Analysis of shortest path algorithms

Pawan Jindal[1],Amit Kumar[2],Shishir kumer[3]

GJCST Classification F.2.1

***Abstract*-Shortest path algorithms have large number ofpractical applications in computer networks to flow the information from one computer to the another computer system in the minimum possible time. Researchers are continuously designing new algorithms to solve the shortest path problems which have less time complexity as well as less space complexity as compared to the existing algorithms. In this paper, analysis of shortest path algorithm is being done and it has been concluded that researchers have got remarkable success in designing better algorithms in the terms of space & time complexity to solve shortest path algorithms.**

General TermsAlgorithms, Theory.

Keywords-Shortest path algorithms.

I. INTRODUCTION

An algorithm is defined as computational procedure which takes a particular input and produces aparticular output. Algorithms are used to solve widerange of problems. If G(V,E) is directed weightedgraph, where V represents the set of vertices ofgraph & E represents the set of edges f graph. |V|represents the total number of vertices in graph & |E|represents the total number of edges in the graph. I nshortest path problems, a directed weighted graph isgiven & the goal is to determine the shortest pathamong vertices. There are many variants of shortestpath problems which are given below. In Singlesource shortest path problems, a graph G(V,E) isbeing given & the goal is to find a shortest pathfrom a given vertex to the remaining vertices of thegraph. In Single destination Shortest path problem,the goal is to determine the shortest path from eachvertex of a graph to a particular destination vertex.In Single pair shortest path problem, a pair ofvertices (u,v) is being given and the goal is to findthe shortest path from vertex u to the vertex v. In allpair shortest path problems, the goal is to determineM a shortest path from u to v for every pair of verticesu & v in the graph G(V,E).

II. COMPARISONS OF ALGORITHMS FOR SHORTEST PATH PROBLEMS

Bellman ford algorithm can be used to solve the single source shortest path problems in which edge weight may be negative. This algorithm returns a Boolean value which indicates whether there is negative weight cycle or not in a particular graph. If there is a negative weight cycle which is reachable from the source vertex, then Bellman Ford algorithm indicates that there is no any solution but if there is negative cycle then the algorithm produces the shortest path from the single source vertex to the remaining vertices. If G(V,E) be the graph, Where V represents the set of vertices & E represents the set of edges, then the timecomplexity for Bellman Ford algorithm is O(|V||E|). Dijikstra algorithm can also be used to solve the single source shortest path problems on a given weighted, directed graph G(V,E) if andonly if all the weights of edges are positive The time complexity of Dijikstra algorithm depends upon the implementation of min priority queue. If the minpriority queue is being implemented by using binary heap, then the time complexity of Dijikstra algorithm is O((V+E)lgv). But if the minpriority queue is being implemented by using Fibonacci Heap, then the time complexity for Dijikstra algorithm is O(VlgV+E). All pair shortest path problems can be solved by Floyd Warshall algorithm within the time complexity of O(V3). But the constraint is that there is no any negative weight cycle in the given graph but the edge may be of negative weight. Johnson‘s algorithm can be used to solve all pair shortest path problems within the time complexity of O(V2lgV+VE) time. If the graph contain negative cycle then Johnson‘s algorithm reports that the graphcontains negative cycle. If the graph does not contain negative cycle then Johnson‘s algorithm returns a particular matrix which shows the shortest distance among vertices. If the lengths of edges of a graph are integers, whose absolute value are bounded by N, then the time complexity of the algorithm which is used to calculate the shortest path from a given source node s to the remaining vertices is O(n05mlg(N)). Researchers are continuously applying their best efforts to design the new algorithms for shortest path problems which have less time complexity as well as less space complexity as compared to the existing algorithms. The time complexity for the shortest path algorithm which is given by Upton et al. [1979] is O(n1.5). Henzinger et al.[1997] designed a new algorithm for single source shortest pat problem which has the time complexity of O(n4/3log2/3 (D)) Where D represents the sum of the absolute value of the length. Fakcharoenphol and Rao[2006] designed a new algorithm for single source shortest pat problem in planar graph which has the time complexity of O(nlog3n) & the space complexity of O(nlogn). Ahuja[1] designed a new algorithm for single source shortest path problem which has the time complexity of O(E+V(lgW)0.5) for graph with positive edge weights where w is the longest weight of any edge in the graph. Thorup[2009] designed a new algorithm for single source

