ISSN:0975-9646

Disha Gupta et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (4) , 2015, 3334-3337

A Review and Evaluations of Real Time Shortest Path according to current traffic on road

Disha Gupta[1], U. Datta [2]

[1,2] *Computer Science and Engineering Department, Maharana Pratap College of technology*

Putli Ghar Road, Near Collectorate, Gwalior-474006, Madhya Pradesh, India

Abstract- The Shortest Path Problem (SPP) is one of the most fundamental and important in combinatorial Problem. SPP is an important problem in graph theory and has applications in communications, transportation, and electronics problems. In this paper different algorithm for solving SPP with their advantage, disadvantage and application has been discussed. But all these algorithms are work on original shortest path but many times original shortest path don't work properly due to many reasons like traffic problem and road blocking problem and many more called real time problems. To remove these real time problems be proposed a technique "A Review and Evaluations of Real Time Shortest Path according to current traffic on road". According to this technique we can find the shortest path according to traffic on road at current time. So we can save the time of all types of driver.

Keywords- Shortest Path Algorithms, Dijkstra's Algorithm, Bell Bellman-Ford's Algorithm, A* search algorithm, Floyd–Warshall algorithm

I. INTRODUCTION

The problem of computing shortest paths is indisputable one of the well-studied problem in computer science. Dijkstra's algorithm is called the single-source shortest path. It is also known as the single source shortest path problem. It computes length of the shortest path from the source to each of the remaining vertices in the graph. Greedy algorithms use problem solving methods based on action to see if there's a better long term strategy. Dijkstra's algorithm uses the greedy approach to solve the single source shortest problem. it is thoroughly surprising that in the setting of real-weighted graph. Many basic shortest path problems have seen little or no progress since the early work by Dijkstra, Bellman and Ford, Floyd and Warshall, and others [1]. For instance, no algorithm for computing single source shortest paths (SSSPs) in arbitrarily weighted graphs has yet to improve the Bellman–Ford O(mn) time bound, where m and n are the number of edges and verties respectively. A* Algoritm is a graph/tree search algorithm that finds a path from a given initial node to a given goal node it employs a "heuristic estimate" h(x) that gives an estimate of the best route that goes through that node. The bellman-ford algorithm computes single-source shortest path in aweighted digraph. The fastest uniform all-pairs shortest path (APSP) algorithm for dense graphs [2][3] requires time O(n3√log log n/ log n), which is just a slight improvement over the O(n3) bound of the Floyd–Warshall algorithm. Similarly, Dijkstra's O(m + n log n) time algorithm [4][5] remains the best for computing SSSPs on nonnegatively weighted graphs, and until the recent algorithms of Pettie [6][7][8], Dijkstra's algorithm was also the best for computing APSPs on sparse graphs [4][10][5]. The techniques developed for integer- weighted graphs (scaling, matrix multiplication, integer sorting, and thorup's hierarchy-based approach). In order to improve these bounds most shortest path algorithms depend on a restricted type of input. There are algorithms for geometric inputs (see Mitchell's survey [12], planar graphs [13][14][23], and graphs with randomly chosen edge weights [15]-[22].seem to depend crucially on the graph being integer-weighted.

II. PROPOSED WORK

In this paper we present that how to explain shortest path problem. According to the example we have source and destination in this area we have different kind of routes are available in the figure we have mention KM in a particular distance. According to the table there are particular block of traffic type in this block there are six conditions are present like No traffic, Low traffic, Average traffic, High traffic, Jam traffic, Road closed. In an another block we have mention about Approx. Time in minute(s) to cover 1KM distance there are condition between two nodes like if No traffic then-2, Low traffic-3, Average traffic-4, High traffic-10,Jam traffic-20. In another block path color are mentioned like Red, Pink, Yellow, Green, Blue, and Black. Now according to the route tables which are create according to the diagram in this table we calculate the distance and number of minutes. According to the diagram there are only three suitable routs another two routes are block. According to the condition there are applying road closed conditions. we discuss from route-1 in this route there are five path color condition are apply we measure KM according to the color visual in diagram now KM mention in path multiply with Approx.

Table 1: Basic Information of approx distance covered during different traffic using color

Traffic Type	Approx. Time in minute(s) to cover 1KM distance	Path Color	Path Color Code		
			Red	Green	Blue
No Traffic	2	Green	0	255	0
Low Traffic	3	Pink	255	0	255
Average Traffic	4	Yellow	255	255	0
High Traffic	10	Blue	0	0	255
Jam Traffic	20	Red	255	0	0
Road Closed	Not Mention	Black	0	0	0

