Finding in Multimodal Networks without Timetables

Pedro Oscar Pérez Murueta
TECNOLÓGICO DE MONTERREY
Querétaro, México
pperezm@itesm.mx

Edgar García
IBM
México, México
garciate@mx1.ibm.com

Ma. de los Angeles Junco Rey
TECNOLÓGICO DE MONTERREY
Estado de México, México
ajunto@itesm.mx

***Abstract*— In this paper, we present an algorithm that finds a set of possible routes in a public transport network with no fixed timetables, and ranked according individual preferences and restrictions. A case study based on Mexico City transit network is presented.**

Keywords-Shortest path algorithm; public transport network; intermodal trip planning

I. Introduction

Moving in a city using public transportation requires for a commuter to plan ahead on the path to follow and the modes to use, as well as the start time of the trip if there is a schedule restriction for arriving at the destination. In this planning process, the commuter uses available network data personal preferences and past experiences.

By representing the network as a graph and using path finding algorithms, tools can be created to help users find the best path according to individual preferences and restrictions [3][6]. These tools can be used as well by transportation agencies to study the transit network in order to improve it.

Usually, these models rely on a deterministic representation of the network where timetables can be defined for the transit routes; however, another level of complexity is added when we deal with cities in developing countries, where a usual problem is the low reliability of public transportation, especially regarding timetables, caused not only by serious urban traffic congestion, but also by the fragmented nature of the industry [1]. Thus, a different approach is needed to design path finding models suited for this environment.

In this paper, an algorithm is proposed to find a set of alternative paths in a multimodal network with no fixed timetables, ranked according to individual preferences and restrictions.

The organization of this paper is as follows. Section II provides the background of our work. Section III describes the algorithm, its implementation and a case study used to test it. And, finally Section IV presents the conclusions and future works.

II. Background

Finding the path to get from one point to another in a city using a multimodal network involves more than finding the shortest path between those points through the network; you have to take in account the possibility (and limitations) to use different ways with different results according to user preferences and restrictions. For a user the best route may be the fastest one, while another one would prefer the more cost effective or the less crowded.

This problem has been usually handled using multi-criteria path finding algorithms on graphs representing the structure of the multimodal network [2][3], and often adding time-wise weights to deal with differences in routes over time [4][5].

The super-network approach [6], where different modal networks are linked using transfer nodes, is useful to handle the multimodal network as a single graph in order to apply known path finding algorithms.

In order to calculate the departure time, it's needed to add the travel time for each segment and the transition between modes, including waiting times. Usually these models require defining timetables for public transportation, so an expected travel time can be computed for the full path. But, what happens when a timetable is not available? This is usually the case in developing countries' cities due to overall road characteristics, network design and public transportation ownership structures [1].

When no timetables are available, users need to rely on time expectation to compute transition time. This expectation can be the product of individual experience or crow-sourced information, but in any case it won't be a deterministic value, but a probabilistic distribution of the time the user has to wait for the transport in a given stop/station, or the time it takes for a vehicle to go from one station to another at a given time of the day.

III. Algorithm

Let $S = (N,L)$ be a directed graph composed of a set N of nodes and a set of links $L \subset N \times N$, represent a super-network where each node is an entry point to the network.

A preprocessing phase is needed in order to prepare some required information. As the network is expected to remain static and there is a finite number of origins and destination points, it's feasible to compute and store the K shortest paths for any origin-destination node pair to produce a set of paths $P=\{(o,d,i,l,n,C) \mid o,d \in N, i = 1...K,\ l \subset L,\ n \subset N\}$, where l is the set of links making the path, n includes the nodes where the |user will have to wait for a vehicle, and C is a vector of quantitative characteristics of the path (as total distance, cost or mode switch count). Even if this process is time consuming, it will only be done once.

The proposed path finding algorithm takes as input a pair of geographical points to be used as origin and destination, and a set of restrictions and preferences. Restrictions define threshold values to be contrasted with path characteristics in order to discriminate them. Preferences express the weight of each path characteristic used to decide between alternative paths.

