International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-3 Number-4 Issue-13 December-2013

Search the Set of Shorter Paths Using Graph Reduction Technique

Tarak Nath Paul[1], Abhoy Chand Mondal[2]

Abstract

The beauty of this paper is to clip a Graph provided by Geographic Information System (GIS) into levels (called Levelled Graph), and find the set of shorter paths with source at the very beginning level and destination node as the leaf nodes in the above level. This paper will formulate an algorithm which will provide other shorter paths from source to destination and with the help of Genetic Algorithm (GA) the proposed algorithm is verified. The graph may represent network lines for transferring packets, pipeline to transfer liquid and it may be transportation lines. The proposed algorithm discovers the shortest path and other shorter paths from source to destination with lesser traffic. The selection of node is done using GIS because it is capable enough to express the connectivity of node with one another. The proposed algorithm is compared with Dijksra's Algorithm and the results are satisfactory. Simulated results are formulated using Matlab. The result assures the potential of the Algorithm.

Keywords

Geospatial Information System, Shortest Path Algorithm, Routing, Graph, Genetic Algorithm, Chromosome, Mutation and Fitness Criteria.

1. Introduction

The idea of this hybrid algorithm is implemented with the help of graph, levelling of graph and genetic algorithm. Checking and testing is based upon the ideas of genetic algorithm [3]. The possible paths are represented by chromosomes and possible paths are achieved by fussing them. Dijkstra's Algorithm finds the paths by weight precedence. It solves the single source shortest path issue [7] when the weights are non-negative values. It is also a greedy algorithm and also similar to Prim's algorithm. The algorithm starts at the source node and a tree grows which reaches to all the nodes describing the shortest path. [2][6][9]

Tarak Nath Paul, Computer Science Department, The University of Burdwan, Burdwan, West Bengal, India.

Abhoy Chand Mondal, Computer Science Department, The University of Burdwan, Burdwan, West Bengal, India.

In the field of artificial intelligence genetic algorithm is a heuristic search algorithm, works of evolution of the individuals. Genetic algorithm is an evolutionary algorithm whose solution solves problem using natural selection, inheritance, crossover, mutation, selection and termination operations using fitness functions. It combines survival for the fittest (Law given by Charles Darwin). The primary monograph on the topic is Holland's (1975) Adaptation in Natural and Artificial Systems. A population undergo selection including operators such as Selection (Based on Misfit), Cross-over (Swapping Information) and Mutation (Change in Individual). A fitness function is used to evaluate individuals. Based on the fitness function the best individuals are fussed together and the new populations are applied to the algorithm until an optimal solution is obtained. [1][5][6]

2. Dijkstra's Algorithm

Dijkstra's algorithm uses the greedy approach to figure out the single source shortest path problem. This problem is associated with the spanning tree. There will be no cycles found as. It repeatedly takes from the unselected vertices v to nearest source s and declares the distance to be the actual shortest distance from s to v. The edges of v are then examined to see if their destination can be reached by v followed by the relevant outgoing edges. G = (V, E), Where V – is a set of vertices; E is a set of edges. [4][6][9]

Dijkstra's algorithm keeps two sets of vertices:
S → the set of vertices whose shortest paths form the source have already been determined.
V – S → the remaining vertices.
The other data structures needed are:
d – array of best estimates of shortest path to each vertex.
pi – an array of predecessors of each vertex.
The basic mode of operation is:

1. Initialize d and pi,
2. Set S to empty,
3. While there are still vertices in V-S,
   i. Sort the vertices in V-S according to the current best estimate of their distance from the source,
   ii. Add u, the closest vertex in V-S, to S,
   iii. Relax all the vertices still in V-S connected to u

DIJKSTRA (G, w, s)
1. INITIALIZE SINGLE-SOURCE (G, s)
2. S ← { } // S will ultimately contains vertices of final shortest-path weights from s
3. Initialize priority queue Q i.e., Q ← V[G]
4. while priority queue Q is not empty do
5. u ← EXTRACT_MIN(Q) // Pull out new vertex
6. S ← S È {u} // Perform relaxation for each vertex v adjacent to u
7. for each vertex v in Adj[u] do
8. Relax (u, v, w)

3. Genetic Algorithm

Genetic Algorithms are the heuristic search and optimization procedures which takes the idea from natural evolution. Principle "Select the Best, Discard the Rest".

