WCTR 2013 RIO

SELECTED PROCEEDINGS

WEB APPLICATION FOR OBTAINING SHORTEST HYPER-PATHS IN CIUDAD UNIVERSITARIA-UNAM

DAVID LÓPEZ
ANGÉLICA LOZANO
HÉCTOR GONZÁLEZ
ALEJANDRO GUZMÁN
INSTITUTO DE INGENIERÍA, UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, MÉXICO

This is an abridged version of the paper presented at the conference. The full version is being submitted elsewhere. Details on the full paper can be obtained from the author.

ISBN: 978-85-285-0232-9

13th World Conference on Transport Research
www.wctr2013rio.com
15-18 JULY 2013
Rio de Janeiro, Brazil

WEB APPLICATION FOR OBTAINING SHORTEST HYPER-PATHS IN CIUDAD UNIVERSITARIA-UNAM

David LÓPEZ[1],

Angélica LOZANO[2],

Héctor GONZÁLEZ[3],

Alejandro GUZMÁN[4]

[1,2,3,4] *Instituto de Ingeniería, Universidad Nacional Autónoma de México, México*

ABSTRACT

This paper describes the generation of a web application for finding shortest hyper-paths on the public transportation system in Ciudad Universitaria Campus (CU) of UNAM (Universidad Nacional Autónoma de México). The considered transportation modes are bus, bicycle and waking; all of them are free. Bus' schedules don't exist, just frequencies are known. Hence, by means of hyper-paths analysis, expected travelled time on bus can be obtained; this time includes waiting time for the bus at stops and considers that multiple bus-lines can be useful to a user at a stop. This expected waiting time at stop in not included in other shortest path web applications. We also consider the number of modal transfers, because any user establishes a maximum number of modal transfers in a trip.

We implemented the Shortest Viable Hyper-path Problem (SVHP) algorithm, by Lozano and Storchi (2002), which was adapted to the characteristics of the transportation modes within CU. The web implementation generates a set of paths (Pareto Optimal set) with different travel time and different number of modal transfer. Then, user can choose her/his path from that set according personal preferences. The web application has a geographical display which facilitates user the choice from among multiple hyper-paths.

Keywords: Hyper-graph, public transport, multimodal, web platform.

INTRODUCCION

As the vehicle fleet grows and the pollution and traffic increase, it is becoming more important changing the use of car to public transportation. Efficient public transportation is important for boosting this transition, where the user has enough information on travel times and paths to reach her/his destination faster. Usually, schedules exist for public transportation, which facilitates users planning their paths. However travel time and waiting time at a bus stop are unknown in some cities due to the lack of schedules. At a stop, a user can choose a set of possible lines to take (for example, bus lines or metro lines), but he/she does not know when a bus of these lines will arrive, and what bus line will arrive first.

Nowadays there are a variety of Web platforms that find shortest paths in multimodal networks such as Google Maps and Bing Maps, however shortest public transportation paths service is only available in cities where schedules exists. Some shortest path services have been implemented on multimodal networks using real-time information, such as PATH2Go (Zhang et al., 2011). Actually, PATH2Go does not consider modal changes; its function is to compare different alternatives that the user has for moving from a source to a destination. The system proposed by Rehrl et al., (2007) considers mode changes and works similar to the travel assistance systems for private vehicles (commonly known as GPS). The *iPhone* application *BayTriper* (Jariyasunant et al., 2011), the portals *Goroo* in Chicago, *Hit the Road* in Dublin and *Trimet* in Oregon are services that integrate multimodal paths and real time information.

All these platforms require schedules to find estimated travel time, and they can find simple shortest paths when there are not schedules.

We present a platform to find shortest hyper-paths in multimodal transport networks, considering that schedules are lacking for some modes, and the number of modal transfers requires be limited by the user. So the user can choose her/his best path, from a set of hyper-paths with different travel time (which includes waiting time at stop) and different number of modal transfers, according personal preferences. Similar platforms are not reported in literature. The presented web application can contribute to a better use of the multimodal public transportation.

The can produce better results when the public modes are not affected by congestion, i.e. they have dedicated lanes (BRT, Suburban Trains, Metro, bicycle and walking).

In our case study (the Ciudad Universitaria Campus of the Universidad Nacional Autónoma de México, CU-UNAM), bus has a dedicated lane but it does not have schedules. Then, waiting time at stops can be estimated by using the frequencies of each line, by means hyper-path analysis. The following free public transportation modes are considered: bus (Pumabus), bicycle and walking.