In order to find a set of feasible alternative paths the algorithm follows the next steps:

- Origin-Destination selection – For every geographical point to be used as origin or destination, there is a variety of entry points to the multimodal network at walking distance. An entry point $e \subset N$ is considered reachable by walking if the distance from geographical reference is lower than a specified parameter w. So, for an origin-destination pair (o,d), let $eO = \{e \mid e \in N, \text{dist}(o,e) < w\}$and $eD = \{e \mid e \in N, \text{dist}(d,e) < w\}$ be the list of all reachable entry points from the origin and destination respectively. $Pr = \{p \mid p \subset P, p.o \in eO, p.d \in eD\}$ contains all the possible paths to get from o to d.
- Feasible routes – For all the paths in Pr, the user restrictions are evaluated, so only the paths meeting all the specified criteria are kept.
- Travel time – for each path in Pr a travel time value is computed as a random variable following a gamma probability distribution. Let $w_{it} = \Gamma(k_{it}, \theta_{it})$, a random variable that is gamma-distributed with shape k_{it} and scale θ_{it}, represent the expected waiting time in spot i at time t, and likewise $m_{jt} = \Gamma(k_{jt}, \theta_{jt})$ the expected travel time for link j at time t. The travel length value for a path p in a given timeframe t corresponds to the sum of the expected travel time for all the segments and waiting times included in path: $tl(p, t)=\sum_{i\in p.l} w_{it} + \sum_{j\in p.n} m_{jt}$. This value is added to vector C of respective path to be taken into account for path ranking.
- Based on travel time probability distribution, departure time can be calculated specifying a required arrival time and risk propensity. In the same way, an expected arrival time distribution can be computed if departure time is specified.
- Finally the individual preferences are used to weight characteristics in V for all the paths included in Pr, so a multivariate ranking algorithm [9] can be applied to decide which path is better for this user to get from origin to destination.

A. Implementation

The preprocessing phase requires computing the K shortest paths for any origin-destination node pair. This problem has been extensively studied since the early fifties and several implementations have been proposed, including works by Yen [8], Lawler [10], Katoh [11], Hoffman [12], Ahuja [13], Eppstein [7] and Hershberger [14] among others. All of them extend the shortest path algorithm defined by Dijkstra [15] and their efficiency is based on how they are constructed each of the K paths. For this implementation Yen's algorithm was preferred because of the ease with which it could develop a multithreaded implementation. Application was developed in the Java language.

For each pair of nodes in the network, the K shortest paths are found and stored in a database; along with estimated travel time computed using a gamma probability distribution, monetary cost of trip, number of mode shifts and line changes. All of these values are based on structural information contained in network definition.

When a user specifies a pair of origin-destination points, the path finding module uses a set of network entry and exit points to form a list of possible routes to be filtered by user constraints and then sorted by user's preferences. The following considerations are considered to choose these points: i) A person never walks more than an hour to gain access to any stop of any transport mode of the multimodal network; ii) The system will always look up to two entry and exit points to the transport system.

ALGORITHM

INPUT: An origin vertex *V*, a set *S* with all possible destinations, a set *R* of restrictions and number *K* of path to obtain for each origin-destination pair.

OUTPUT: A set *W* with de *K* best evaluated paths for each origin-destination pair.

METHOD:

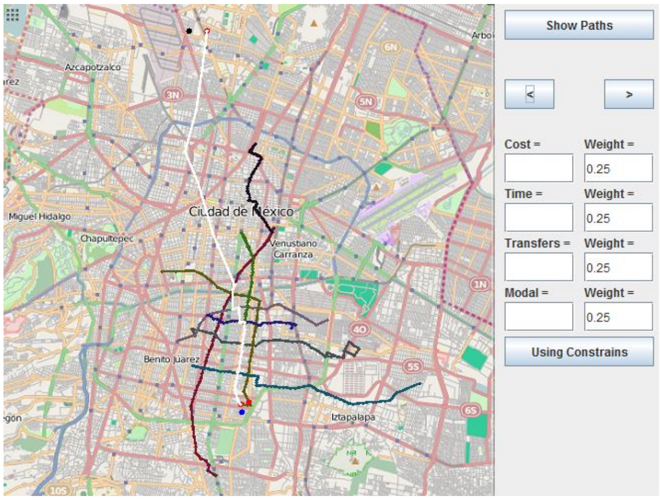
```
    for (each vertex I in S) {
        create_thread(J[I] := Yen's Algorithm(V, I, K, R));
    }
    while (!finish_all_threads) {}
    W := join_all_sets(J);
    return W;
```