The implementation of genetic algorithms is as follows:

1. Initialization: At the very beginning numerous individual solutions are randomly selected to form an initial population.
2. Selection: Individual solutions are picked out according to the fitness function.
3. Reproduction: Crossover, Mutation and Selection methods.
4. Termination: The process continues until a termination condition is identified. [8][12][13]

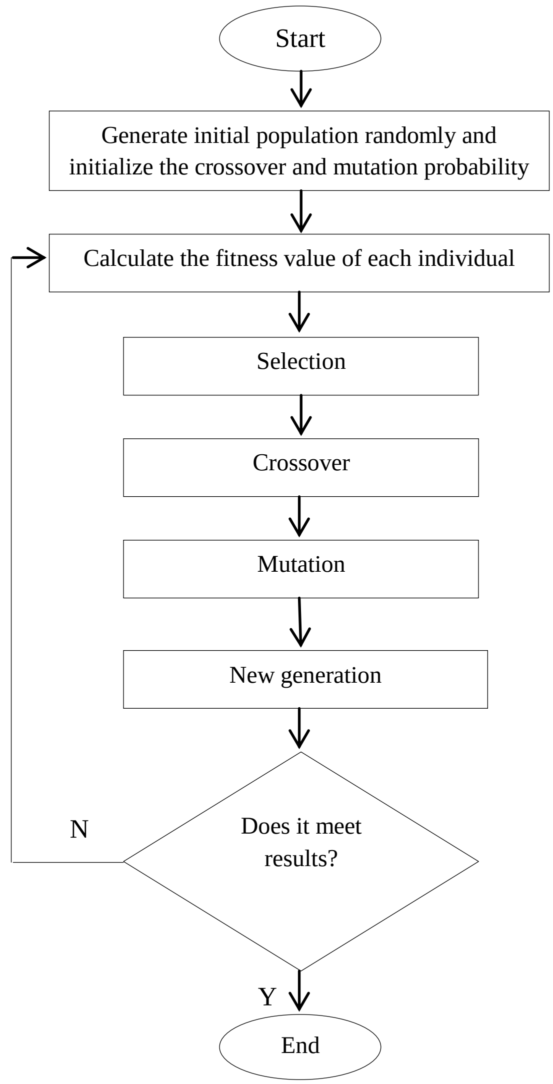

Figure 1: Flow Chart for Genetic Algorithm

4. Geospatial Information System

Geographical Information System (GIS) [8] use geospatial analysis. It is a growing technology paradigm, which is a powerful tool to solve complex problem in special environment in tabular presentation. GIS provide geographic data of any location successfully with its altitude, height, location and connectivity etc. The potential of GIS is beyond vision. Geospatial analysis works beyond 2D mapping operations and spatial analysis. It involves:

1) ***Surface Analysis*:** Analyzing the properties of physical surfaces (gradient, aspect and visibility).

2) ***Network Analysis*:** Analyzing the attributes of natural and man-made networks in order to understand the behavior of flows within and around the network and location. This may be used to address a wide range of practical problems-route selection, facility location and problems involving flows like those found in hydrology and transportation problem.
3) ***Geovisualization*:** The creation and manipulation of maps, images, charts, diagrams, 3D views and their associated data. This provides a range of tools, which provide static or rotating views, providing animations, dynamic linking and brushing and spatio-temporal visualizations etc. This tool is the least developed and reflects limited range of suitable and compatible datasets in which the limited set of analytical methods are available. All these facilities increases the core tools utilized in spatial analysis throughout the patterns and relationships, construction of models and communicating the results.[10][11]

5. Proposed Algorithm

Let us consider an arbitrary graph of n (=21) number of nodes (Cities / Connectors / Distribution points) from the real world co – ordinates (Figure 1) with the help of GIS. Consider the real time data about their location and connectivity between the nodes. Geospatial Information System (GIS) will provide the location of the city and the distance (weight) between the nodes. If the nodes are cities then it will be on land, if the nodes are connector for internet wire then it can be very deep under water, if the nodes are of distribution point for liquid transportation then it can also be very deep under water. This paper has taken a generalized selection of nodes which is under water and on land. Connection and Distance between nodes of Real World Geospatial Space is shown in Figure 2 & 3 and tabular representation in Table 1.

