A Survey on Shortest Path Routing Algorithms for Public Transport Travel

GJCST Classifications: C.2.2, I.2.0 , I.2.9, K.4.m

S.Meena Kumari & Dr.N.Geethanjali

Abstract- **Routing is the act of moving information across a network from a source to a destination. Along the way, at least one intermediate node typically is encountered. Routing is often contrasted with bridging, which might seem to accomplish precisely the same thing to the casual observer. Routing involves two basic activities: determining optimal routing paths and transporting information groups through a network. Routing also refers to path finding between source and destination. This literature review investigates some of the gateways to path finding in different networks that are listed in present research literature. A selected set of different approaches are highlighted and set in a broader context, illustrating the various aspects of path finding in different networks. Because path finding is applicable to many kinds of networks, such as roads, utilities, water, electricity, telecommunications and computer networks alike, the total number of algorithms that have been developed over the years is immense. The aim of this survey is to compromise a selected cross-section of approaches towards path finding and the related fields of research, such as transportation GIS, network analysis, operations research, artificial intelligence and robotics, to mention just a few examples where path finding theories are employed.**

Keywords- Routing, Shortest path algorithms, ITS, KSP.

I. Introduction

This paper projects about various shortest path algorithms of routing in transportation networks. Routing algorithm is the key element in any networks performance, and thus it can be seen as the brain of the network. “Was it possible to find a path through the city crossing each of its seven bridges once and only once and then returning to the origin?” – This was Euler’s famous “Königsberg bridge” question, dating back as far as 1736. It is often seen as the starting point of modern path finding. The basis of what is now known as graph theory was formed by Euler’s methods and this theory in turn paved the way for path finding algorithms. In long–distance road travelling, where successful route planning, prior to travelling and en-route is essential to finding the optimal path from origin to destination. “Optimal” refers to shortest time, shortest distance, or least total cost, the latter being of major concern in some parts of the country, where travelling by car may mean many costly ferry crossings and expensive to all roads roads in order to get from one’s departure to one’s arrival. Path finding in a fixed static network, set costs for traversing the network, and path finding in a dynamic network, the cost of traversing the network varies over the time of traversing. Because path finding is applicable to many kinds of networks, such as roads, utilities, water, electricity, telecommunications and computer networks alike, the total number of algorithms that have been developed over the years is immense. The aim of this survey is to compromise a selected cross-section of approaches towards path finding and the related fields of research, such as transportation GIS, network analysis, operations research, artificial intelligence, and robotics, to mention just a few examples where path finding theories are employed. Road networks are the backbone of modern society. Consequently, the reliability of this road network is thus a decisive factor not only in terms of market outreach and competition, but also in terms of continuity, to ensure a 24/7 operation of the community we live in. Any threat to the reliability of the road network constitutes a vulnerable spot, a weakness, that need to addressed in order for the network not to fail, given the right (in fact: “wrong”) circumstances. This is of particular concern when considering sparse, rural networks, because what by urban standards is a minor degradation (i.e. car accident, resulting in queuing, delays and diversions) may have severe consequences if occurring in a rural setting (i.e. blocking the only access road for hours, even days or weeks).One hazard to transportation networks that has emerged recently and what may become an increasing concern in the near future are the effects global climate change, with extreme weather and precipitation patterns not seen before, and thus closing or degrading links that were thought invulnerable to such threats (Askildsen, 2004).

S.Meena Kumari, Assistant Professor, Department of M.C.A., G.Pulla Reddy Coillege, Kurnool, A.P., India.
(telephone: 9160333625 email: sreemeena@rediffmail.com)
Dr.N.Geethanjali, Associate Professor, Department of C.S.T., Sri KrishnaDevaraya University, Anantapur, A.P., India.
(telephone: 9440896370 email: anjalicsd@yahoo.com)

II. Intelligent transport systems

In paper, the recent decade’s road transportation systems have undergone considerable increase in complexity and congestion proclivity. From a user point of view, what matters most in relation to a road network is the following: Can I, at the desired time of departure, get from A to B by using the intended route and means of transport, and arrive at a desired time, which would be the best case. Or, does there exist no route or means of travel at all that can take me from A to B at the desired time of departure, let alone within arriving at the desired time, which to the user would be the worst case. This gave rise to the field of ITS, Intelligent Transport Systems, with the goal to apply and merge advanced technology to make transportation more safe and efficient, with less congestion, pollution and environmental impact. In working towards this goal, ITS can take many different forms. Vehicle location and navigation systems are one of these forms and have come along with the emerging field of transport telematics. Transport telemetric implies the large-scale integration and implementation of telecommunication and informatics technology in the field of transportation, penetrating all areas and modes of transport, the vehicles, the infrastructure, the organization, and management of transport.