About-[1]Deptt. of Computer Science & Engineering Jaypee Institute of Engineering & Technology, Guna , M.P. India
(e-mail - pawan.jindal@jiet.ac.in)
About-[2]Deptt. of Computer Science & Engineering Jaypee Institute of Engineering & Technology, Guna , M.P. India
(e-mail-amit.kumar@jiet.ac.in)
About-[3]Deptt. of Computer Science & Engineering Jaypee Institute of Engineering & Technology, Guna , M.P. India
(e-mail-amit.kumar@jiet.ac.in)

shortest path problems which has the tiem complexity of O(ElglgV). Thorup[2] also designed a new algorithm for single source shortest path problem for undirected graph which has the time complexity of O(E +V). Researchers are continuously applying best efforts in designing new improved algorithm for computing shortest path. Fredman[3] proves that all pair shortest path problems can be solved by using O(V5/2) comparisons between the sums ofweights of edges and has designed a new algorithm which has the time complexity ofO(V3(lglgV/lgV)1/3)time, which is better than the running time complexity of theFloyd-Warshall algorithm. Suppose O(nw) be the running time of the algorithm for multiplying n × n Matrices. As w < 2.376 . Galil and Margalit [4, 5] and Seidel [6] designed an algorithms that solve the all-pairs shortest paths problem for undirected graphs with the time complexity of (Vw p(V)), where p(V) represents a particular function which is polylogarithmically boundedin v. After then several researchers have extended these results to give algorithms to solve the all-pairs shortest paths problem in undirected graphs in which the weights of are integers in the range {1,2,_ _ _,W}. Shoshan and Zwick [7], designed an algorithm which has the time complexity of O(W Vw p(V W)). Karger, Koller, and Phillips[8] and independently McGeoch[9] have designed a new algorithm for a graph with nonnegative edge weights, which has the time complexity of O(V E*+V2 lg V) where E* represents the set of edges in E that participate insome shortest path. For graph with real edge weights, Yuster[10] designed a new algorithm which achieves subcubic running time with the constraint that the number of weight edges emanating from each vertex is O(n0.338). If n is the total number of vertices then the space complexity of this algorithm is O(n^2). The upper bound of the space complexity matches the lower bound of the space complexity.N The quadratic bound for space complexity for all pair shortest path problems is the major bottleneck for many various large scale applications e.g. in the case of internet, the table size of the order of n*n for answering the given distance queries is much larger than the network itself. The n*n table size is too large to be stored in random access memory. So researchers are applying their best efforts to design the efficient algorithms for the all pair approximate shortest path problems. Approximate shortest distance is different from exact shortest distance between two vertices. It means there is some error in the case of approximate shortest distance between two vertices. This error can be additive(surplus) or multiplicative(stretch). Suppose _(x,y) denotes the actual distance between two vertices x & y in a given graph G(V,E). An algorithm is said to compute all pair approximate (stretch) distance for any given graph G(V,E) if for any pair of vertices x,y$\in$V, the distance determined by that algorithm isat least δ(x,y) and at most t δ(x,y). Similarly an algorithm is said to be compute distance with surplus c if the distance determined by the algorithm isatleast δ (x,y) and at most c+ δ(x,y). An algorithm which compute all pair t approximate distance wit t<2 can be easily used to calculate the Boolean matrix multiplication of two n*n boolean matrices. So computing all pair distances with stretch less than two is as hard as the multiplication of two Boolean matrices[11]. Any kind of data structure which is capable of answering a distance query with stretch less than three in constant time must occupy at least Ω(n^2)space in the worst case. Zwick and Cohen[12] designed a new O(n1.5m0.5) algorithm to calculate all pair 2 approximate distances. They also designed an algorithm to compute all pair 7/3 approximate distances which has the time complexity of O(n7/3) & the space complexity of _(n^2)for stretch less than three. Zwick and Cohen[12] also designed an algorithm for stretch equal to three which has the
time complexity of O(n^2lg(n)) & the spacecomplexity of _(n^2). Thorup and Zwick[13] designeda new algorithm for all pair approximate shortestpaths. They showed that for an integer c>=2, anundirected weighted graph can be preprocessed inthe expected tiem of O(cmn1/k) to design a datastructure of size O(cn1+1/c). This particular datastructure is being capable of answering any distancequery with a stretch 2c-1 within the time complexityof O(c). In fact this particular data structure is notstoring all pair approximate distances explicitly,even then it can give the answer of any distancequery in the constant time. So this particular datastructure is known as approximate distance oracle.Algorithm for all pair three stretch distances as given
by Thorup & Zwick[13] is preferred when space has to be optimized as compared to the time. Algorithm for all pair three stretch distance as given by Cohen & Zwick[12] is being preferred when time has to be optimized as compared to the space. Aingworth et al. [14] designed a simple algorithm for finding all distances with an additive error of at most 2 in an unweighted, undirected graph which has the time complexity of O(n5/2). Dor et al.[11] extended the algorithms as given by Aingworth et al. [14] and designed a new algorithm to determine the distances with surplus 2(k−1) for all pair of vertices in unweighted undirected graphs which has the time complexity of O($kn^{2-1/k}m^{1/k}$polylogn). There are also large number of algorithms for all pair approximate shortest paths in unweighted graphs which have multiplicative error as well as additive error simultaneously and which achieve close to quadratic running time.