Let Graph G = {V, E} where V = {A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, V} and E = {{A,B}, {A,C}, {B,E}, {B,D}, {C,D}, {C,F}, {C,G}, {D,E}, {D,F}, {E,I}, {E,H}, {F,H}, {F,G}, {G,H}, {G,K}, {G,Q}, {H,I}, {H,J}, {H,K}, {I,O}, {I,V}, {I,J}, {J,V}, {J,L}, {J,K}, {K,L}, {K,M}, {L,N}, {L,M}, {V,N}, {M,P}, {M,Q}, {N,O}, {N,R}, {N,P}, {O,R}, {O,P}, {P,R}, {P,T}, {P,S}, {P,Q}, {Q,S}, {R,T}, {S,T}}

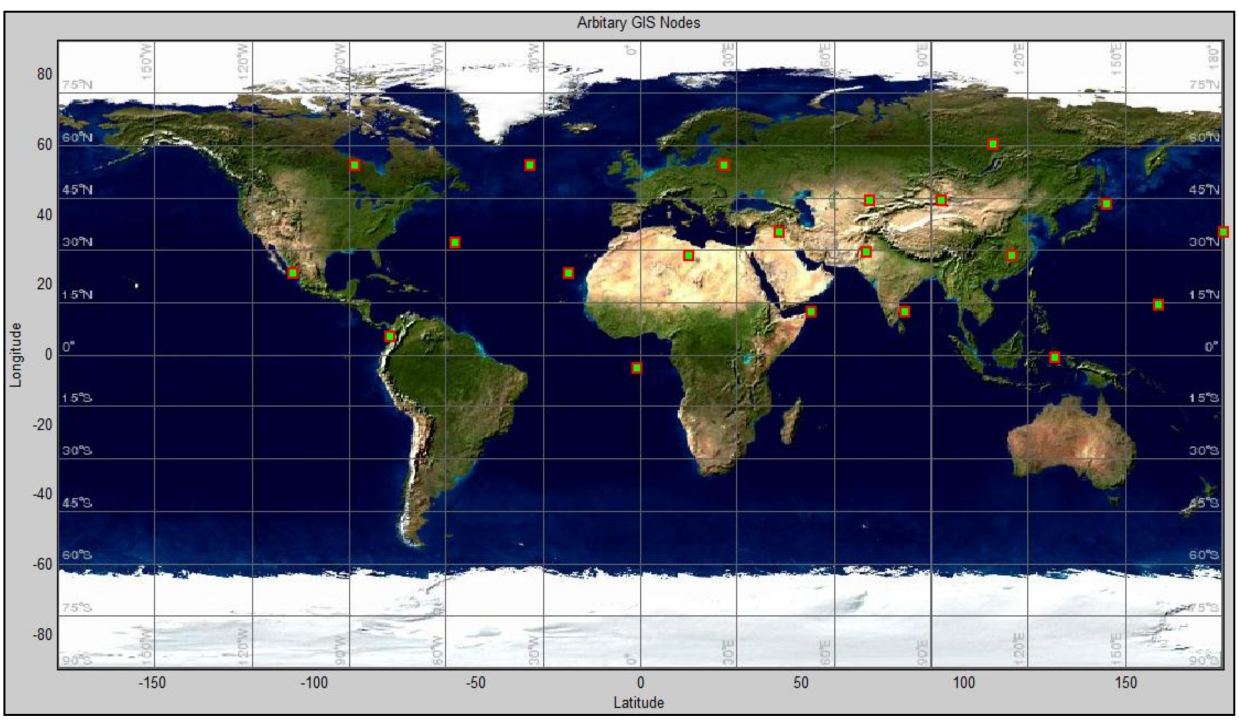

Figure 2: Real Geospatial Nodes are Considered

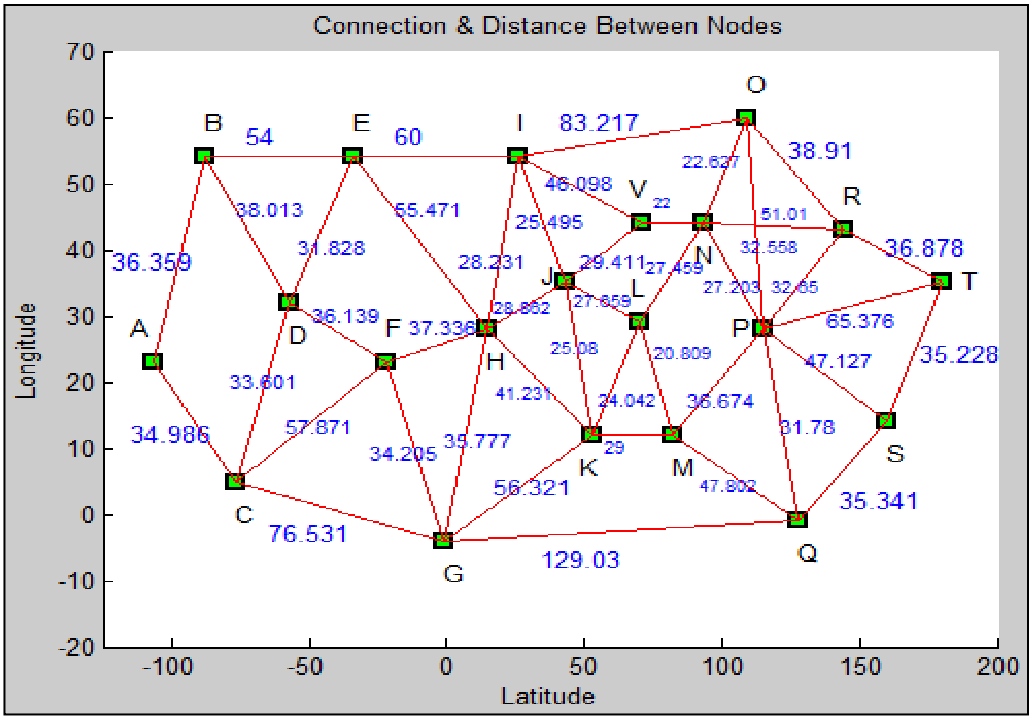

Figure 3: Connection and Distance between Nodes found in Real Geospatial Space (Main Graph).

Table 1: Representation of Connection and Distance between Nodes in Tabular Format.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	V
A	0	36.3593	34.9857	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	36.3593	0	0	38.0132	54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	34.9857	0	0	33.6006	0	57.8705	76.531	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	38.0132	33.6006	0	31.8277	36.1386	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	54	0	31.8277	0	0	0	55.4707	60	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	57.8705	36.1386	0	0	34.2053	37.3363	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	76.531	0	0	34.2053	0	35.7771	0	0	56.3205	0	0	0	0	0	129.035	0	0	0	0
H	0	0	0	0	55.4707	37.3363	35.7771	0	28.2312	28.8617	41.2311	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	60	0	0	28.2312	0	25.4951	0	0	0	0	83.2166	0	0	0	0	0	46.0977
J	0	0	0	0	0	0	0	28.8617	25.4951	0	25.0799	27.6586	0	0	0	0	0	0	0	0	29.4109
K	0	0	0	0	0	0	56.3205	41.2311	0	25.0799	0	24.0416	29	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	27.6586	24.0416	0	20.8087	27.4591	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	29	20.8087	0	0	0	36.6742	47.8017	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	27.4591	0	0	22.6274	27.2029	0	51.0098	0	0	22
