USING K-SHORTEST PATHS ALGORITHMS TO ACCOMMODATE USER PREFERENCES IN THE OPTIMIZATION OF PUBLIC TRANSPORT TRAVEL

QIUJIN WU, JOANNA HARTLEY

School of Computing and Technology, The Nottingham Trent University,
Burton Street, Nottingham, NG1 4BU, U.K.
E-mail: qiujin.wu@ntu.ac.uk

Abstract: Traffic congestion is becoming a serious problem in more and more modern cities. Encouraging more private-vehicle drivers to use public transportation is one of the most effective and economical ways to reduce the ever increasing congestion problem on the streets (Hartley and Bargiela, 2001). To make public transport services more attractive and competitive, providing travellers with individual travel advice for journeys becomes crucial. However, with the massive and complex network of a modern city, finding one or several suitable route(s) according to user preferences from one place to another is not a simple task. In this paper, the author presents an approach that uses K-shortest paths algorithms to compute a reasonable number of ranked shortest paths, with the ultimate 'most optimal' path being selected by consideration of the preferences. Some experiments have been done based on the public transportation network of Nottingham City.

Keywords: User preferences, K-shortest paths algorithms, Multi-objective algorithms

1. INTRODUCTION

With the increase of cars on the streets, more and more congestion occurs. The traffic congestion causes not only a monetary problem but also a pollution problem at the same time (Peytchev, 1999). Policy debates, promoted by publication of the Transport White Papers at UK and Scottish levels, have identified the need to reduce the number of car journeys and to encourage public transport usage (Hine and Scott, 2000).

As public transport services become more popular, public transportation users need individual route information to help them plan journeys more efficiently. One of the most important pieces of information to be delivered to public transportation users is the quickest bus route(s) between their specified origin and destination according to their preferences.

Route finding is a shortest path problem. Dijkstra's algorithm (Dijkstra, 1959) is often used for solving this problem due to its efficiency and effectiveness. However, Dijkstra's algorithm does not allow for time-dependent links, which is a necessary property of bus routes. Dreyfus (1969) has considered a number of methods incorporating time-dependent links. However, it is still not sufficient because public transport users have various preferences. New algorithms are needed to accommodate these preferences.

In this paper, K-shortest paths (KSP) algorithms are used to compute a reasonable number of ranked shortest paths with the ultimate 'most optimal' path being selected by consideration of the preferences. The results of experiments based on the public transportation network of Nottingham City are also given.

2. USERS' PREFERENCES AND BI-MODAL TRAVEL

One important feature of this research is that the developed algorithm can accommodate public transportation users' preferences. Public transport users' preferences are various. Actually, it is impractical to take every individual preference into consideration. In this paper, the following preferences are considered:

1. Minimum travel time, which can mean either arriving at the destination at the earliest time or leaving the origin at the latest time.

2. Minimum number of bus-changes. Some travellers do not like changing buses, so they would rather travel on a single bus even if the journey time is longer.

3. Minimum walking distance. A traveller carrying heavy or awkward objects may prefer to walk to the nearest bus stop rather than walk a longer distance to another bus stop on a quicker route.

Another feature is that the route finding is based on a bi-modal travel network which considers not only travelling by bus but also on foot. In reality, all bus stops are linked to each other by foot. To make the generated optimal route more practical in the real world, the developed algorithm is based on a bi-modal travel network. This makes the research more challenging because it causes a fundamental topological change to the network and significantly increases the complexity of the network.

3. K-SHORTEST PATHS ALGORITHMS

The KSP problem is closely related to the well-known network optimization problem. It is similar to finding the shortest path in a network except the objective is to identify a number of ranked shortest paths from node *s* (origin) to node *t* (destination) in a network. This problem is of wide interest because it arises in several real world applications, for example in airline optimization (fleet management) and in vehicle routing problems (Palmgren and Yuan, 1998). There are a large number of papers concerning different algorithms for solving the KSP (Palmgren and Yuan, 1998) but not many papers dealing with the applications in real world problems. One of the most important reasons is that the KSP algorithm often generates a large number of infeasible paths. Consequently, a more direct approach is often used. For instance, Huang and Peng's 'objective-oriented algorithm' (2000) is used to accommodate users' preferences in a transportation network. The algorithm computes one shortest path according to the user objective. However, the algorithm may not find the right route if the constraint is ill-defined.