We assumed that each user has: her/his own opinion about advantages and disadvantages of each mode, her/his preferred modes, and her/his maximum number of tolerated modal transfers in her/his path. For example, some people prefer bicycle over bus for short distances, because bicycle doesn't have waiting time at stops; other people prefer a combined use of bicycle and bus, where speed and comfort meet, and other people don’t like modal transfers.

Hence, we implemented the Shortest Viable Hyper-path Algorithm (SVHA), by Lozano and Storchi (2002), which was adapted to the characteristics of the transportation modes within CU. Our web implementation generates a set of paths (Pareto-Optimal set) with different travel time and different number of modal transfer. Then, user can choose her/his path from that set according personal preferences. The SVH algorithm find shortest viable hyper-paths in multimodal networks, where viable means that certain combinations of modes are respected in a path.

The following sections show how to model a multimodal transportation network using hyper-graphs and also present the developed tool for finding the shortest viable hyper-paths in CU-UNAM. First, concepts on hyper-paths and viability in multimodal networks, and an introduction to the SVH algorithm are presented. Then, the case study characteristics are described, and the structure of the web application is presented. Later, examples of results from the web application are described. Finally, conclusions and references are included.

MULTIMODAL HYPER-GRAPH AND HYPERPATH CONCEPTS

Hyper-graphs and hyper-paths

A hyper-graph is helpful to model public transportation networks because it allow represent waiting time at bus stops. Unlike digraphs whose arrows represent only a cost (distance, time, comfort, etc.), hyper-graphs also consider a probability at stops which is based on the frequency of those lines at a certain stop which are useful to user, called attractive set of lines (Lozano and Storchi, 2002).

Some multimodal hyper-graph concepts, taken from Lozano and Storchi (2002) and Voloshin (2009), are presented below:

A directed hyper-graph or h-graph is a pair $H=(V,E)$ where $V=(v_1,v_2...v_n)$ is a set of nodes and $E=(e_1,e_2... e_m)$ is a set of hyper-arcs. A h-arc $e \in E$ is the couple $e=(t(e),h(e))$ where $t(e) \subset V$ is the set of tail nodes and $h(e) \subset V$, is the set of head nodes.

Let H be a hyper-graph. The forward star $u \in V(H)$ is the set of arcs emerging from u and is defined by $FS(u) = \{(u, y) \in E \mid y \in V(H)\}$. Furthermore, the backward star $u \in V(H)$ is the set of arcs coming to u and is defined by $BS(u) = \{(y, u) \in E \mid y \in V(H)\}$.

Let H be a hyper-graph. A path, q_{od}, connecting a source o and a destination d, is a sequence of nodes and h-arcs, $q_{od}=(o=t(e_1),e_1,t(e_2),e_2,... e_m,d)$, where $t(e_{i+1}) \in h(e_i)$ for i=1,2,...m − 1 and $d \in h(e_m)$.

A hyper-path $\underline{q}_{od}$ is the minimum set of acyclic paths p_{od} such that destination d is connected to any node that belongs to p_{od}.

Let L be the set of lines of a public mode r. Let L_i be the subset of lines passing through node $i \in N$. The *attractive set* $L'_i \subset L_i$ is a set of lines to go from *i* to d such that at node *i* the user is willing to board the first vehicle of subset L'_i which arrives. Each attractive set L'_i associates with a contained boarding *h-arc* $e'_i = (i, h(e'_i))$, where $h(e'_i)$ is the set of line-stops at *i,* of the lines belonging to L'_i.

Expected travel time (Lozano and Storchi, 2002)

The following assumptions are made: passengers arrive randomly at every node stop, and always board the first vehicle of their *attractive set*, which arrives; and lines are statistically independent and vehicles of a line arrive at a node with exponential distribution (equal to the inverse of the line frequency).

Let be φ_j the frequency of the line $l_j \in L_i$ i.e., the frequency of some line l_j that stops at node *i*. Then, $\Phi(e'_i) = \sum_{j \in e'_i} \varphi_j$ denotes the combined frequency *attractive set*; $w(e'_i) = \frac{1}{\Phi(e'_i)}$ represents the average waiting time of the *attractive set* at the stop *i*, for the *attractive set* L'_i; and $\pi_j(e'_i, j) = \frac{\varphi_j}{\Phi(e'_i)}$ denotes de probability that the first vehicle arriving at stop *i* is of line $j \in L_i$.