O	0	0	0	0	0	0	0	0	83.2166	0	0	0	0	22.6274	0	32.5576	0	38.9102	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	36.6742	27.2029	32.5576	0	31.7805	32.6497	47.1275	65.3758	0
Q	0	0	0	0	0	0	129.035	0	0	0	0	0	47.8017	0	0	31.7805	0	0	35.3412	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	51.0098	38.9102	32.6497	0	0	0	36.8782	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47.1275	35.3412	0	0	29	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	65.3758	0	36.8782	29	0	0
V	0	0	0	0	0	0	0	0	46.0977	29.4109	0	0	0	22	0	0	0	0	0	0	0

If the count of the nodes decreased then the execution time will also decrease. And the search space is also reducing for the shortest and shorter paths. Procedure to create the Reduced Graph - Select those nodes of whose x-values lies between the x-values of the source and destination and whose magnitude is less than the magnitude of the source and the destination nodes, shown in Figure 4 followed by Algorithm to create Sub - Graph from Main - Graph.

<u>Algorithm for Creation of Sub – Graph from the Main – Graph</u>

Step 1: Choose Source Node. [TSource = Source; TSource stands for Temporary Source]
Step 2: Create First Level.
Step 3: Place TSource in the Level.
Step 4: Find connecting nodes to TSource and Add it to Queue.
Step 5: If all Nodes of the Same Level not Traversed.
        TSource = un – Traversed node in the same Level.
        Goto Step 6.
    else
        Goto Step 3
Step 6: Create the next Level (Level++).
Step 7: Add the Queue to the Level.
Step 8: If Destination Found && Every Nodes is Traversed in the same Level
        Goto Step 9
    else
        TSource = Next un – Traversed node in the Same level.
        Goto Step 4.
Step 9: End.