B. Case Study

In order to test the implementation, a multimodal network model was built using structural data from Mexico City's public transport network publicly available from Datos Abiertos [16]. The test network includes five different modes: subway, BRT, bus, trolley and suburban train, for a total of 433 nodes. None of these modes offer timetable information or real time data to their users.

In the preprocessing step, the 10 shortest paths were computed for each pair of nodes in the network. Using an Intel Xeon server with 8 GB of RAM, OpenSUSE 12.1 operating system, the full set was generated in 703 seconds.

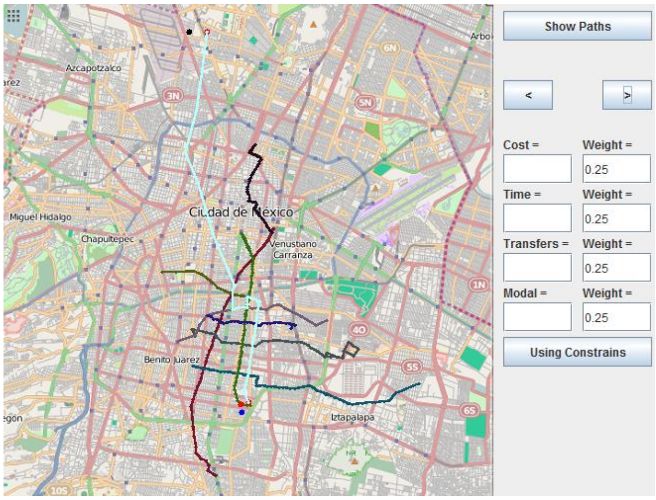
A GUI was developed to offer a visual, interactive, interface to confirm the correct operation of the path finding algorithm. This application displays the public transportation network and, given a pair of origin-destination points in the city, uses the presented algorithm to compute the list of paths and highlights the best path to get from one point to the other, allowing viewing alternative paths with higher costs. Fig. 1 shows the best path for a certain set of points.



Cost =		Weight =	0.25
Time =		Weight =	0.25
Transfers =		Weight =	0.25
Modal =		Weight =	0.25

Figure 1. -The best multimodal path between origin-destination pair of points (without restrictions).

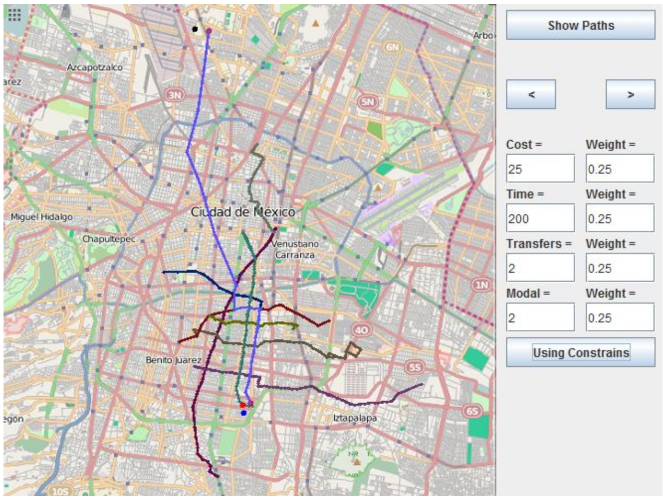
The application also shows us the next best routes. For example, Fig. 2 shows the second best path for the same defined set of points.



Cost =		Weight =	0.25
Time =		Weight =	0.25
Transfers =		Weight =	0.25
Modal =		Weight =	0.25

Figure 2. The second best multimodal path between origin-destination pair of points (without restrictions).