Zhao (1997) distinguishes between route planning and route guidance as two key elements in vehicle location and navigation systems as part of ITS. Route planning is the process that helps vehicle drivers plan a route prior to driving a specific part of his or her journey. Route guidance is the real-time process of guiding the driver along the route generated by a route planner.

Huang et al. (1995) discriminates route guidance even further, distinguishing between centralized and decentralized route guidance. In the former, vehicles conduct their own path finding using on-board computers and static road maps in CD-ROMs, and applying heuristic search algorithms. Centralized route guidance relies on traffic management centers (TMC) to answer path queries submitted by vehicles linked to it. In this case, Huang et al. (1995) describe a central server holding a materialized view of all shortest paths at that given time, accessed by lookup requests from the vehicles equipped with this system. Although not explicitly stated, it can be assumed that this also is the case in the Advanced Traveler Information System (ATIS) detailed by Shekar and Fetterer (1996) or the ADVANCE project portrayed by both Revels (1998) and Zhao (1997). Boyce et al. (1997) provide a detailed evaluation study of the ADVANCE project for further reference.

III. SHORTEST PATH ALGORITHMS

Efficient management of networks requires that the shortest route from one point (node) to another is known; this is termed as the shortest path. It is often necessary to be able to determine alternative routes through the network, in case any part of the shortest path is damaged or busy. The analysis of transportation networks is one of many application areas in which the computation of shortest paths is one of the most fundamental problems. These have been the subject of extensive research for many years. The shortest path problem was one of the first network problems studied in terms of operations research. Fixed two specific nodes s and t in the network, the goal is to find a minimum cost way to go from s to t. Several algorithms for computing the shortest path between two nodes of a graph are known. This one is due to Dijkstra (1959). Each node is labeled with its distance from the source node along the best-known path. Initially, no paths are known, so all nodes are labeled with infinity. As the algorithm proceeds and paths are found, the labels may change, reflecting better paths. A label may be tentative or permanent. Initially, all labels are tentative. When it is discovered that a label represents the shortest possible path from the source to node, it is made permanent and never changed thereafter. A network consists of arcs, or links, and nodes. The fastest path is calculated as a function associated with the cost of travelling the link. Even though the different research literature tends to group the types of shortest paths problems slightly different, one can discern, in general, between paths that are calculated as one-to-one, one-to-some, one-to-all, all-to-one, or all-to-all shortest paths. In software packages solving static network shortest path problems the software usually aggregates a once-off all-to-all calculation for all nodes, from which subsequent routes then are derived. Clearly, this approach is not feasible for dynamic networks, where the travel cost is time-dependent or randomly varying. However, the majority of published research on shortest paths algorithms has dealt with static networks that have fixed topology and fixed costs. A few early attempts on dynamic approaches, referenced by Chabini (1997), are Cooke and Halsey (1966) and Dreyfus (1979). Not more than a decade ago, Van Eck (1990) reports several hours as an average time for a computer to churn through an all-to-all calculation on a 250-nodes small-scale static network, and several days on a 16.000-nodes large-scale network.

One way of dealing with dynamic networks is splitting continuous time into discrete time intervals with fixed travel costs, as noted by Chabini (1997). Thus, understanding shortest path algorithms in static networks becomes fundamental to working with dynamic networks.

A. Shortest Path In Static Networks

Several algorithms and data structures for algorithms have been put forward since the classic shortest path algorithm by Dijkstra (1959). In its modified version, this algorithm computes a one-to-all path in all directions from the origin node and terminates when the destination has been reached. Deonardo and Fox (1979) introduce a new data structure of reaching, pruning, and buckets. The original Dijkstra algorithm explores an unnecessary large search area, which led to the development of heuristic searches, among them the A* algorithm, that searches in the direction of the destination node. This avoids considering directions with non- favorable results and reduces computation time.

A significant improvement is seen in the bi-directional search, computing a path from both origin and destination, and ideally meeting at the middle. In relation to this search technique, it should be remarked that Jacob et al. (1998) discard bi-directional algorithms as impractical in their computational study of routing algorithms for realistic transportation networks.

Zhan and Noon (1996) had a comprehensive study of shortest path algorithms on 21 real road networks from 10 different states in the U.S., with networks ranging from 1600/500 to 93000/264000 nodes/arcs. In this study, Dijkstra-based algorithms, however differing in data structure, outperform other algorithms in one-to-one or one-to-all fastest path problems.

In summary, the A* algorithm, along with Dijkstra-based algorithms, are preferred in most of the literature researched by the author the author. It is in fact noteworthy that the

Dijkstra algorithm has prevailed to the present date, proving its universal validity.