In this paper, the author explores the possibility of using KSP algorithms to accommodate user preferences in the transportation network. The KSP results will be compared with those provided by a single shortest path algorithm with carefully defined constraints.

4. NETWORK REPRESENTATION

Similar to the shortest path problem, the network for the KSP problem is represented as a graph G = (N, A) where N is a finite set of *n* nodes and A is a finite set of *m* arcs. Each arc (i,j) Є A also has a length (or weight) $l_{ij} \geq 0$. In the route finding context, the network is the transportation network. Nodes are bus-stops and arcs represent the time taken to travel between each pair of nodes either by bus or on foot. The task is to find a series of ranked shortest paths between two nodes in this transportation network. A simple graph representation of such a network is shown in Figure 1, where the nodes are shown as numbered circles and the arcs are represented by lines and arrows linking the nodes.

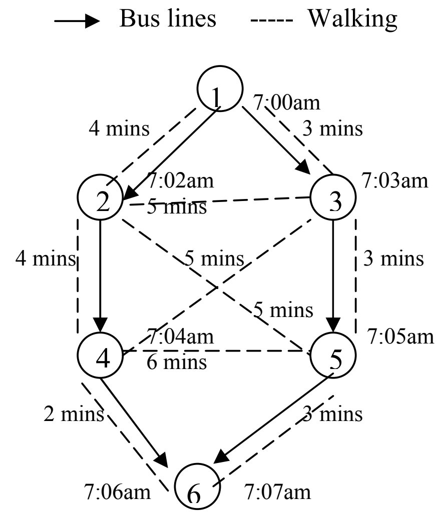

Figure 1: A Simple Transportation Network

5. FINDING RANKED K-SHORTEST PATHS

There are two types of KSP problems: the unconstrained and the constrained KSP problem. In the former, no restriction is considered in finding a path, in the constrained KSP problem all the paths have to satisfy some conditions, such as user preferences. Consequently, this paper describes the constrained KSP problem.

In order to find the K-shortest paths, it is important for the algorithms to choose different routes throughout the network (Brander and Sinclair, 1995). This can be done by labelling the nodes and edges or by removal of a node and removal of an edge: labelling algorithms and path deletion (deviation) algorithms respectively. Shier's (1974,1976,1979) labelling shortest paths algorithms are representative of the first class and Yen's (1971) algorithm is of the second class. The labelling algorithms can be further divided into two sets (Cherkassky et. al., 1996): label correcting algorithms and label setting algorithms. While label correcting algorithms determine the set Pk = $\{p_1, p_2, \ldots p_k\}$ only in the last step of the algorithm, the lable setting algorithms determine each path p_k throughout the computation. Since it is easier to implement the labelling algorithms than the path deletion algorithms for the transportation network and the label setting algorithms are much more efficient than the label correcting algorithms, the algorithm developed in this paper is based on the label setting algorithms.

All types of Labelling algorithms have one thing in common -- the use of one or more labels assigned to each node. While in the shortest path problem there is a single label assigned to each node i, in the KSP problem there may be several labels assigned to each node.

Transportation networks in the real world are much more complex. The considered network is a bi-modal travel network. Route finding for travelling on buses is timetable-based which means the length between any pair of nodes are not given directly, the information must be retrieved from the bus timetables. Route finding for travelling on foot requires the distance between any pair of nodes to be calculated according to their location. The two types of arc information are mixed up. The algorithm must be able to distingush between the two different types of information.

Suppose the network in Figure 1 is a transportation network, the ranked K-shortest paths from node 1 (origin) to node 6 (destination) can be found by following the steps described below:

Step 1: Find the closest node to the origin node by bus.

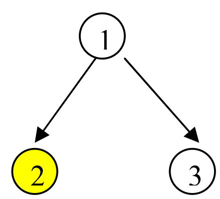

Figure 2: Step 1 of the Algorithm

Suppose node 2 is the closest node to node 1 by bus travelling, record the arrival time timebus(2).

Step 2: Find the closest node to the origin 1 by walking.