The algorithm that is discussed above (Figure 7 Flow Chart of same), TSource is considered as Temporary Source. Firstly our source from where we need to travel is selected and then the destination to where we need to reach is considered. Now creating a level is important. Then we place the source in the level and these nodes in the next level will be connected to the source node. Now we will move down to the next level and will do the same with every node in the level. The process will continue until the destination is reached. [14][15][16]

The essence of the work is that it reduces the graph in levels (Figure 5 Levelled Graph) and the first level and the last level will contain only the source vertex and destination vertex respectively. As “Algorithm for creation of Sub – Graph from Main Graph” gives birth to levelled Graph whose end leafs are destination node. Any traverse from first level (Source Node) to any of the leaf will be a path form source to destination, shown in Figure 6.

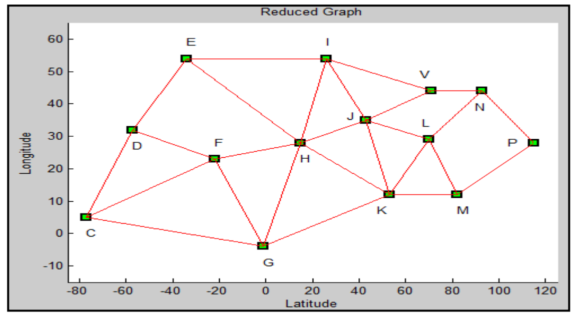

Figure 4: Reduced Graph.

The resultant is a reduced Levelled Graph which contains the shortest and the shorter paths in it. This reduces the number of vertex traversal increasing efficiency of the work done.

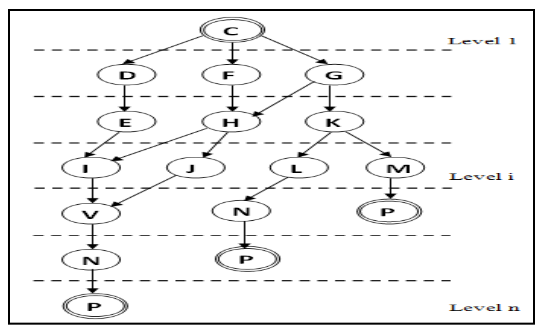

Figure 5: Formation of Levelled Graph.

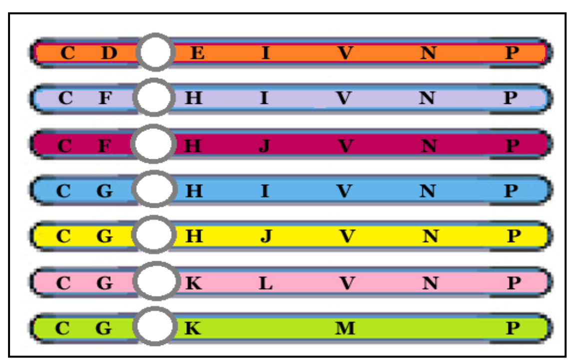

Figure 6: Representation of Possible Paths as Chromosomes.