A hyper-arc has associated a waiting time $C(e'_i) = w(e'_i)$ plus as many coefficients as the number of nodes that in $h(e'_j)$ i.e. the coefficients $\pi(e'_i, j) \quad \forall \ j \in h(e'_j)$ such that $\sum_{j \in h(e'_j)} \pi(e'_i, j) = 1$.

A value $V_p(i)$, which represents the expected travel time for going from the node *i* to *d* is associated with each hyper-path; $V_p(d)=0$ and

$$V_p(i) = \begin{cases} c(i, j) + V_p(j) & \text{if} \quad e = (i, j) \\ w(e'_i) + \sum_{j \in h(e'_j)} \pi(e'_i, j) V_p(j) & \text{if } e' \text{ is a } boardingh - arc \end{cases}$$

Viable multimodal hyper-paths and modal transfers

Lozano and Storchi (2001) define a viable path as a path whose sequence of modes is feasible with respect to a set of constraints. They consider that the constrained modes depend on the specific structure and characteristics of the transportation network in a city.

Lozano and Storchi (2002) indicate that the mode can be classified in several subset according the available modes of specific problem, here *M* is the set of modes, $M_i \subset M$.

They proposed the following general classification: public rail modes (*M1*), private surface modes (*M2*), private modes with parking needs (*M3*), private modes without parking needs (*M4*), walking mode (*M5*) and modal transfer mode (*M6*).

The following concepts are taken from Lozano and Storchi (2002):

A multimodal hyper-graph is the triplet *H= (N, E, M)*, where *N* is the set of nodes, *E* is the set of the h-arcs and *M* are modes associated with the h-arcs.

A multimodal hyper-graph is composed of three types of h-arcs: boarding h-arcs, travel arcs and modal transfer arcs.

A *multimodal hyper-path* is viable if the paths composing the hyper-path do not include more than one maximal mode-r-subpath for each $r \subset M_v$. That is, the paths of a viable multimodal hyper-path do not use a constrained mode more than once.

A wide explanation of multimodal viable paths and hyper-paths is presented in Lozano and Storchi (2001; 2002).

The state s of a viable path is a key to indicate an admissible composition of the modes on the viable path (Lozano and Storchi, 2001). A path with an associated state is a viable path which has a specific sequence of used modes. The state is used to check viability of the paths obtained from the paths concatenation (Lozano and Storchi, 2001). Since a viable hyper-path is composed of a set of viable paths, each one with an associated state, then a state can also be associated with the viable path. This state indicates the specific sequence of the used modes in all of the paths composing the hyper-path (Lozano and Storchi, 2002).

Lozano and Storchi (2002) define a *hypertransition* from states s_z and s_x to state s, as the specification of a state s which indicates the sequence of modes used in both the hyper-paths of states s_z and s_x. They use a *transition states diagram* for representing states for viable paths, obtained from concatenation of *h-arcs* and arcs belonging to a set of modes, forming hyper-paths.

Another important element, which had been considered in multimodal paths is that users don't desire to make many changes of mode, so the number of modal transfers can be limited (Lozano and Storchi, 2001).

Shortest Viable Hyper-path Algorithm (SVHA)

The algorithm for the shortest viable hyper-path problem (SVHP), from Lozano and Storchi (2002), finds the viable hyper-paths with the minimum expected travel time, where the user does not have to execute more than *k* modal transfers. The solution is a set of shortest viable hyper-paths with modal transfer between o and *k*. These hyper-paths have different values associated with the following criteria: the expected travel time and the upper limit of modal transfer, so this set is a Pareto-optimal set (Lozano and Storchi, 2002).

The algorithm for the SVHP is formed of a main procedure which is a label correcting approach, a procedure for arc concatenation (for modes represented by arcs) and a procedure for *h-arc* concatenation (for modes represented by h-arcs). The procedure for arc concatenation calls a procedure for each mode; such procedures determine the state of the path, obtained by the concatenation of a state s path with an arc mode *r*. The procedure for h-arc concatenation calls a procedure for determining the state *s* resulting from the concatenation of the current hyper-path of state s_z and the hyper-path of the h-arc concatenation s_x; this procedure is based on the *transition states diagram*. Both procedures for concatenate arcs and *h-arcs* call procedure-states, which identifies the set of preferred states to *s*. The algorithm is explained in Lozano and Storchi (2002).

