ISSN:0975-9646

A. B. Sadavare et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (6) , 2012,5296-5300

A Review of Application of Graph Theory for Network

A. B. Sadavare[1], Dr. R. V. Kulkarni[2]

[1] *Assistant Professor,*
Krishna Institute of Management & Studies, Karad,
Maharashtra, India.

[2] *Professor and HOD,*
Chh. Shahu Institute of Business Education and Research Center,
Kolhapur, Maharashtra, India.

Abstract- **In the modern world, planning efficient routes is essential for business and industry, with applications as varied as product distribution. Networks are used to move people, transport goods, communicate information and control the flow of matter and energy. Networks are all around us. Roads, railways, cables, pipelines are phenomena that frequently need to be represented and analyzed as a network. The complexity of network, cost and time required for networking is increasing in different kinds of network based systems. *(e.g., T.V. cable networks, Telephone networks, Electricity supply networks, Gas pipe network and water supply system).* A graph is a mathematical abstraction that is useful for solving many kinds of problems. Finding shortest paths plays an important role in such kind of network based systems. In graph theory number of algorithms can be applied for finding shortest paths in a graph based network system. It reduces complexity of network paths, cost and time to build and maintain network based systems. In this review of literature authors have reviewed the literatures. This work addresses the problem by presenting analysis of different researches on shortest path problem in various areas of applications. This paper analysis different shortest path algorithms like Dijkstra's Algorithm, Bellman ford Algorithm and Warshall's Algorithm by considering network base systems such as Cable network (T.V. cabling, Telephone cabling, Electricity power supply network) and water supply system network.**

This review of literature also aims to encourage additional research on topics, and concludes with several suggestions for further research.

Keywords- **Graph, Network based systems, shortest path problem, Dijkstra's Algorithm, Bellman ford Algorithm and Warshall's Algorithm.**

INTRODUCTION

A network is a system of points with distances between them. A network can represent roads, pipelines, cables etc. Typical problems with networks involve finding the shortest path between one point in the network and another. In many real occasions, various attributes (various costs and profits) are usually considered in a shortest path problem. Because of the frequent occurrence of such network structured problems, there is a need to develop an efficient procedure for handling these problems.

The shortest path algorithm is among a small group of efficient algorithms that exist for this class of problems. In a network, any node may be connected by edges to any number of other nodes. In most representations, a link also has a cost / weight / distance / length that gives the cost for traveling across the link. Finding the shortest paths can give you the best route from one point to another.

Think of a map as an instance of a graph, the cities are the nodes and the roads between cities are the edges. The length of the road is the weight of the corresponding edge. Shortest path analysis finds the path with the minimum cumulative impedance between nodes on a network. The path may connect just two nodes – the origin and destination –or have specific stops between the nodes. It is based on a network with the objective of finding the path with the minimum cumulative cost in either time or distance between points on the network.

In graph theory, the **shortest path problem** is the problem of finding a path between two vertices (or nodes) in a directed weighted graph such that the sum of the weights of its constituent edges is minimized.

In this study all nodes of graph are represented as places in the city, edges represents roads or paths between places and the weight of the edge represents cost or length of cable or water pipeline.

Graph Theory Basics:

A graph is a mathematical abstraction that is useful for solving many kinds of problems. Fundamentally, a graph consists of a set of vertices, and a set of edges, where an edge is something that connects two vertices in the graph. A *graph* is a pair (V,E), where V is a finite set and E is a binary relation on V. V is called a *vertex set* whose elements are called *vertices*. E is a collection of edges, where an *edge* is a pair (u,v) with u,v in V.

Graph $G = (V, E)$ is a collection of V nodes connected by E links. This definition of a graph is vague in certain respects; it does not say what a vertex or edge represents. They could be cities with connecting roads, or web-pages with hyperlinks.

Path: A path is a simple graph whose vertices can be ordered so that two vertices are adjacent if and only if they are consecutive in the list.

Undirected Graph: A graph in which each edge symbolizes an unordered, transitive relationship between two nodes. Such edges are rendered as plain lines or arcs.

Directed Graph/Digraph: A graph in which each edge symbolizes an ordered, non-transitive relationship between two nodes. Such edges are rendered with an arrowhead at one end of a line or arc.

