and 4 can be reached from the current permanent node (node 2). Though node 1 is also a neighbor to node 2 but it cannot be reached from node 2 because this action causes a revisit (making the current path as 1-2-1) which is not allowed.
f) It computes the cumulative distance (cost) of all the nodes which can be reached through the current permanent node. If the existing cumulative cost of the nodes reached is greater than the computed cost, it is replaced with the computed cost. Also, the nodes reached are set to be tentative nodes. For example, existing cost of node 3 is 9 and the computed cost (through node 2) is 17, so it will leave its existing cost unchanged. But, the computed cost of node 4 is 22, which is lesser then its existing cost (infinity), so it will be replaced.

g) Steps 2 to 5 are repeated until all the nodes become permanent.

4. BIONOMIC ALGORITHM (BA) AND ITSAPPLICATIONS

Bionomic algorithm was proposed by N. Christofides in 1994 [11]. BA follows the GA framework, but replaces randomness of GA operators with normative procedure. It also accepts multiple parents'combinations and variable cardinality solution sets [6]. Just like GA or other Evolution Strategies (ES), it updates a whole population of solutions at each iteration,where updating means defining a child solution from a set of parents (crossover). GA is limited to two parents, but BA can use multiple parents; an approach which is shared with scattered search. BA formally requires maturation (local optimization of the solution) [7]. A detailed generic framework of BA is present in [6-7]. Figure2 describes the flow of BA algorithm. It has five major steps; 1) initialization, 2) maturation, 3) propagation, definition of parents sets, 4) propagation, definition of child solutions and 5) termination. Procedure for maturation step could be bare steepest descent but TS or other type 2 heuristics could be used as well depending upon the underlying problem.

Bionomic Algorithm's framework can be represented as follows:

h) Initial population generation.
i) Maturation (improving each individual locally).
j) Representing the population structure on a graph and using this graph to compute a set of parents.
k) Producing a set of new individuals by the linear combination of parents.
l) Maturation (local improvement of the individuals).
m) Updating the solution set (population)
n) Repeating step-3 to step-6 until a predefined optimal solution is reached or a maximum number of iterations has been done.

The structure of BA (specially a structured procedure for parent selection) makes it suitable for combinatorial optimization problems where as GA hardly describes the structure and properties of the solution, and also the Evolutionary Strategy (ES) is ill adapted because it works on continuous spaces. Example of BA algorithm utilization is capacitated p-Median Problem (CPMP) [7]. They attest the effectiveness of algorithm on standard data sets from the literature and on more difficult symmetric and asymmetric cost instances, and found satisfactory results.
Another application of BA is scheduling. [8] describes the utilization of BA for air craft landing problem. Their results suggest that it can produce good quality feasible results in short time. They conduct their experiments on publicly available test problems involving up to 500 aircraft and five runways.

5. PROPOSED ALGORITHM

For explanation purpose, consider the network of 6 nodes (n1, n2, n3, n4, n5 and n6) as shown in the figure 1. The shortest path finding problem calls for computing a sequence of nodes representing the shortest path- a path with the lowest cumulative cost- from a source node to a destination node. Consider the following assumptions:
Source node (s) = n1
Destination node (d) = n5

Then, one of the possible paths between these two nodes is:

Possible sequence= n1-n3-n2-n4-n5

keeping in mind that none of the nodes can be revisited and that the sequence must start at the source node and end at the destination node. Maximum number of nodes present in a path should be equal to the total number of nodes present in the network as it can never exceed this limit and also for the worst case, a path may contain all the nodes present in the network. Cost of any possible path with *u* number of nodes may be computed as follows:

$$W = \sum_{t=1}^{u} (n_t) \qquad (1)$$

The bionomic algorithm that we have proposed for solving the shortest path finding problem starts with generating a random initial population of solutions (sequences representing the path). Each individual of a population in genetic algorithm and bionomic algorithm is termed as a chromosome. So, a chromosome in this particular case will be a representation of a path like n1-n3-n2-n4-n5. A chromosome in evolutionary algorithms can be represented either as a binary string or as a simple decimal numbers string representing a possible solution. For simplicity of using the decimal representation and inadequacy of using binary representation in this particular case, we have used the decimal representation of chromosomes. Fitness of a solution is said to be the measure of feasibility/optimality of the solution, which may be computed based on the values of all different feasibility variables counted for the underlying problem. Fitness of chromosomes in this particular problem will simply be shortness of the path. So, for a population of *i* number of solutions, it can be computed using the following equation:

$$\mathbf{fitness(i)} = \frac{1}{W_i} = \frac{1}{\sum_{t=1}^{u}(n_t)}$$

(2)

Consider the case of the path n1-n3-n2-n4-n5. If it is one of the solutions, its cumulative cost comes out to be 40. So, its fitness value comes out to be 1/40.