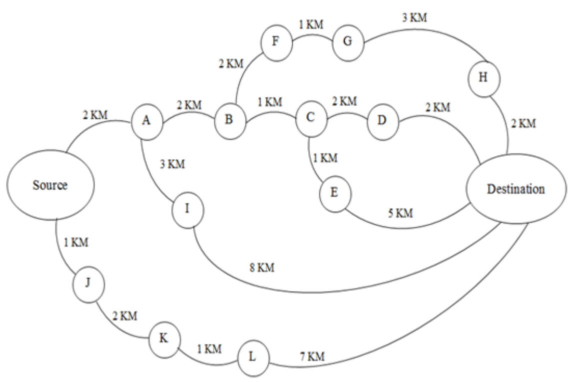

Fig. 1: Basic Graph between source and destination with different routes

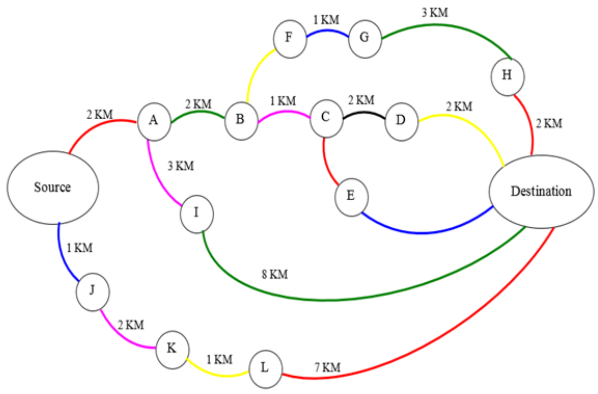

Fig. 2: Graph between source and destination with different routes according to road's traffic using colors

In this Table 1 we explain types of traffic and approx time to be covered between to nodes with the help of color we explained. And explain in figure 1 number of routes from one source to destination. According to this diagram the shortest path is Source-A-B-C-D-Destination and total distance is 9KM but due to some real time reasons may be road is blocked or jammed then this road path not suitable for passerby because this path is shortest path but its takes lot of extra time at a particular. To remove these types of problems we proposed a technique to find the traffic on road at particular time.

In figure 2 we explain distance between source to destination via different routes and color indicates how much traffic on road. And how much approx time will take. In Table 2, 3, 4, 5, 6 we explained how to reach source to destination via different routes at a particular time. According figure 2 we can see A-B-C-D is shortest path (total distance is 9KM) on behalf distance but cannot reach from source to destination because between route C-D is Blocked at particular time explained in Table 2. and in Table 3 we reach source to destination via A-B-C-E (total distance is 11KM). According figure 2 we can reach source to destination in 117 minutes; and in Table 4 we reach source to destination via A-B-F-G-H (total distance is 12KM). According figure 2 we can reach source to destination in 108 minutes; and in Table 5 we reach source to destination via A-I (total distance is 12KM). According figure 2 we can reach source to destination in 65 minutes; and in Table 6 we reach source to destination via J-K-L (total distance is 11KM). According figure 2 we can reach source to destination in 188 minutes; now we can seen shortest path is not better for real time passerby because if these conditions are occur for any passerby then they cannot reach their destination. That's by we proposed this technique to find the shortest path on the basis of current time traffic. Show in figure 1 & 2 and explain by tables 2 to 5.

At is time(by figure 2) the route is Source-A-I-Destination is the best route for passerby because it has 13KM distance but it covers with 65minutes that's by this route save the time of passerby.

Table 2: Route Distance covered in minutes from Source to Destination via A-B-C-D at a particular time

Route 1	Distance between two nodes in KM	Current Path Color	Approx Time in minute(s) to cover 1KM distance according to color	KM mention in path* Approx. Time in minute(s) to cover 1KM distance
Source – A	2	Red	20	40
A – B	2	Green	2	04
B – C	1	Pink	3	03
C – D	2	Black	-	-
D - Destination	2	Yellow	-	-
	Total Distance: 9 KM		-	-

Table 3: Route Distance covered in minutes from Source to Destination via A-B-C-E at a particular time

Route 1	Distance between two nodes in KM	Current Path Color	Approx Time in minute(s) to cover 1KM distance according to color	KM mention in path* Approx. Time in minute(s) to cover 1KM distance
Source – A	2	Red	20	40
A – B	2	Green	2	04
B – C	1	Pink	3	03
C – E	1	Red	20	20
E - Destination	5	Blue	10	50
	Total Distance: 11 KM			**Total Time to be Covered: 117**

Table 4: Route Distance covered in minutes from Source to Destination via A-B-F-G-H at a particular time

Route 1	Distance between two nodes in KM	Current Path Color	Approx Time in minute(s) to cover 1KM distance according to color	KM mention in path* Approx. Time in minute(s) to cover 1KM distance
Source – A	2	Red	20	40
A – B	2	Green	02	04
B – F	2	Yellow	04	08
F – G	1	Blue	10	10
G – H	3	Green	02	06
H - Destination	2	Red	20	40
	Total Distance: 12 KM			**Total Time to be Covered: 108**

Table 5: Route Distance covered in minutes from Source to Destination via A-I at a particular time

Route 1	Distance between two nodes in KM	Current Path Color	Approx Time in minute(s) to cover 1KM distance according to color	KM mention in path* Approx. Time in minute(s) to cover 1KM distance
Source – A	2	Red	20	40
A – I	3	Pink	03	09
I - Destination	8	Green	02	16
	Total Distance: 13 KM			**Total Time to be Covered: 65**

Table 6: Route Distance covered in minutes from Source to Destination via J-K-L at a particular time

Route 1	Distance between two nodes in KM	Current Path Color	Approx Time in minute(s) to cover 1KM distance according to color	KM mention in path* Approx. Time in minute(s) to cover 1KM distance
Source – J	1	Blue	10	40
J – K	2	Pink	03	04
K – L	1	Yellow	04	04
L - Destination	7	Red	20	140
	Total Distance: 11 KM			**Total Time to be Covered: 188**

CONCLUSION

Many shortest path algorithms are proposed but all algorithms are proposed on the basis of distance but these algorithms always not good for passerby because they have cannot explain road's traffic on real time that's by be proposed a technique on the basis of real time traffic. What is the traffic situation? Traffic is Low, Medium, High, Jam or Blocked on route nodes, according to these condition we can decide which route is best and take minimum time to send the passerby from source to destination and passerby can saves the time.