CASE STUDY

Area Description

CU campus is located in the southern part of Mexico City, covering an area of 2.7 square kilometres. It has over 107,000 students, 20,000 academics and 25,000 workers. In 2007, the CU campus was declared a World Heritage Site by the United Nations Educational, Scientific and Cultural Organization (UNESCO).

Respectively, Figure 1, Figure 2 and Figure 3 show the bus (Pumabús), bicycle and waking networks within CU Campus. All these transportation modes are free. The bus service includes 12 lines, 83 stops and three main stations (UNAM, 2009b). The bicycle service has 12 borrowing modules and exclusive lanes (6 km). There are eight big free parking lots located at the stadium where students, academics and employees can park-and-ride to their destination using Pumabús or bicycle.

The frequency of each bus line is known, and it changes along the day, so four periods are considered (morning, noon, evening and night). The bus service operates from 6:00 am to 10:00 pm.

Modes and viability

Bicycle and waking modes can be represented by graphs, while bus service by hyper-graphs. At a stop of Pumabús service, several lines can form the attractive set for a user, so the expected waiting time at this stop can be obtained for such user.

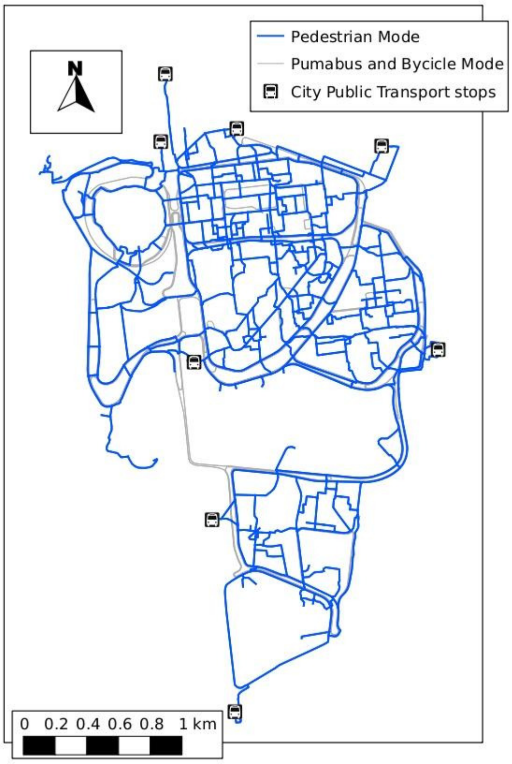

Figure 1 – Pedestrian network in CU Campus

Figure 2 – Bicycle mode network in CU Campus

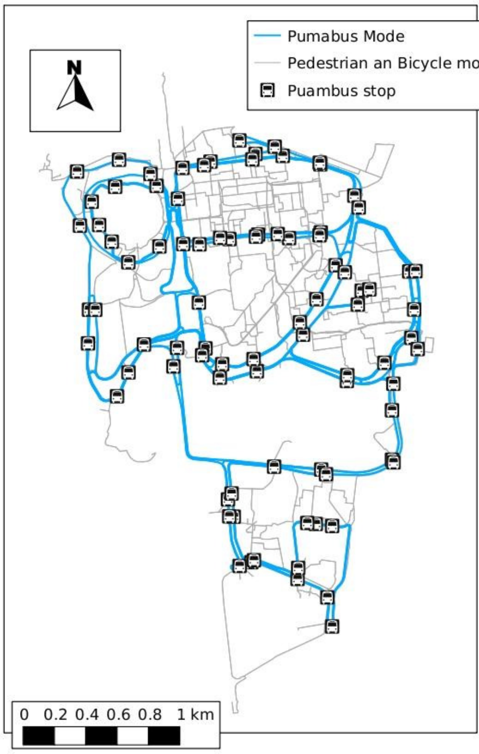

Figure 3 – Coverage of bus mode in CU Campus (12 lines)