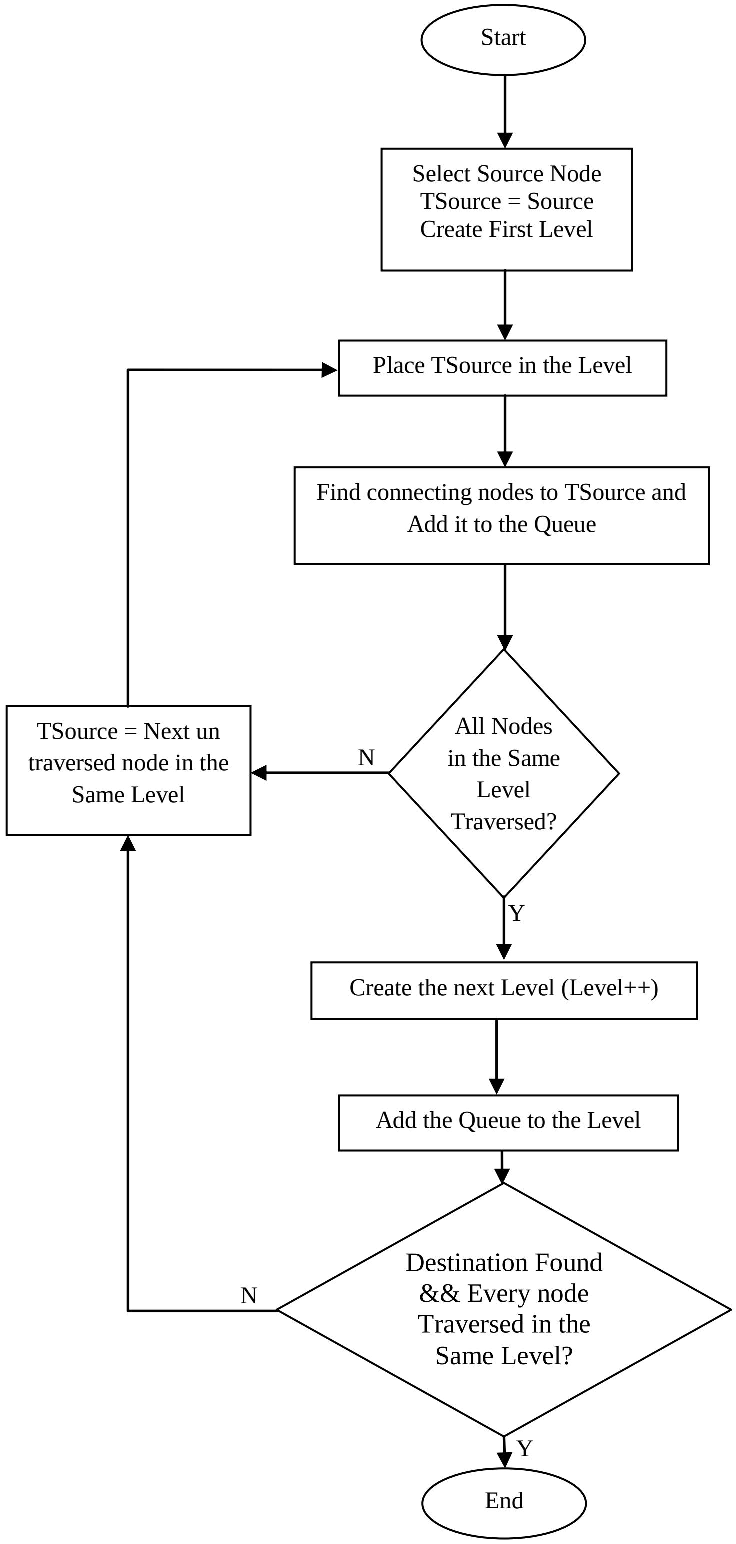

Figure 7: Flow Chart Representation of Creation of Sub-Graph from Main-Graph Algorithm.

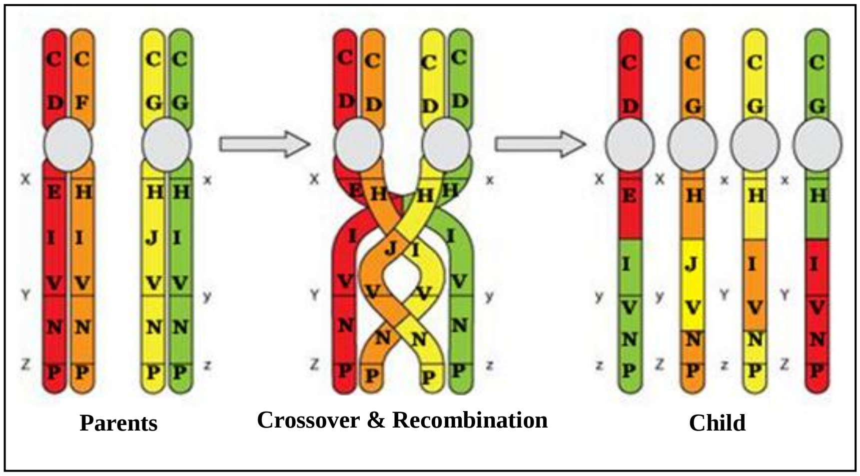

Figure 8: Fusion of Chromosome.