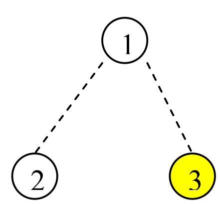

Figure 3: Step 2 of the Algorithm

Suppose node 3 is the closest node to node 1 by walking, record the arrival time timewalk(3)

Step 3: Compare the arrival time of timebus(2) and timewalk(3). If (timebus(2)< timewalk(3)) then label node 2, if (timewalk(3) < timebus(2)) then label node 3.

As explained before, for the KSP problem more than one label is assigned to each node, a flag is used to record how many times a node has been labelled.

Step 4: Upgrade the network according to the labelled node.

Suppose in Step 3: (timebus(2) < timewalk(3)) and node 2 is labelled then the network is upgraded as follows:

In Figure 1, there is a bus link from node 2 to node 4, so the network is upgraded by adding a bus link from node 1 to node 4.

There are also walking links from node 2 to node 3, node 4 and node 5, so the network is upgraded by adding links from node 1 to node 3, node 4 and node 5.

Since there is already a walking link between node 1 and node 3, the walking link 1→2→3 becomes the second walking link from node 1 to 3. It must be recorded separately as a second walking link.

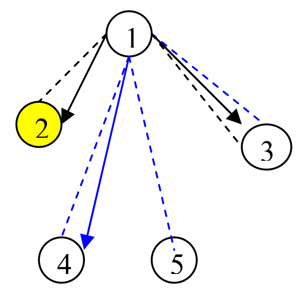

Figure 4: Step 4 of the Algorithm

Or, suppose in Step 3: timewalk(3) < timebus(2) and node 3 is labelled then the network is upgraded using the same principle:

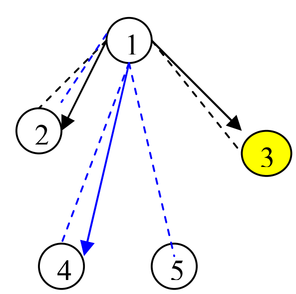

Figure 5: Step 4 of the Algorithm

Step 5: If the labelled node is the destination node, output the route from the origin to the destination.

Step 6: Repeat step 1 to step 5 until the Kth ranked shortest path has been outputted.

Algorithm – Finding ranked KSPs for transportation network.

```
{ count(i) – number of paths that were determined
from origin to destination
elm(i) – flag assigned to each node
K – number of routes to be computed

count(origin)=1
elm(origin)=1
if (there is a bus link or a walking link from origin
node to node i),then
do begin
  elm(i)=1
end if
While (count(destination)<K)
do begin
  j=closest node to origin node
timej=arrival time of the closest node
 count(i)=count(i)+1
if (j=des) then
do begin                //If j=des then one shortest path
  output the route      //is found, output the route
end if

if (count(i)<=K) then
do begin
  if (there is a bus link or walking link from j to i)
then do begin
  elm(i)=elm(i)+1
  upgrade the network by adding nodes and links
end
end if
end do }
```

6. USING THE K-SP ALGORITHM TO ACCOMMODATE USERS' PREFERENCES

To accommodate users' preferences, the routes must satisfy more conditions. This can be done by either comparing each route with the preferences and choosing the first route which satisfies the preferences or regarding the preferences as constraints and embedding them into the algorithm.

On the current stage of my research, the constraints have not been embedded into the KSP algorithm. The designed algorithm generates a list of ranked shortest routes and those routes are compared with the users' preferences until the ultimate path being selected. It remains a problem that how many routes need to be generated to find the 'desired' path for the public transport users.

7. COMPUTATIONAL EXPERIMENT

Experimental results are reported in this section, comparing the performance of the single purpose shortest path algorithm and the KSP algorithm. They are based on the public transportation network of Nottingham City which has 2398 bus-stops in total and 292 bus services running everyday. The average runtime for the single purpose shortest path algorithm is 0.5 second while the average runtime for the KSP is between 3 seconds to 20 seconds for finding 2 to 4 shortest paths. The longer execution time of the KSP algorithm is expected as the multi-objective algorithm finds several required routes at the same time. Detailed comparison of computational runtime between the KSP algorithm and those single-purpose shortest path algorithms can be found in the Figure 6.