REFERENCES

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, MIT Press, Cambridge, MA, 2001.

[2] U. Zwick, A slightly improved sub-cubic algorithm for the all pairs shortest paths problem with real edge lengths, in Proceedings of the 15th International Symposium on Algorithms and Computation (ISAAC), Lecture Notes in Comput. Sci. 3341, Springer, New York, 2004, pp. 921–932.

[3] M. L. Fredman, New bounds on the complexity of the shortest path problem, SIAM J. Comput., 5 (1976), pp. 83–89

[4] E. W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math., 1 (1959), pp. 269–271.

[5] M. L. Fredman and R. E. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms, J. ACM, 34 (1987), pp. 596–615.

[6] S. Pettie, A new approach to all-pairs shortest paths on real-weighted graphs, Special Issue of Selected Papers from the 29th International Colloqium on Automata Languages and Programming (ICALP 2002), Theoret. Comput. Sci., 312 (2004), pp. 47–74

[7] S. Pettie, On the comparison-addition complexity of all-pairs shortest paths, in Proceedings of the 13th International Symposium on Algorithms and Computation (ISAAC'02), Vancouver, 2002, Springer, New York, pp. 32–43.

[8] S. Pettie, On the Shortest Path and Minimum Spanning Tree Problems, Ph.D. thesis, Department of Computer Sciences, The University of Texas at Austin, Austin, TX, 2003; also available online as Technical

report TR-03-35 at http://www.cs.utexas.edu/ftp/pub/techreports/tr03-35.ps.g
[9] E. W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math., 1 (1959), pp. 269–271.
[10] D. B. Johnson, Efficient algorithms for shortest paths in sparse networks, J. ACM, 24 (1977), pp. 1–13.
[11] M. L. Fredman and R. E. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms, J. ACM, 34 (1987), pp. 596–615.
[12] J. S. B. Mitchell, Geometric shortest paths and network optimization, in Handbook of Computational Geometry, North–Holland, Amsterdam, 2000, pp. 633–701.
[13] G. N. Frederickson, Planar graph decomposition and all pairs shortest paths, J. ACM, 38 (1991), pp. 162–204.
[14] M. R. Henzinger, P. N. Klein, S. Rao, and S. Subramanian, Faster shortest path algorithms for planar graphs, J. Comput. System Sci[FR01] J. Fakcharoenphol and S. Rao, Planar graphs, negative weight edges, shortest paths, and near linear time, in Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS), Las Vegas, NV, 2001, IEEE Press, Piscataway, NJ, pp. 232–241.., 55 (1997), pp. 3–23
[15] P. M. Spira, A new algorithm for finding all shortest paths in a graph of positive arcs in average time O(n2 log2 n), SIAM J. Comput., 2 (1973), pp. 28–32
[16] A. M. Frieze and G. R. Grimmett, The shortest-path problem for graphs with random arc-lengths, Discrete Appl. Math., 10 (1985), pp. 57–77
[17] A. Moffat and T. Takaoka, An all pairs shortest path algorithm with expected time O(n2 log n), SIAM J. Comput., 16 (1987), pp. 1023–1031.
[18] D. R. Karger, D. Koller, and S. J. Phillips, Finding the hidden path: Time bounds for all-pairs shortest paths, SIAM J. Comput., 22 (1993), pp. 1199–1217.
[19] S. G. Kolliopoulos and C. Stein, Finding real-valued single-source shortest paths in o(n3) expected time, J. Algorithms, 28 (1998), pp. 125–141.
[20] U. Meyer, Single-source shortest-paths on arbitrary directed graphs in linear averagecase time, in Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), Washington, DC, 2001, SIAM, Philadelphia, pp. 797–806.
[21] A. V. Goldberg, A simple shortest path algorithm with linear average time, in Proceedings of the 9th European Symposium on Algorithms (ESA), Lecture Notes in Comput. Sci. 2161, Springer, New York, 2001, pp. 230–241
[22] T. Hagerup, Simpler computation of single-source shortest paths in linear average time, in Proceedings in the 21st Annual Symposium on Theoretical Aspects of Computer Science (STACS), Montpellier, France, 2004, Springer, New York, pp. 362– 369
[23] J. Fakcharoenphol and S. Rao, Planar graphs, negative weight edges, shortest paths, and near linear time, in Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS), Las Vegas, NV, 2001, IEEE Press, Piscataway, NJ, pp. 232–241