B. K-Shortest Path In Dynamic Networks

It paper is a result of the recent advances in computer and communications technology, together with the developments in ITS, that have flared a renewed interest in dynamic networks. This interest in the concept of dynamic management of transportation has also brought forward a set of algorithms that are particularly aimed at optimizing the run-time of computations on large-scale networks.

Chabini (1998) lists the following types of dynamic shortest path problems depending on (a) fastest versus minimum cost (or shortest) path problems; (b) discrete versus continuous representation of time; (c) first-in-first-out (FIFO) networks versus non-FIFO networks, in which a vehicle departing at a later time than a previous vehicle can arrive at the destination before the pervious vehicle; (d) waiting is allowed at nodes versus waiting is not allowed; (e) questions asked: one-to-all for a given departure time or all departure times, and all-to-one for all departure times; and (f) integer versus real valued link travel costs.

Fu and Rilett (1996) investigate what they call the dynamic and stochastic shortest path problem by modeling link travel times as a continuous-time stochastic process. The aim of their research was to estimate travel time for a particular path over a given time period. They deviate from the mainstream appraisal of the A* algorithm and advocate the k-shortest path. The reason for this is that standard shortest path algorithms may fail to find the minimum expected paths, particularly when dealing with non-linear optimization, as is the case in developing travel time estimation models. However, in lieu of real data, their research is based on a hypothetical change pattern in travel time.

Based on the research of path finding algorithms, in static networks, Chabini (1997) remarks that a time-space expansion representation can be used in dynamic networks, applying discrete time intervals with fixed costs. Hence, depending on how time is treated, dynamic shortest path problems can be subdivided into two types: discrete and continuous. In the discrete case, if using 15-second time intervals, a full 24-hour implementation would involve calculations on 5760 time discretization, multiplied with the number of nodes and links. Chabini (1997) makes a distinct separation between fastest time paths, in which the cost of a link is the travel time of that link, and minimum cost paths, in which link costs can be of a general form. The difference between these two is nonetheless not explored until Chabini (1998).

Chabini (1997) identifies two key questions in dynamic path finding: (1) what are the fastest paths from one origin to all destinations departing at a given time, and (2) what are the fastest paths from all nodes to one destination for all departure times. He sees the latter as the most significant in relation to ITS, which is true, if one assumes that ITS aims at finding the best path for multiple vehicles with the same destination. In Chabini (1998) the focus extends slightly. Now three questions are put forward: (1) one-to-all fastest path at a given time interval, (2) all-to-one fastest path for all departure times and (3) all-to-one minimum cost path for all departure time intervals.

Chabini (1997, 1998) places emphasis on the all-to-one minimum cost path as the key algorithm with relation to ITS, the reason being that only a limited set of all network nodes are destination nodes in realistic road networks, while there is a considerably larger number of nodes that will be origin nodes. (Moving vehicles tend more to converge to the same goal than to spread in all directions)

Horn (1999) continues along the research trails of Chabini (1997) and Fu and Rilett (1996), but uses a less detailed articulation of travel dynamics, reflecting as he puts it, the recognition that information about network conditions in most parts of the world are most likely to be sparse and that merely estimates of average speed on individual network links are available in most cases. With the presumption that these estimates allow for variation in speed, congestion and delays at nodes, he studied a number of Dijkstra variant algorithms that address these particular conditions. Most important, he propounds an algorithm that calculates an approximation of shortest time path travel duration (path travel time), independent of the particular navigation between nodes. For an experienced vehicle driver, estimated travel time may be more important than the exact route that is to follow. This is a noteworthy addition to the fastest path algorithms in dynamic networks.

C. K-Shortest Path

The shortest path through a network is the least cost route from a given node to another given node and this path will usually be the preferred route between those two nodes. When the shortest path between two nodes is not available for some reason, it is necessary to determine the second shortest path. If this too is not available, a third path may be needed. The series of paths thus derived are known collectively as the k-shortest paths(KSP) and represent the first, second, third,…, k^{th} paths typically of least length from one node to another. The k-shortest path problem is a variant of the shortest path problem, where one intends to determine k paths $p_1,\ldots, p_k$ (in order), between two fixed nodes. The k-shortest paths represent an ordered list of the alternative routes available.

In obtaining the KSPs, it is normally necessary to determine independently the shortest path (k=1) between the two given nodes before computation of the remaining k-1 shortest paths can be carried out. The term shortest does not just apply to the distance between two nodes, but can involve any single component made up of one or more factors, including cost, safety or time, that put a weighting on the route. KSP algorithms are thus widely used in the fields of telecommunications, operations research, computer science and transportation science.