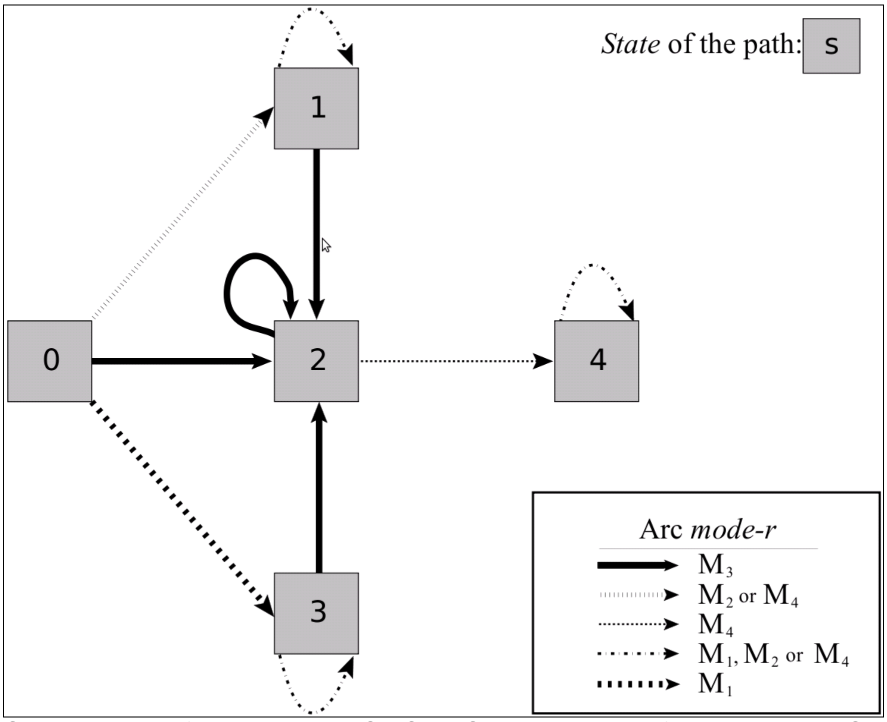

Figure 4 – State transitions for viable paths- CU Case Study. Adaptation from Lozano and Storchi (2002)

The public transportation modes within the CU Campus are classified in four sets, i.e. exist $M_i \subset M$ for $1 \leq i \leq 4$, as follows:

1. M_1: Motorized public transportation ("Pumabús").

2. M_2: pedestrian transportation.

3. M_3: Non motorized public transportation (bicycle).

4. M_4: Modal transfers.

The viability of hyper-paths in CU Campus need to define the transitions of states associated to the hyper-paths, which must be based on the existing modes in the CU Campus.

The *transition states diagram*, which indicates the feasible concatenations of arcs and *h-arcs* belonging to the mentioned four modes, is presented in Figure 4. The unique restricted mode is the bicycle, this can be used only once in a path.

Additional restrictions can be included, like the length of the sub-paths for the bicycle and waking modes (according to personal physical condition).

Finally the limit on the number of modal transfer must be included, given by the user.

Main modifications of the SVHA for the CU Campus Case

Our objective is to develop a web application for finding a viable hyper-path for any origin-destination pair in CU Campus, such that: its expected time is minimum, it uses at least one of the mentioned four public transportation modes, and it contains the maximum number of modal transfers given by the user. A hyper-graph is used for bus mode, because the schedules of bus lines are unknown and just their frequencies are known.

The web applications implemented the SVHA algorithm with a modification on the assignment of a state to the new path or hyper-path. In the SVHA, the procedures for arc and h-arc concatenation, call a procedure for determining the state of the new path or hyper-path. This procedure is now based on the *transition states diagram* shown in Figure 4, i.e., the procedure-states which identifies the set of preferred *states* to *s*, is based on Figure 4. This *transition states diagram* considers only bus, bicycle and waking modes, and that bicycle just can be used once in a trip.

WEB APPLICATION STRUCTURE

The web application of the shortest hyper-path algorithm is based on *Open Source* or free-to-download tools. *Apache* is the web server and *Tomcat* is the application server, where all the *frameworks*, applications and tools live in. These servers are widely used around the world and also have an easy integration with the java code and the infrastructure which provides geographic information. The tools and *frameworks*, used for the web application, are described as follows:

- The geographical extension of *PostgreSQL*, *PostGIS*, handles the database and shape (shp) files of maps.

- *Geoserver* reads the geodatabase stored in *PostgreSQL*, in order visualize it as a map.
- *OpenLayers* integrates and manages the *Geoserver* shape files and the web map services (Google Maps or Open Street Maps).
- Communication between user and application was build using *framework Struts2* which implements the design pattern *Model View Controller*, as illustrated in Figure 5. The core of Struts2 is a filter known as *FilterDispatcher*, which is the entry point to the *framework*. All the *Actions*, like setting initial and final nodes in the algorithm, are executed from the *FilterDispatcher*. The *Actions* are responsible classes of the logic that serves requests by the user; they build the algorithm's data and logic. When an *Action* finishes, a response is send from the server to the user in form of a *Result*. A *Result* is presented to the user in the form of a JSP (Java Server Page). *Struts2* also includes another component called *Interceptor*, which executes tasks before and after *Actions*, but it was not used in this application.