Second step in bionomic algorithm is the maturation step which is to be applied to all the individuals of initial population as well as to those of each subsequent generation. Maturation is one of the attributes which distinguish bionomic algorithm from genetic algorithm. Maturation demands local improvement of the individuals based on the problem specific knowledge. There may be lots of maturation procedures; we adopt one based on our objective. The improvement procedure which we have considered is to limit the individual production to possible solutions only. For this purpose, we have imposed two restrictions on the individuals 1) they will start from the source node and reach the destination node and every next node coming in the path needs to be from among those having a connection with the current node. 2) There will be no repetition of nodes i.e. a path will not pass through one particular node more than once.

Table 1: Problem specific knowledge for the 6 nodes network

	Destination Nodes					
Source Nodes	0	7	9	0	0	14
	7	0	10	15	0	0
	9	10	0	11	0	2
	0	15	11	0	6	0
	0	0	0	6	0	9
	14	0	2	0	9	0

Next phase is parent selection for reproduction. Parent selection in bionomic algorithm is a structured procedure. It gives more opportunity to the individuals with greater fitness and also for one particular parents set, it selects parents which are not too close. This is achieved by the techniques of adjacency graph and inclusion frequency. Individuals with greater frequency have a greater inclusion frequency which decreases every time an individual is selected as a parent. Adjacency graph is used to ensure that the parents selected for one particular parent set are far apart. Every individual (solution/chromosome) of a population is represented by a node on the graph. Edge between any two nodes of the graph depends on the distance between those two nodes (individuals). That distance can be measured in different ways based on the underlying problem. Our method of calculating distance is to find similarity between two individuals. Like, consider one of the individuals be:

n1-n3-n2-n4-n5

and the other individual be:

n1-n3-n4-n5

In our example, any two parents of a set need to have at least three nodes in common; the first node, the last node and any other node in between which may be used for crossover later on. So, any two parents having no more than 3 nodes in common will have the minimum similarity and hence the maximum distance from each other. Similarity between any two individuals can be measured using the following equation:

$$similarity = \frac{no.\,of\ common\ nodes}{total\ no.\,of\ nodes\ in\ the\ shorter\ path}$$

(3)

$$distance = \frac{\mathbf{total\ nodes\ of\ shorter\ path - common\ nodes}}{\mathbf{total\ no\ of\ nodes\ in\ the\ shorter\ path - 3}} \quad (4)$$

A certain distance is set as a threshold value to decide whether or not to put an edge between any two nodes. This particular step helps us reasonably spread the parents (belonging to one parents' set) over the whole problem search space. Also, the concept of inclusion frequency helps giving more opportunity to the chromosomes which have a higher fitness value.

Crossover and mutation follow the parent selection phase. In bionomic algorithm, more than 2 parents can mate to produce a child. One parents set produces one child. For our simulation purpose, we have used 2 parents for crossover. Mutation is dealt with differently for different problems. Even for the same problem like SPF, it can be accomplished in more than one ways. In our simulations, we have used to flip a path segment between two specific nodes from one possible path to another possible path. For example, the individualn1-n3-n4-n5, if selected for mutation, can be mutated by altering n3 with n2 (n1-n2-n4-n5) because it in no way makes it an inapplicable combination. Mutation site should be selected randomly, because there may be more than one site where alternate paths are possible. Maturation procedure is again applied to the new individuals and their respective fitness values are evaluated. After applying maturation and assessing the fitness of a child it is included to the new population. After having done with all these things, older generation is replaced with the new one. Bionomic algorithm uses the generational approach for population replacement [7, 14]; whole of the new generation replaces the older generation instead of doing that every time a new child is created.

6. RESULTS AND DISCUSSION

Dijkstra algorithm and bionomic algorithm for solving the shortest path finding problem were implemented in Matlab. We have considered two cases for finding out their optimum solutions using both the Dijkstra algorithm and the Bionomic algorithm. Now, we consider both of the cases for both of the algorithms and discuss their results.

Case-I

For case-I, we consider the 6 nodes network as shown in the figure 1 above. Consider node 1 to be the source node and node 5 to be the destination node.

Table 2: Results of Dijkstra Algorithm (case-I)

Source	Destination	Shortest Path	Cost	Time (sec)
1	5	n1-n3-n6-n5	20	0.0293

Dijkstra's algorithm that we implemented computes the shortest paths from the source node to every other node in the network. So, the computation time for computing path from node 1 to any other node in the network will be the same. It is to be noted that Dijkstra's algorithm is called greedy algorithm because it is not concerned with an optimum solution, it rather finds the best path by any mean.