Loop: A loop is a special type of edge that connects a vertex to itself. Loops are not used much in street network graphs.
Degree: The number of edges which connect a node.
In Degree: Number of edges pointing to a node.
Out Degree: Number of edges going out of a node.
Un-weighted edge: A graph in which all the relationships symbolized by edges are considered equivalent. Such edges are rendered as plain lines or arcs.
Weighted edge: Weighted edges symbolize relationships between nodes which are considered to have some value, for instance, distance or lag time. Such edges are usually annotated by a number or letter placed beside the edge. If edges have *weights*, we can put the weights in the lists. Weight: $\mathbf{w}: \boldsymbol{E} \rightarrow \mathbf{R}$
Tree: An undirected connected graph T is called tree if there are no cycles in it. There is exactly one *simple path* between any vertices u and v.
Simple path: Simple path is a path in which all the vertices are distinct.
Spanning Tree: A sub graph T of a connected graph G, which contains all the vertices of G and T is called a spanning tree of graph G. It is called spanning tree because it spans over all vertices of graph G.
Shortest Path Algorithms:
A) Djikstras' Algorithm:
1) Finds single-source shortest path in weighted graph
2) It replaces the Breadth First Search (BFS) queue with a Priority Queue. Vertices are added to the Priority Queue by their distance away from the source.
3) If negative weight is used, Dijkstra’s algorithm might fail.
4) The runtime of Djikstras' depends on how the Priority Queue is implemented.
5) Dijkstra's algorithm does not work with negative weight arcs.

B) Floyd-Warshall's Algorithm:
1) Finds all-pair shortest path in weighted graph
2) Uses Adjacency matrix.
3) The Floyd-Warshall algorithm compares all possible paths through the graph between each pair of vertices.
4) Negative weights are allowed but Negative cycle is not allowed.
5) The time complexity of this algorithm is O(V^3) and it is slower.

C) Bellman-Ford Algorithm:
1) Finds single-source shortest path in weighted graph and detects negative cycles.
2) Its basic structure is very similar to Dijkstra's algorithm, but instead of greedily selecting the minimum-weight node not yet processed to relax, it simply relaxes all the edges, and does this $|V| - 1$ times, where $|V|$ is the number of vertices in the graph. The repetitions allow minimum distances to accurately propagate throughout the graph, since, in the absence of negative cycles; the shortest path can only visit each node at most once.
3) Bellman-Ford cannot find the shortest path that does not repeat any vertex in such a graph.
4) The runtime: Bellman–Ford runs in $O(V*E)$ time.

REVIEW OF LITERATURE

The researcher has made an attempt to review the literature related to finding shortest path in various applications. However there are number of literatures, the researcher managed to review few articles / papers published in the journal and books.

List of other study title in journals & articles related to research problem:

Dr. Natarajan Meghanathan [1] reviewed Dijkstra algorithm and Bellman-Ford algorithm for finding shortest path in graph. He conclude that the time complexity for Dijkstra algorithm is $O(|E|*log|V|)$ and the time complexity of Bellman-Ford algorithm is $O(|V||E|)$.

Lili Cao, Xiaohan Zhao, Haitao Zheng, and Ben Y. Zhao [2] conclude that search for shortest paths is an essential primitive for a variety of graph-based applications, particularly those on online social networks. For example, LinkedIn users perform queries to find the shortest path “social links” connecting them to a particular user to facilitate introductions. This type of graph query is challenging for moderately sized graphs, but becomes computationally intractable for graphs underlying today’s social networks, most of which contain millions of nodes and billions of edges. They propose *Atlas*, a novel approach to scalable approximate shortest paths between graph nodes using a collection of spanning trees. Spanning trees are easy to generate, compact relative to original graphs, and can be distributed across machines to parallelize queries. They demonstrate its scalability and effectiveness using 6 large social graphs from Facebook, Orkut and Renren, the largest of which includes 43 million nodes and 1 billion edges. They describe techniques to incrementally update Atlas as social graphs change over time. They capture graph dynamics using 35 daily snapshots of a Facebook network, and show that Atlas can amortize the cost of tree updates over time. Finally, they apply Atlas to several graph applications, and show that they produce results that closely approximate ideal results.