Table 2: Dijkstra's Shortest Path Algorithm

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	V
C	34.986	∞	0	33.601	∞	57.871	76.531	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
D	34.986	71.615	0	33.601	65.429	57.871	76.531	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
A	34.986	71.345	0	33.601	65.429	57.871	76.531	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
F	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
E	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
B	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
G	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	∞	132.852	∞	∞	∞	∞	∞	205.561	∞	∞	∞	
H	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	124.069	132.852	∞	∞	∞	∞	∞	205.561	∞	∞	∞	
I	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	124.069	132.852	∞	∞	∞	206.655	∞	205.561	∞	∞	∞	179.536
J	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	124.069	132.852	150.528	∞	∞	206.655	∞	205.561	∞	∞	∞	153.48
K	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	124.069	132.852	150.528	161.852	∞	206.655	∞	205.561	∞	∞	∞	153.48
L	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	124.069	132.852	150.528	161.852	177.987	206.655	∞	205.561	∞	∞	∞	153.48
V	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	124.069	132.852	150.528	161.852	175.48	206.655	∞	205.561	∞	∞	∞	153.48
M	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	124.069	132.852	150.528	161.852	175.48	206.655	198.526	205.561	∞	∞	∞	153.48
N	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	124.069	132.852	150.528	161.852	175.48	198.107	198.526	205.561	∞	∞	∞	153.48
O	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	124.069	132.852	150.528	161.852	175.48	198.107	198.526	205.561	237.017	∞	∞	153.48
P	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	124.069	132.852	150.528	161.852	175.48	198.107	198.526	205.561	231.176	245.653	263.902	153.48
Q	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	124.069	132.852	150.528	161.852	175.48	198.107	198.526	205.561	231.176	240.902	263.902	153.48
R	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	124.069	132.852	150.528	161.852	175.48	198.107	198.526	205.561	231.176	240.902	263.902	153.48
S	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	124.069	132.852	150.528	161.852	175.48	198.107	198.526	205.561	231.176	240.902	263.902	153.48
T	34.986	71.345	0	33.601	65.429	57.871	76.531	95.207	125.429	124.069	132.852	150.528	161.852	175.48	198.107	198.526	205.561	231.176	240.902	263.902	153.48

6. Complexity

Taking the idea of the level we can reach the destination in a single level in a favourable case (destination is the nearest node in a series) and in the worst case we can reach the destination with n levels, n is maximum number of nodes in the graph (if all the nodes lies in a series). The complexity comes to O(n). If the problem require only a set of m nodes to reach to the destination then the complexity comes to O(m) because the worst case if all the m nodes are in a series, where m is always less than n.

Table 3: Comparison of the Proposed Algorithm with Dijkstra's Algorithm

Proposed Algorithm				**Dijksra'a Algorithm**	
Chromosome	**Path**	**Value**	**Discussion**	**Path**	**Value**
1	C-D-E-I-V-N-P	222.73	These are the possible feasible shorter paths from Source C to Destination P other than dijkstra's shortest path (Chromosome7)	C-G-K-M-P	198.526
2	C-F-H-I-V-N-P	220.739		Dijkstra's Algorithm only find the possible shortest path but do not find any other possible shorter path form source to destination so that it can be used in difficult or crisis situation.	
3	C-F-H-J-V-N-P	202.683			
4	C-G-H-I-V-N-P	273.84			
5	C-G-H-J-V-N-P	219.784			
6	C-G-K-L-N-P	211.556			
7	C-G-K-M-P	198.526			
Fusion of Chromosome shown in Figure : 8					
Parent1	C-D-E-I-V-N-P	222.73	Chromosome1		
Parent2	C-F-H-I-V-N-P	220.739	Chromosome2		
Parent3	C-F-H-J-V-N-P	202.683	Chromosome4		
Parent4	C-G-H-I-V-N-P	273.84	Chromosome5		
Child Chromosomes					
Child1	C-D-E-I-V-N-P	222.73	Chromosome1		
Child2	C-F-H-J-V-N-P	202.683	Chromosome3		
Child3	C-F-H-I-V-N-P	220.739	Twin Chromosome4		
Child4	C-F-H-I-V-N-P	220.739			

7. Comparative Studies

If compare the above example with Dijkstra's Algorithm, we receives equivalent results with a better prospect. Proposed algorithm produce the shortest path with other shorter paths options where as Dijkstra's produces only the shortest path, shown in Table 3.

Now, finding the shortest Route to reach P from C. According the dijkstra's algorithm we get the shortest path evaluated in Table 2 is:

[C – G – K – M – P] → 198.526

The above path is one of the elite chromosomes/individual provided by the proposed work. Dijkstra's algorithm provides only the shortest path, but do not provide any alternative shortest path to reach to the destination.

8. Conclusion and Future Work

The proposed algorithm finds the shortest path from source to destination and also finds alternative paths to reach destination from source. If there is congestion in the shortest path then it take more time, if we follow other path having less congestion then destination can be reached in much more less time, if the path is available or valid.

Run time decision is not dealt with the proposed algorithm, which can be called a mutation in the path. In my forthcoming work I will be dealing this problem.

Acknowledgment

I would like to pay my sincere gratitude towards my guide, Dr. A. C Mondal (Associate Professor & Head of Computer Science Department, Burdwan University), without his extended support, this presentation for National Conference would not have been possible.

I would like also like to convey my deep thanks to the department lecturers who had helped me in this work.

References

[1] Al Jadaan, Omar, Lakishmi Rajamani, and C. R. Rao. "Improved Selection Operator for GA." Journal of Theoretical & Applied Information Technology 4, no. 4 (2008).