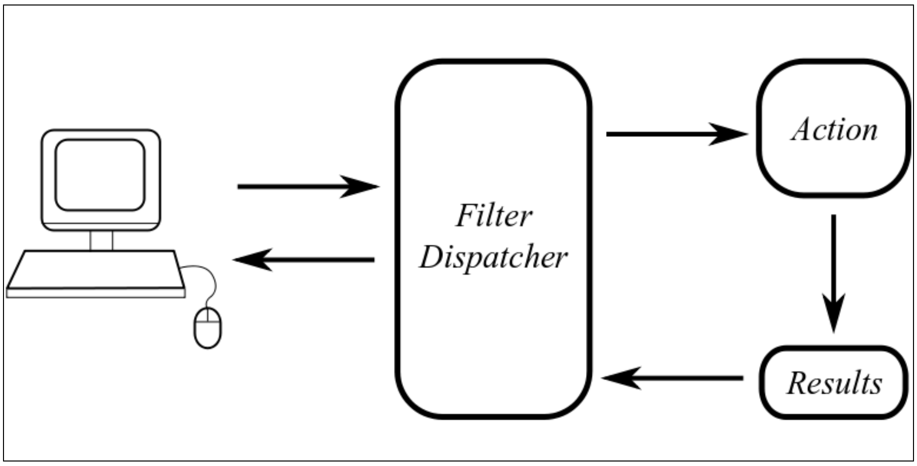

Figure 5 – Struts flow

The web application runs the following steps, when a user is connected:

a) When the web page is opened, *OpenLayes* calls web map services (Google Maps or Open Street Maps) to display an empty map.

b) The user selects an origin, a destination, a schedule and the maximum number of modal transfers. The *FilterDispatcher* reads the inputs and send the user information to the corresponding *Actions* (where the algorithm code is stored).

c) When an answer is found, the *Result* sends a query to *Geoserver*, through *OpenLayers*. *Geoserver* in turn produces a result displayed as a map (the shortest hyper-path).

d) Finally *OpenLayers* shows the map from the previous step, within a web map service. It also displays expected travel time, number of modal transfers and a text travel guide.

WEB APPLICATION RESULTS

The running platform where the algorithm was implemented is shown in Figure 6 (as a close-up screenshot). On the left side panel, user can choose origin, destination, period and maximum number of modal transfers that he/she is willing to do.

When the result is calculated, the left panel displays a drop-down list where the hyper paths belonging to the Pareto Optimal set can be selected and also a list of written directions (instructions for following the hyper-paths).

Figure 6 shows an example of results, where a user asked for multimodal viable hyper-paths from origin “Pumabús E-3” stop to destination “Camino Verde”. Here, the Pareto Optimal contains the following three hyper-paths:

1. Hyper-path 1: expected travel time equal to 17 minutes, and a maximum of 3 modal transfers.

2. Hyper-path 2: expected travel time equal to 18 minutes, and a maximum of 2 modal transfers.

3. Hyper-path 3: expected travel time equal to 19 minutes, and no modal transfers.

The hyper-path 1 is the fastest but has the largest number of modal transfers, this hyper-path may be fit someone who needs arrive at their destination faster no matter how. If a user doesn´t want modal transfers, she/he can choose the hyper-path 3 which is a bit slower than the number 1. There, a user can choose the hyper-path according her/his preferences respect to travel time and modal transfers.

Giving as much as possible information to travelers helps them to understand and navigate a transit system (US DOT, 2011), so the inclusion of a Pareto Optimal set of hyper-paths along with their pros and cons make this platform more flexible and useful.

Figure 6 – Screenshot from the Web application showing the following fields: source, destination, period and maximum number of modal transfers. It also shows the Pareto Optimal set as a *list-box*

Some hyper-paths may be difficult to interpret by the user, because they are composed of several paths to get the destination. Hence, the web application separates results (hyper-paths) into sub-hyper-paths, such that each sub-hyper-path is displayed alone in the map (eliminating hyper-arcs and arcs that don't belong to it).

Also specific directions (instructions) for following the sub-hyper-path are shown in the left panel, indicating the number of modal transfers and the travel time on bicycle and walking modes. This information facilitates user to take a good decision.