Sanchit Goyal [3] studied Travelling Salesman Problem (TSP) problem in combinatorial optimizationin both, operations research and theoretical computer science. Given a list of cities and their pair wise distances, the task is to find a shortest possible tour that visits each city exactly once. It was first formulated as a mathematical problem in 1930 and is one of the most intensively studied problems in optimization. Problems having the TSP structure most commonly occur in the analysis of the structure of crystals, in material handling in a warehouse, the clustering of data arrays. Related variations on the traveling salesman problem include the resource constrained traveling salesman problem which has applications in scheduling with an aggregate deadline. The prize collecting traveling salesman problem and the orienteering problem are also special cases of the resource constrained TSP. Most importantly, the traveling salesman problem often comes up as a sub problem in more complex combinatorial problems, the best known and important one of which is the vehicle routing problem, that is, the problem of determining for a fleet of vehicles which customers should be served by each vehicle and in what order each vehicle should visit the customers assigned to it. Due to the nature of TSP, most common solutions to the problem were found to run feasibly only for a graph with small number of nodes. Not much research was encountered in the survey over problem space analysis of the Travelling Salesman problem. However common approaches have been found to possess a critical region roughly around n = 40[14], after which they start taking increasingly even more time to halt. This happens due to their non efficient time growth functions with respect to the number of nodes. Concorde a very well known fast and exact solution, however holds a record to have solved for 15,112 cities as well. In this research author refer Warshall's Algorithm.

Shalu Wadhwa *(*2000) [5] in this work researcher's have targeted a Network Design Problem (Cable and Trench Problem), which involves a trade-off between utilization costs and capital costs for network construction. A larger network,(the shortest path tree) may cost more to build but may reduce utilization costs by including more attractive origin-destination paths. Conversely, a smaller network, (minimum spanning tree) may increase the utilization costs. A heuristic has been provided which gives us optimal or near-optimal solutions. This heuristic is an adaptation of the Savings algorithm given by Clarke and Wright in 1964, for solving a vehicle routing problem. The heuristic provides us good solutions which can be used as upper bounds for branch and bound methods, giving us the optimal solutions in lesser times than that given by branch and bound without the upper bounds.

F. Benjamin Zhan (Journal of Geographic Information and Decision Analysis, vol.1, no.1, pp. 69-82.) [6] in GIS environment type of analysis, the computation of shortest paths is often a central task because shortest path distances are often needed as input for "higher level" models in many transportation analysis problems such as facility location, network flows, vehicle routing and product delivery, just to name a few. In addition, the shortest path problem usually captures the essential elements of more complicated transportation analysis problems. Hence, it can often be used as a benchmark or a starting point for solving more complicated problems in transportation analysis. With the advancement of GIS technology and the availability of high quality road network data, it is possible to conduct transportation analysis concerning large geographic regions within a GIS environment. Sometimes, this type of analysis has to be completed in real time. As a consequence, these analysis tasks demand high performance shortest path algorithms that run fastest on real road networks. A recent evaluation of shortest path algorithms using real road networks has identified a set of three algorithms that run fastest. These three algorithms are: 1. The Graph Growth Algorithms implemented with two queues (TQQ), 2. The Dijkstra's algorithm implemented with approximate buckets (DKA), and 3. The Dijkstra's algorithm implemented with double buckets (DKD). As a sequel to that earlier evaluation, this paper has reviewed and summarized the data structures and procedures related to the three algorithms. This paper provides a direct source that summarizes a set of shortest path algorithms that run fastest on real road networks. This source should be particularly useful for researchers and practitioners whose research and practice are related to the use of shortest path algorithms.

Kevin M. Curtin (2007) [7] concludes that the network is a compelling research paradigm because its form can so intuitively represent complex systems. The ability to comprehend the complex systems around us—whether they are transportation systems, intricate communication systems such as the internet, or interactions at the cellular level—is of increasing importance in an increasingly complex world. Since networks are fundamentally spatial there is a clear opportunity for research in network analysis that could prove valuable across a wide range of disciplines.

Kamal A. Ahmat [8] studied extensively in association with complex communication networks. They described basic concepts of graph theory and their relation to communication networks. Then they are presented some optimization problems that are related to routing protocols and network monitoring and showed that many of the optimization problems are NP-Complete or NP-Hard. Finally, they explained some of the common tools used to generate network topologies based on graph theory.