It is also possible to set restrictions (cost, time, number of transfers, number of mode changes) in the searches. Fig. 3 shows one path for a defined set of points with certain restrictions.



Cost =	25	Weight =	0.25
Time =	200	Weight =	0.25
Transfers =	2	Weight =	0.25
Modal =	2	Weight =	0.25

Figure 3. Computed path using restrictions.

IV. CONCLUSIONS AND FUTURE WORK

In countries where public transport systems have no defined time tables is very difficult to determine the travel time between some source-destination pair. We present an algorithm that is able to estimate this time based on probability distributions. At the same time, it generates K best routes taking into account a set of specific restrictions. In a future work, this development will be used as a planning tool for multimodal travel systems and multi-agent traffic simulation.

REFERENCES

[1] R. Iles, Public Transport in Devoloping Countries, Emerald Group Publishing Limited, 2005.

[2] L. Antsfeld and T. Walsh, "Finding Multi-criteria Optimal Paths in Multi-modal Public Transportation Networks using the Transit Algorithm," in Proceedings of ITS World Congress 2012, Octuber 2012, pp. 25.

[3] J. Jariyasunant, E. Mai, and R. Sengupta, "Algorithm for finding optimal paths in a public transit network," Transportation Research Record: Journal of the Transportation Research Board, no. 2256, 2010, pp. 34-42.

[4] A. Orda and R. Rom., "Shortest-path and minimumdelay algorithms in networks with time-dependent edge-length," Journal of the ACM, vol. 37, no. 3, July 1990, pp- 607-625.

[5] E. Pyrgay, F. Schulzz, and D. Wagnerz and C. Zaroliagisy, "Experimental Comparison of Shortest Path Approaches for Timetable Information," ALENEX/ANALC, January 2004, pp. 88-99.

[6] K. Carlier, S. Fiorenzo-Catalano, C. Lindveld, and P. Bovy, "A supernetwork approach towards multimodal travel modeling," in TRB 2003 Annual Meeting, January 2003, pp. 10-2460.

[7] M. Köppen and R. Vicente-Garcia, "A Fuzzy Scheme for the Ranking of Multivariate Data and its Application," in Fuzzy Information, 2004. Processing NAFIPS '04. IEEE Annual

Meeting of the, Fraunhofer IPK, Berlin, Germany, June 2004, pp. 140-145.

[8] J. Y. Yen, “Finding the K Shortest Loopless Paths in a Network,” Manage. Sci., vol. 17, no. 11, 1971, pp. 712–716.

[9] E. L. Lawler, “A Procedure for Computing the K Best Solutions to Discrete Optimization Problems and Its Application to the Shortest Path Problem,” Manage. Sci., vol. 18, no. 7, 1972, pp. 401–405.

[10] N. Katoh, T. Ibaraki, and H. Mine, “An efficient algorithm for K shortest simple paths,” Networks, vol. 12, no. 4, 1982, pp. 411–427.

[11] W. Hoffman and R. Pavley, “A Method for the Solution of the Nth Best Path Problem,” J. ACM, vol. 6, no. 4, 1959, pp. 506–514.

[12] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, Networks flows: Theory, Algorithms and Applications, New Jersey: Prentice Hall, 1993.

[13] D. Eppstein, “Finding the K Shortest Paths,” SIAM J. Comput., vol. 28, no. 2, 1999, pp. 652–673.

[14] J. Hershberger, M. Maxel, and S. Suri, “Finding the K Shortest Simple Paths: A New Algorithm and Its Implementation,” ACM Trans. Algorithms, vol. 3, no. 4, 2007, p. 45.

[15] E. W. Dijkstra, “A note on two problems in connexion with graphs,” Numer. Math., vol. 1, no. 1, 1959, pp. 269–271.

[16] Dirección de Sistemas de Comunicación para la Atención Ciudadana de la Coordinación General de Modernización Administrativa, "Datos Abiertos," 11 Junio 2014. [Online]. Available: http://datosabiertos.df.gob.mx. [Accessed 11 Mayo 2014].