As mentioned earlier that BA follows the GA framework for the most part; this implies that basic parameters will be the same as for GA. Such parameters are mutation rate, cross-over, population size and number of generation.

Table 3: parameters for Bionomic Algorithm

Parameter	Value
Max. Population Size	Equal to number of nodes
Mutation Rate	0.2
Cross-over	0.9
Number of Generation	12

Table 4: Results of bionomic algorithm (case-I)

Source	Destinati	Path	Cost	Time (sec.)	Generation
1	5	n1-n2-n3-n6-n5	28	0.1640	2
		n1-n6-n5	23	0.1653	4
		n1-n3-n6-n5	20	0.1665	6
		n1-n3-n6-n5	20	0.1677	8

Table 4 shows the results of applying bionomic algorithm to case-I. After reaching the 8th generation, we can establish that the solution has converged and it is the most optimum result that we can find.

Case-II

For case-II, we consider the 12 nodes network as shown in figure 3. Consider node 6 to be the source node and node 12 to be the destination node. Table 6 shows the results of applying bionomic algorithm to the network. Parameters of bionomic algorithm considered here are the same as they were for case-I. At 12th generation, we can stop and establish that the solutions have converged.

Table 5: Results of applying Dijkstra's algorithm (case-II)

Source	Destination	Shortest Path	Cost	Time (sec.)
6	12	n6-n11-n7-n10-n12	11	0.0301

Table 6: Results of bionomic algorithm (case-II)

Source	Destination	Path	Cost	Time (sec.)	Generation
6	12	n6-n2- n7-n3-n10-n12	20	0.1717	3
		n6-n11-n7-n10-n12	11	0.1728	6
		n6-n11-n7-n10-n12	11	0.1737	9
		n6-n11-n7-n10-n12	11	0.1746	12

Simulation results show that the shortest path finding problem can be solved successfully by applying bionomic algorithm with a good degree of accuracy. It is worth noticing that the optimum results found in both cases are exactly the same as found using Dijkstra’s algorithm. Our bionomic algorithm can work fine most of the time for the networks of average complexity and it works satisfactory even for very large and complex cases though it takes more time to converge in a complex and larger situation.

Figure 4 shows the convergence behavior for shortest path finding problem in terms of both Dijkstra and Bionomic algorithms. These results are taken for case-1 (6-nodes network) and highlights that the former gives optimum results quite earlier than latter, for a smaller network. However, it provides with only one solution for the given problem, whereas bionomic algorithm provides with multiple results; thus enables diversity for practical purposes.

7. CONCLUSION

The shortest path finding problem can be efficiently solved using bionomic algorithm which has already been used successfully for solving some other optimization and searching problems. It has certain advantages over Dijkstra algorithm and other evolutionary algorithms, like it caters better for avoiding local optima. In future, we plan to assess this algorithm on some other parameters and also to apply it to other optimization problems of practical value.

REFERENCES

[1]. http://www.mcs.anl.gov/~itf/dbpp/text/node35.html

[2]. Ehab Rushdy Mohamed et al. “Performance Evaluation of Genetic Algorithm for solving Routing Problem in Communication Network”, International Journal of Computer Science and Information Security, pp 37-43, Vol 8 Number 3, 2010. Online Available: http://www.scribd.com/doc/33949199/Performance-Evaluation-of-Genetic-Algorithm-For-Solving-Routing-Problem-In-Communication-Networks

[3]. Genetic Algorithms for Solving Shortest path Problems by Mitsuo Gent, Runwei Cheng, and Dingwei Wang, IEE Xplore, pp 401-406, 1997.

[4]. Anant Oonsivilai , Wichai Srisuruk , Boonruang Marungsri ,Thanatchai Kulworawanichpong “Tabu Search Approach to Solve Routing Issues in Communication Networks ”, World Academy of Science, Engineering and Technology, vol 53, pp 1174-1177, 2009.

[5]. Masaya Yoshikawa, Kazuo Otani “Ant Colony Optimization Routing Algorithm with Tabu Search”, Proceeding of the International MultiConference of Engineers and Computer Scientists (IMECS), Vol 3, Hong Kong, 17-19 March 2010.

[6]. Vittorio Maniezzo, Antonella Carbonaro “Ant Colony Optimization: an overview”, Scienze dell’ informazione , University of Bologna, Itlay, 1999.

[7]. Vittorio Maniezzo, Aristide Mingozzi , Roberto Baldacci “A Bionomic Approach to the Capacitated *p*-Median Problem”, Journal of Heuristics, Vol 4, pp 263–280, Kluwer Academic Publishers, 1998.