III. CONCLUSION

In this paper, analysis of shortest path algorithm is being done and it has been concluded that researchers have got remarkable success in designing better algorithms in the terms of space & time complexity to solve shortest path algorithms.

IV. REFERENCES

1) Ravindra K. Ahuja, Kurt Mehlhorn, James B.Orlin, and Robert E. Tarjan. Faster algorithms forthe shortest path problem. Journal of the ACM,37:213–223, 1990.
2) Mikkel Thorup. Undirected single-sourceshortest paths with positive integer weights in linear time. Journal of the ACM, 46(3):362–394, 1999.

3) Michael L. Fredman. New bounds on thecomplexity of the shortest path problem. SIAMJournal on Computing, 5(1):83–89, 1976.
4) Zvi Galil and Oded Margalit. All pairs shortest distances for graphs with small integer length edges.Information and Computation, 134(2):103–139,1997.
5) Zvi Galil and Oded Margalit. All pairs shortestpaths for graphs with small integer length edges. Journal of Computer and System Sciences,54(2):243–254, 1997.
6) Raimund Seidel. On the all-pairs-shortest-pathproblem in unweighted undirected graphs. Journal ofComputer and System Sciences, 51(3):400–403,1995.
7) Avi Shoshan and Uri Zwick. All pairs shortestpaths in undirected graphs with integer weights. InProceedings of the 40th Annual Symposium onFoundations of Computer Science, pages 605–614,1999.
8) David R. Karger, Daphne Koller, and Steven J.Phillips. Finding the hidden path: time bounds forall-pairs shortest paths. SIAM Journal onComputing, 22(6):1199–1217, 1993.
9) C. C. McGeoch. All pairs shortest paths and the essential subgraph. Algorithmica, 13(5):426–441,1995**.**
10) Raphael Yuster. Efficient algorithms on sets ofpermutations, dominance, and real-weighted apsp. InProceedings of 20th Annual ACM-SIAMSymposium on Discrete Algorithms,pages 950–957,2009.
11) Dorit Dor, Shay Halperin, and Uri Zwick. Allpairs almost shortest paths. Siam Journal onComputing, 29:1740–1759, 2000.
12) Edith Cohen and Uri Zwick. All-pairs smallstretch paths. Journal of Algorithms, 38:335– M353,2001.
13) Mikkel Thorup and Uri Zwick. Approximatedistance oracles. Journal of Association ofComputing Machinery, 52:1–24, 2005.
14) Donald Aingworth, Chandra Chekuri, PiotrIndyk, and Rajeev Motwani. Fast estimation ofdiameter and shortest paths(without matrixmultiplication). SIAM Journal onComputing,28:1167–1181, 1999.