A. W. Brander and M. C. Sinclair made a comparative study of k-shortest path algorithms. Four algorithms were selected for more detailed study from over seventy papers written on this subject since the 1950's. The network was represented as a graph G = (V, E) where V is a finite set of n nodes or vertices V(G) and E is a finite set of m edges (i.e. links or arcs) E(G) that connect the nodes. The work presented was driven by the desire to find a faster algorithm to calculate the KSPs between nodes in a network. The two original algorithms: Yen and Lawler were implemented to provide a reference to the expected speed and improvement available. Katoh was included as it represented a comparatively recent update and modification to Yen. The fourth algorithm Hoffman was implemented after further study as it was felt that it had the potential to outperform the other algorithms.

Based on solving the k-shortest path problem, Jose L.Santos focused on three codes of Removing path algorithm, Deviation path algorithm-first version and Deviation path algorithm-second version were described and compared on rand and grid networks using random generators. Codes were also tested on the USA road networks. One million paths were ranked in less than 3 seconds on random instances with 10,000 seconds for real-world instances. Dreyfus and Yen cite several additional papers on this subject going back as far as 1957.

Shi-Wei LEE and Cheng-Shong WU proposed an algorithm for finding the k-best paths connecting a pair of nodes in a graph G. Graph extension is used to transfer the k-best paths problem to a problem which deploys well-known maximum flow (MaxFlow) and minimum cost network flow (MCNF) algorithms. Two kinds of path finding procedures are often needed in the design of reliable communication networks. The first one is to find k shortest paths between a pair of nodes. Those paths may be simple or allow loops.

For the k-shortest simple paths problem, Lawler proposed the best known algorithm in computation order O(k(m+nlogn) in undirected graphs, where n and m are the no of nodes and links of the input network. For the directed counterpart, Katoh et al. gave the best known bound in O(k n(m+nlogn)). Recently Eppstein developed an efficient KBP algorithm for finding the k shortest paths allowing loops in O(m+nlogn+k), for highly reliable communication network. The solution output by KBP is a real optimal solution for k disjoint paths and it is very useful for planning highly reliable communication networks.

Francesca Guerriero, Roberto Musmanno, Valerio Lacagnina and Antonio Pecorella dealt with the problem of finding the k shortest paths from a single origin node to all other nodes of a directed graph. The data structure used is characterized by a set of k lists of candidate nodes, and the proposed methods differ in the strategy used to select the node to be extracted at each iteration.

IV. CONCLUSION

Evaluation of any heuristic method is subject to the comparison of a number of criteria that relate to various aspects of algorithm performance. Examples of such criteria are running time, quality of solution, ease of implementation, robustness, and flexibility (Barr et al., 1995; Cordeau et al., 2002). Since heuristic methods are ultimately designed to solve real world problems, flexibility is an important consideration. An algorithm should be able to easily handle changes in the model, the constraints and the objective function. As for robustness, should not overly be sensitive to differences in problem characteristics: a robust heuristic should not perform poorly on any instance. Moreover, an algorithm should be able to produce good solutions every time it is applied to a given instance. This is to be highlighted since any heuristics are non-deterministic, and contain some random components such as randomly chosen parameter values. The output of separate executions of these non-deterministic methods on the same problem is in practice never the same. This makes it difficult to analyze and compare results. Using only the best results of a non-deterministic heuristic, as is often done in the literature, may create a false picture of its real performance. So based on the heuristics we would like to do further research work on public transport travel using K-Shortest path algorithm (based on Dijkstra’s algorithm), considering user preferences.

V. REFERENCES

1) Eppstein, D., Finding the K Shortest Paths, SIAM COMPUT 28(2), 652-673.
2) Peytchev E., Coggan J. (2002). See before you go, Traffic Technology International.
3) Patriksson, M. (1994). The Traffic Assignment Problem - Models and Methods. Utrecht: VSP.
4) Ruihong H. & Zhong-Ren P. (2002). An Object-Oriented GIS Data Model for Transit Trip Planning Systems, Journal of the Transportation Research Board: Transportation Research Records 1804 , 205-211.
5) Qiujin Wu & Joanna Hartely., Using K-Shortest Paths Algorithms To Accommodate User Preferences In The Optimization of Public Transport Travel.
6) Benjamin Zhan, F. (February 1998). Shortest Path Algorithms: An Evaluation using Real Road Networks. Transportation Science 32(1).
7) Cherkassky, B. V., Goldberg, A.V. & Radzik, T. (1993). Shortest Paths Algorithms: Theory and Experimental Evaluation. Technical report (93-1480), Computer Science Department, Stanford University.
8) Bellman R. E. (1958) On a Routing Problem. In Journal Quarterly of Applied Mathematics 16, 87-90
9) Dreyfus, S.E. (1969). An Appraisal of Some Shortest-path Algorithms. In Journal Operations Research 17, 395-412.
10) Qiujin Wu, Joanna Hartely, & David AL-Dabass. Time-Dependent Stochastic Shortest Paths Algorithms For A Scheduled Transportation Network.