Yanghua Xiao [9] takes a problem of online answering shortest path queries by exploiting rich symmetry in graphs. The most famous and widely used algorithm to solve the shortest path problem is Dijkstra, which is fast using heap data structures for priority queues shortest path queries are important in many applications.

Je_ Chen, Jacob Steinhardt (2006) [10] concludes that, Dijkstra's Algorithm traversal algorithms are specialized for finding the shortest paths between vertices on the graph.

Andrew V. Goldberg (Microsoft Research – Silicon Valley) [11] studies Point-to-Point Shortest Path Algorithms. P2P shortest path algorithms with preprocessing have been developed recently. These algorithms are very efficient in practice on road networks and some other kinds of graphs. Many open questions remain, however, in particular theoretical questions. One would like to have a theoretical justification for these algorithms. Two possible directions are proving good worst-case bounds for the algorithms on special graph types or proving average-case bounds on graph distributions. For the latter, random grids like the one used in Section 6 are interesting candidates.

Another set of questions has to do with computing reaches. One can modify a standard all-pairs shortest path algorithm to compute reaches in the same time bound, which is O*(n2) for sparse graphs. Since the size of the output for the all-pairs problem is (n2), there is limited room for improvement. As reaches need only one value per vertex, this argument does not apply to the problem of computing reaches. An interesting open question is the existence of an algorithm that computes reaches – or provably good upper bounds on reaches – in o(n2) time.

Karsten M. Borgwardt and Hans-Peter Kriegel (2005) [12] defined graph kernels based on shortest paths, which are polynomial to compute, positive definite and retain expressivity while avoiding the phenomenon of "tottering". In experiments on classifying graphs model of proteins into functional classes, they outperformed kernels based on random walks significantly. The shortest-path kernels prevent tottering. It is not possible that the same edge appears twice in the same shortest path, as this would violate the definition of a path. Subsequently, artificially high similarity scores caused by repeated visiting of the same cycle of nodes are prohibited in our graph kernel. The shortest-path kernel as described in this article is applicable to all graphs on which Floyd-Warshall can be performed. Floyd-Warshall requires that cycles with negative weight do not exist. If edge labels represent distances, which is the case in most molecular classification tasks, this condition generally holds.

Stefano Pallottino and Maria Grazia Scutella (1997) [13] reported on Shortest Path Algorithms in Transportation models: classical and innovative aspects. Stefano Pallottino and Maria Grazia Scutella had reviewed shortest path algorithms in Transportations in two parts. In first part of the paper, after presenting the problem, they reviewed those classical primal and dual algorithms which seem to be the most interesting in transportation, either as a result of theoretical considerations or because of their efficiency, and also in view of their practical use in transportation models. Promising re-optimization approaches are then discussed. The second part is devoted to dynamic shortest path problems that arise very frequently in the transportation field. They analyzed the main features of the problems and present, under suitable conditions on travel time and cost functions, a general "chronological" algorithmic paradigm, called *Chrono-SPT*.

S.G.Shirinivas, S.Vetrivel and Dr. N.M.Elango (2010) [14] presented the importance of graph theoretical ideas in various areas of computer applications like Shortest path algorithm in a network, Finding a minimum spanning tree, Finding graph planarity, Algorithms to find adjacency matrices, Algorithms to find the connectedness, Algorithms to find the cycles in a graph, Algorithms for searching an element in a data structure (DFS, BFS)

Christian Sommer (2010) [15] investigates shortest path query processing in networks both from a theoretical and a practical point of view. He performs experimental study by considering road transportation network. He had presented a simple and general method based on Voronoi duals to efficiently support shortest path queries in undirected graphs with very low preprocessing overheads and competitive query times, at the cost of exactness. The method was shown to be effective on a variety of graph types while remaining a reasonable alternative to existing exact methods specifically designed for transportation networks.

Sahar Abbasi and Sadoullah Ebrahimnejad (2011) [16] in this paper they considered the dynamic shortest path problem, motivated by its applications in dynamic minimum cost flows in transformation problem. They showed that this problem is equivalent to a classical shortest path problem in a so-called time-expanded network. Although our approach allows us to apply any standard technique on the time-expanded network, the size of this network is typically very large for realistic problems and it may be beneficial to avoid such explicit expansion. They used the Label Correcting Algorithm for solving this problem that the time complexity of the algorithm is *O* (|nT || mT |).