[8]. H. Pinol a, J.E. Beasley, “Scatter Search and Bionomic Algorithms for the aircraft landing problem”, European Journal of Operational Research 171 (2006), pp 439–462, 2004.

[9]. Gihan Nagib and Wahied G. Ali, “Network Routing Protocol using Genetic Algorithms”, International Journal of Electrical & Computer Sciences IJECS-IJENS Vol: 10 No: 02, pp 40-44, 2010.

[10]. Sachith Abeysundara, Baladasan Giritharan, Saluka Kodithuwakku, “A Genetic Algorithm Approach to Solve the Shortest Path Problem for Road Maps”, Proceedings of the

International Conference on Information and Automation, pp 272-275, December 15-18, 2005, Colombo, Sri Lanka

[11]. N. Christofides. "The Bionomic Algorithm", AIRO'94 Conference, Servona, Italy, 1994.

[12]. Faraz Ahsen, Khalid Hussain, Nyla Khadam, Muhammad Sharif and Noor Zaman, "Identification of a Lossy Channel in Wireless Mesh Network using Conservation of flow", Journal of Information & Communication Technology, Vol. 1, No. 2, pp 60-70, 2007.

[13]. E. W. Dijkstra. "A note on two problem in connexion with graphs", Numerische Mathematik, pp 269-271, 1959.

[14]. N. Christofides. "Optimal design of telecommunication networks", CO96 Conference, Imperial College, London, 1996.

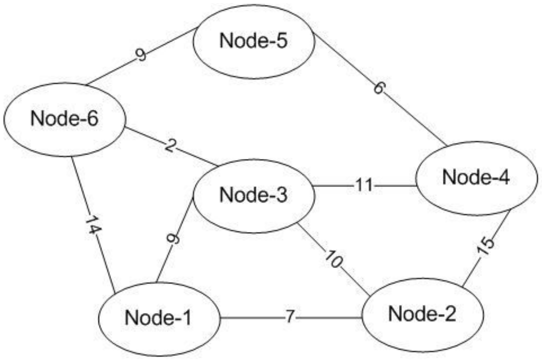

Figure 1: 6 nodes network [9]

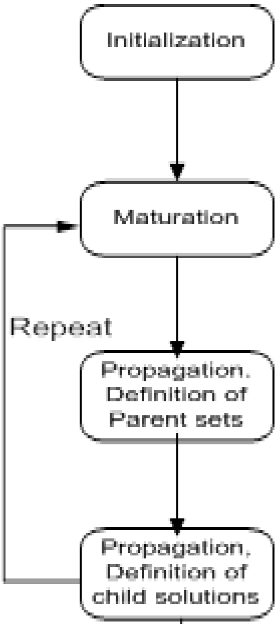
Initialization
Maturation
Repeat
Propagation, Definition of Parent sets
Propagation, Definition of child solutions

Figure 2: Basic structure of Bionomic Algorithm

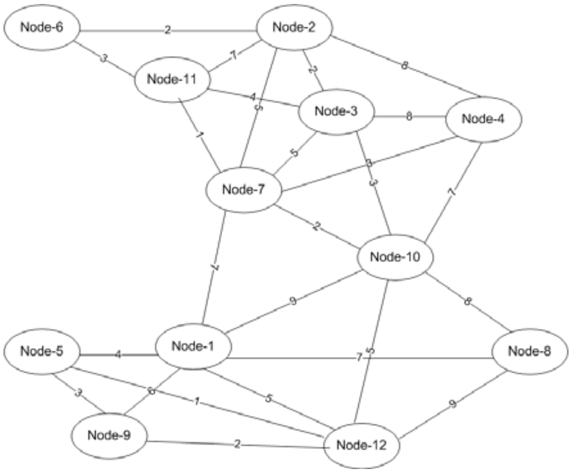

Figure 3: 12 node network

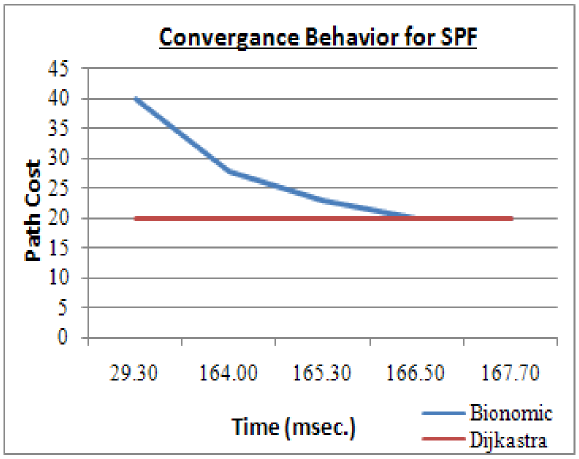

Figure 4: Convergence behavior for SPF