The shortest hyper-path from stop "Pumabús E-3" (in the Olympic Stadium) to stop "Camino Verde", at the morning period, is shown in Figure 7. This hyper-path has an expected travel time of 17.00 minutes, and a maximum of three modal transfers. This hyper-path is composed of the following five sub-hyper-paths (whose instructions are shown in the left area of the web application, see Figure 7):

1. Sub-hyper-path with three modal transfers. On the stop "E-3" board a bus of line 6, leave the bus at stop "E-1", walk four minutes to "Bicicentro Estadio Olímpico", request a bicycle and drive two minutes, leave the bike in "Bicicentro Anexo de Ingeniería" and walk four minutes to destination.

2. Sub-hyper-path with three modal transfers. On the stop "E-3" board a bus of line 7, leave the bus at stop "E-1", walk four minutes to "Bicicentro Estadio Olímpico", request a bicycle and drive two minutes, leave the bike in "Bicicentro Anexo de Ingeniería" and walk four minutes to destination.

3. Sub-hyper-path with two modal transfers. On the stop "E-3" board a bus of line 8, leave the bus at stop "Frontones", transfer to "Bicicentro Química", request a bicycle and drive two minutes, leave the bike in "Bicicentro Anexo de Ingeniería" and walk four minutes to destination.

4. Sub-hyper-path with no modal transfers. On the stop "E-3" board a bus of line 6, leave the bus at stop "E-2", board the first bus of lines 7 or 8. If you board a bus of line 8, leave the bus at stop "Camino Verde" (destination). If you board a bus of line 7, leave the bus at stop "E-1" and board the bus line 8 to stop "Camino Verde" (destination).

5. Sub-hyper-path with no modal transfers. On the stop "E-3" board line 8 bus to stop "Camino Verde" (destination).

The calculation and display of the hyper-path in the web browser is fast enough for the network of CU case. It takes lower than 30 seconds in an Ubuntu Server, with 8GB of RAM and an Intel i7 processor. As the network grows, results take longer to be calculated. Response times can be reduced by means of the pre-calculation of hyper-paths for the most popular origin-destination pairs.

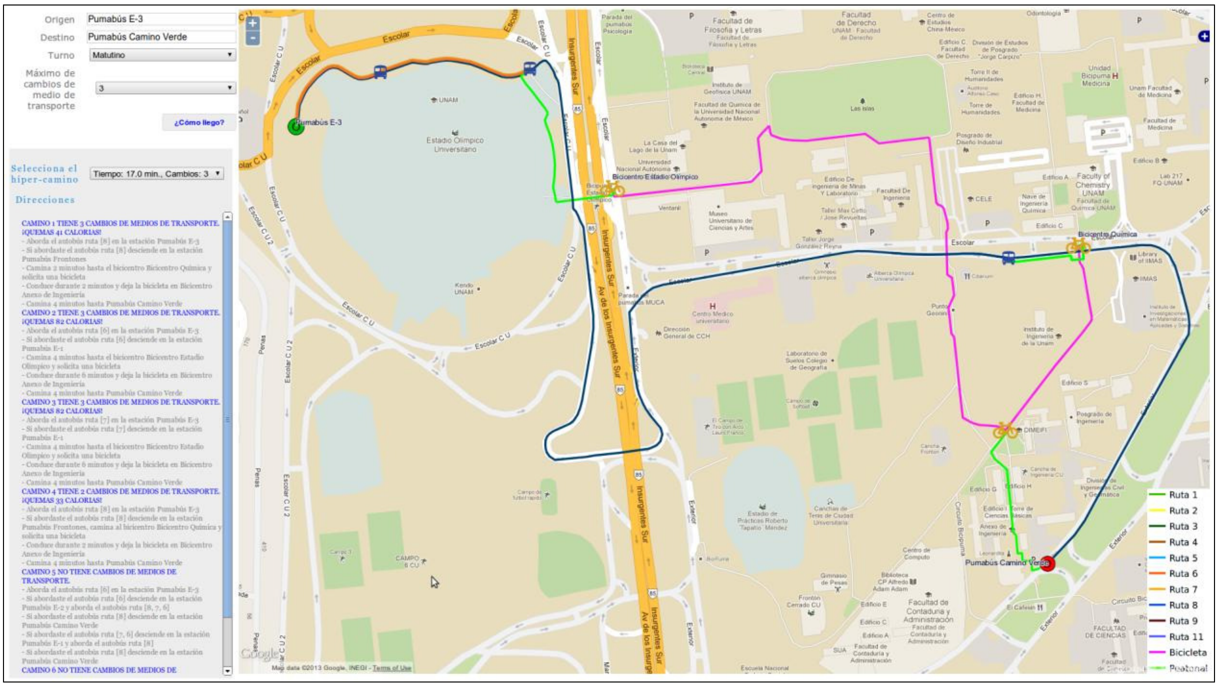

Figure 7 – Screenshot from the Web application showing the shortest hyper-path from "Pumabús E-3" to "Camino Verde".