Cheng-Huang Hung (2003) [17] in this thesis, researcher deals with the inverse shortest path length problem (ISPL) in transportation network improvement and bandwidth pricing.

Karsten M. Borgwardt and Hans-Peter Kriegel (2005) [18] in this article the shortest-path kernel is applicable to all graphs on which Floyd-Warshall can be performed. Floyd-Warshall requires that cycles with negative weight do not exist. If edge labels represent distances, which is the case in most molecular classification tasks, this condition generally holds.

By considering previous studies authors find out the scope to go for research to determine optimum path for the T.V. cable network by considering different algorithms namely Dijkstra's, Bellman-Ford and Warshall's Algorithm. Authors will make a comparative study of these three algorithms by developing software tool and adopt the best conclusion.

REFERENCES

[1] Review of Graph Theory Algorithms by Dr. Natarajan Meghanathan, Department of Computer Science, Jackson State University, Jackson, MS 39217.

[2] Atlas: Approximating Shortest Paths in Social Graphs by Lili Cao, Xiaohan Zhao, Haitao Zheng, and Ben Y. Zhao, Computer Science Department, U. C. Santa Barbara.

[3] A Survey on Travelling Salesman Problem by Sanchit Goyal, Department of Computer Science, University of North Dakota, Grand Forks, North Dakota 58203.

[4] The Traveling Salesman Problem for Cubic Graphs by David Eppstein, Univ. of California, Irvine School of Information and Computer Science TSP for cubic graphs D. Eppstein, UC Irvine, WADS 2003.

[5] Analysis of a network design problem by Shalu Wadhwa, *Lehigh University* [2000].

[6] Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures by F. Benjamin Zhan [Department of Geography and Planning, Southwest Texas State University, San Marcos, TX 78666, USA] Journal of Geographic Information and Decision Analysis, vol.1, no.1, pp. 69-82.

[7] Network Analysis in Geographic Information Science: Review, Assessment, and Projections by Kevin M. Curtin Vol. 34, No. 2, 2007, pp. 103-11 (CAGIS) Cartographic and Geographic Information Society.

[8] Graph Theory and Optimization Problems for Very Large Networks. By Kamal A. Ahmat, CITY UNIVERSITY OF NEW YORK/Information Technology New York, USA.

[9] Efficiently Indexing Shortest Paths by Exploiting Symmetry in Graphs by Yanghua Xiao, Wentao Wu, Jian Pei, Wei Wang and

Zhenying He (Department of Computing and Information Technology, Fudan University, Shanghai, China)

[10] Je_ Chen, Jacob Steinhardt -Senior Computer Team, November 2, 2006.

[11] Point-to-Point Shortest Path Algorithms with Preprocessing by Andrew V. Goldberg [Microsoft Research – Silicon Valley].

[12] Shortest-path kernels on graphs by Karsten M. Borgwardt and Hans-Peter Kriegel [Institute for Computer Science Ludwig-Maximilians-University Munich] Proc. 5th IEEE Int. Conf. on Data Mining (ICDM'05), Houston, TX, 2005.

[13] Stefano Pallottino and Maria Grazia Scutella: 1997, Technical report on Shortest Path Algorithms in Transportation models: classical and innovative aspects.

[14] S.G.Shirinivas, S.Vetrivel and Dr. N.M.Elango, International Journal of Engineering Science and Technology: 2010, Application of Graph theory in Computer Science an overview.

[15] Christian Sommer, Department of Computer Science, Graduate School of Information Science and Technology, University of Tokyo: 2010, Approximate Shortest Path and Distance Queries in Networks.

[16] Sahar Abbasi and Sadoullah Ebrahimnejad, International Journal of Business and Social Science: 2011, Finding the Shortest Path in Dynamic Network using Labeling Algorithm.

[17] Cheng-Huang Hung, School of Industrial and systems engineering Georgia Institute of Technology: 2003, Inverse Shortest Path Length Problem.

[18] Karsten M. Borgwardt and Hans-Peter Kriegel, Institute for Computer Science, Ludwig-Maximilians-University Munich, Germany: 2005, Shortest-path kernels on graphs.