	Single Purpose Algorithms' Average Runtime (sec)	KSP Algorithms' Average Runtime (sec)
Short (≤ 10 bus stops)	0.08	2.80
Medium (≤ 20 bus stops)	0.60	8.96
Long (≥ 30 bus stops)	0.84	21.60

Figure 6: Executive Time Comparison Between KSP Algorithms and Single-purpose Algorithms

8. CONCLUSION

The feasibility of using KSP algorithms to accommodate users' preferences for a public transportation network is discussed in this paper. Different kinds of KSP algorithms are compared and an algorithm based on the label setting algorithms is developed and implemented on the Nottingham Public Transportation Network. The developed KSP algorithm is ideally expected to find a set of ranked shortest paths which can accommodate users' preferences easily and without many infeasible routes. However, at the current stage, there is still a high overlap rate for the generated K shortest routes and the execution time of the KSP algorithm is still long. The next stage of the research will focus on reducing the overlap rate by carefully embedding the constraints into the KSP algorithm and improving the efficiency of the KSP algorithm by investigating different hierarchical methods to ensure credible execution times. Neural networks, distributed and parallel computing systems will be investigated and implemented one after another until a satisfying performance is gained.

REFERENCE:

Brander A.W., Sinclair M.C., 1995. 'A Comparative Study of k-Shortest Path Algorithms' Proc. 11th UK Performance Engineering Worksh. for *Computer and Telecommunications Systems*

Cherkassky B.V., Goldberg, and T. Radzik A.V., 1996. 'Shortest Paths algorithms: Theory and experimental evaluation'. *Mathematical Programming,* 73: 129-196.

Dijkstra, E., 1959. 'A note on two problems in connexion with graphs', *Numerische Mathematik* 1, 269-271.

Dreyfus, S.E., 1969. 'An Appraisal of Some Shortest-path Algorithms'. *Operations Research,* vol.17, no. 395-412.

Hartley, J.K., Bargiela, A., 2001. "Decision Support for Planning Multi-Modal Urban travel", *Proc. of 13th European Simulation Symposium,* Marseille, October 2001, ISBN: 90-77039-02-3, pp 387-391.

Hine J., Scott J., 2000, 'Seamless, accessible travel: user's views of the public transport journey and interchange', *Transport Policy* 7, 217-226.

Palmgren M. and Yuan D., 'Shortest summary on K shortest path: Algorithms and applications', http://www.esc.auckland.ac.nz/People/Staff/Mason/Courses/LinkopingColGen99/kth.pdf

Peytchev E., Coggan J., 2002. 'See before you go', *Traffic Technology International*

Ruihong H. and Zhong-Ren P., 2002. 'An Object-Oriented GIS Data Model for Transit Trip Planning Systems', *Journal of the Transportation Research Board: Transportation Research Records*, No.1804, pp. 205-211.

Shier D., 1974. 'Computational experience with an algorithm for finding the k shortest paths in a network', *Journal of Research of the NBS,* 78:139-164

Shier D., 1976. 'Interactive methods for determining the k shortest paths in a network', *Networks,* 6:15 1-159.

Shier D., 1979. 'On algorithms for finding the k shortest paths in a network', *Networks,* 9:195-214.

Yen J.Y. 1971, 'Finding the k shortest loopless paths in a network', *Management Science,* 17:712-716

BIOGRAPHIES:

Ms. Qiujin Wu is a research student at the School of Computing and Technology, the Nottingham Trent University. She was rewarded a MA degree in Information Technology at the University of Nottingham in 2002 and got her bachelor in engineering in year 1998 from Shanghai University, China. Ms. Qiujin Wu started her PhD study in October 2002 under the supervision of Dr. Joanna Hartley and Professor David Al-Dabass. Her research topic is: "Accommodating User Preferences in the Optimisation of Public Transport Travel".

Dr. Joanna Hartley was awarded a BSc (Hons) degree in Mathematics at the University of Durham in 1991. In 1992, she became a research assistant at The Nottingham Trent University and was awarded a PhD in 1996. The title of her PhD is "Parallel Algorithms for Fuzzy Data Processing with Application to Water Systems". She is now a senior lecture at The Nottingham Trent University and an active member of the simulation and Modelling group. Her current research interests include parallel processing, mathematical modelling and probabilistic state estimation relating to urban traffic networks and water distribution systems.