[2] Nagib, Gihan, and Wahied G. Ali. "Network routing protocol using Genetic Algorithms." International Journal of Electrical & Computer Sciences IJECS-IJENS 10, no. 02 (2010): 40-44.

[3] Cheng, Hui, and Shengxiang Yang. "Genetic algorithms with elitism-based immigrants for dynamic shortest path problem in mobile ad hoc networks." In Evolutionary Computation, 2009. CEC'09. IEEE Congress on, pp. 3135-3140. IEEE, 2009.

[4] Pangilinan, José Maria A., and Gerrit JANSSENS. "Evolutionary algorithms for the multi-objective shortest path problem." (2007).
[5] Reddy, V. Purushotham, G. Michael, and M. Umamaheshwari. "Coarse-Grained ParallelGeneticAlgorithm to solve the Shortest Path Routing problem using Genetic operators." V. Purushotham Reddy et al./Indian Journal of Computer Science and Engineering (IJCSE) 2, no. 1 (2011).
[6] Li, Yinzhen, Ruichun He, and Yaohuang Guo. "Faster genetic algorithm for network paths." In The 6th International Symposium on Operations Research and Its Applications,(ISORA'06). August, pp. 8-2. 2006.
[7] Mukhef, Hayder A., Ekhlas M. Farhan, and Mohammed R. Jassim. "Generalized shortest path problem in uncertain environment based on PSO." Journal of Computer Science 4, no. 4 (2008): 349.
[8] Ismail, Rakip Karas, and Atila Umit. "A genetic algorithm approach for finding the shortest driving time on mobile devices." Scientific Research and Essays 6, no. 2 (2011): 394-405.
[9] Zhan, F. Benjamin. "Three fastest shortest path algorithms on real road networks: Data structures and procedures." Journal of geographic information and decision analysis 1, no. 1 (1997): 69-82.
[10] Naghibi, F. "Application GIS in Petroleum Industry." In M. Sc. Research Seminar, Dept. of Surveying and Geomatic Eng., Eng. Faculty, University of Tehran. 2002.
[11] Reddy, M. Narayana, and N. H. Rao. "Integrating geo-spatial information technologies and participatory methods in agricultural development." Current Science 96, no. 1 (2009): 23.
[12] Gonen, Bilal. "Genetic Algorithm finding the shortest path in Networks." Reno: University of Nevada (2006).
[13] Beaslev, D., David R. Bull, and Ralph R. Martin. "An overview of genetic algorithms: Part 2, research topics." University computing 15, no. 4 (1993).
[14] Tarak Nath Paul and Dr. Abhoy Chand Mondal, “Constitute A Sub-Graph With n-levels To Search A Set Of Shorter Paths Using Genetic Algorithm”, IEEE Internaitonal Conference on Intelligent Computing and Intelligent System, 3-0096-10943, Volume 3, 452-456, November 2011.
[15] Tarak Nath Paul and Dr. Abhoy Chand Mondal, “Faster Genetic Algorithm For Traverse Large Distance”, National Conference on Computing and System, 56-59, January 2010.
[16] Tarak Nath Paul and Dr. Abhoy Chand Mondal, “Intelligent Traversing Method Using Genetic Algorithm For Long Distance”, International Conference on Computing and System, 69-74, November 2010.

Tarak Nath Paul born in 1984 Adra, Purulia, West Bengal, India pursuing his PhD. from The University of Burdwan, Burdwan, West Bengal. He has received his MCA (West Bengal University of Technology) and M. Tech in IT (Karnataka State Open University) with star marks and currently as Assistant Professor in Maharana Group of Institution (MPGI), CSE Department. He is a reviewer of IEEE Journals. His research area in Shortest Path with Genetic Algorithm, on which he has presented a paper in 2011 IEEE International Conference on Intelligent Computing and Intelligent System (ICIS-2011) at Guandong University, Guangzhou, China.

Abhoy Chand Mondal. He is the Head and Associate Professor of Dept. of Computer Science, The University of Burdwan, Burdwan, West Bengal. He was born in 27-2-1964. He received his B.Sc and M. Sc. Mathematics in 1987 and 1989 respectively. Then he completed his MCA from Jadavpur University in 1992. He received his PhD. Degree from Burdwan University in 2004. His research areas are Soft Computing, Document Processing, Web Mining etc. He has a year of Industry and 16 years of teaching experience. He published 30 Journal and 20 Conferences papers. Two students are awarded PhD. Degree and eight students are under his supervision.