CONCLUSIONS

A difficulty for the implementation of the algorithm to a real case is the lack of information. If the information on frequency of lines (during the day and during the year) is good, the results will also be also good. In the case study, Pumabús' frequencies are not known by the transport managers; however they have a GPS monitoring system where they keep track the location of all buses throughout the day. Although this system is focused on monitoring the buses, it was possible to determine the frequency of lines for each hour of year 2012, through a spatial queries procedure.

The SVH algorithm by Lozano & Storchi (2002) was never implemented before on a real network. We present the first known platform which incorporates such algorithm, and do it by means a web application, which can be simultaneously used by multiple users.

The implementation of the SVH algorithm in the web application is a powerful tool which can help users find fastest paths, made transfers easy, and get a better use of the existing public transportation infrastructure (Laine et.al, 2003).

The web application for the case study shows the potential of this tool as an Advanced Traveler Information System (ATIS) for multimodal public transport networks. Also, this tool is an economical alternative for boosting changes in travel behavior (Skoglund & Karlsson, 2012), like discourage the use of car or encourage the use of park-and-ride facilities.

The presented web application can be a useful tool in cities where schedules do not exist.

Existing web applications for shortest-path don't cover cases such as the above studied, mainly because waiting times are not handled. In such applications, waiting time can be added as a weight in the arcs, but the results are inaccurate because waiting time is directly related to the *attractive set* and this set is not handled by other algorithms. Also, algorithms included in other web applications do not consider that the user want to limit the number of modal transfers.

The SVH algorithm permits that user limits the number modal transfers, but not the number of line changes. This constraint can be included in the algorithm in future works.

REFERENCES

Jariyasunant, Jerald, Eric Mai, y Raja Sengupta (2011). Algorithm for Finding Optimal Paths in a Public Transit Network with Real-Time Data. Transportation Research Record: Journal of the Transportation Research Board 2256 (-1) (December 1): 34–42.

Laine, T., Pesonen, H. & Moilanen, P. (2003). Joukkoliikenteen internet-eittineuvontapalvelun vaikutusten ja kannattavuuden arviointi. Helsinki: Ministry of Transport and Communications, Helsinki Metropolitan Area Council (YTV).

Lozano A. and Storchi G. (2001). Shortest Viable Path Algorithm in Multimodal Networks. Transportation Research A, 35, 225-241.

Lozano, A. & Storchi, G. (2002). Shortest viable hyper-path in multimodal networks, Transportation Research Part B 36, 853–874.

Rehrl, Karl, Stefan Bruntsch, y Hans-Joachim Mentz (2007). Assisting Multimodal Travellers: Design and Prototypical Implementation of a Personal Travel Companion. IEEE Transactions on Intelligent Transportation Systems 8 (1) (March): 31–42.

Skoglund, T., Karlsson, I.C.M. (2012). Appreciated–but with a Fading Grace of Novelty! Traveller's Assessment of, Usage of and Behavioural Change given Access to a Co-modal Travel Planner. Procedia - Social and Behavioral Sciences 48, 932–940.

UNAM (2009a). Servicio de transporte alternativo, Dirección General de Atención a la Comunidad Universitaria. Accessed 31 August 2009, <http://www.tucomunidad.unam.mx/Bicipuma/>.

UNAM (2009b). Sistema de transporte interno Pumabús, Dirección General de Servicios Generales. Accessed 27 July 2012, <http://www.pumabus.unam.mx/>.

US DOT (2011). Traveler Information Systems and Way finding Technologies in Transit Systems. Research and Innovative Technology Administration, Center, J.A.V.N.T.S. US Department of Transportation. Washington DC.

Voloshin, V. I. (2009). Introduction to Graph and hyper-graphTheory, Nova Science Publishers, United States.

Wachs, M., Samuels, J. M. & Skinner, R. E. (2000). Highway Capacity Manual, Transportation Res-earch Board, United States of America.

Zhang, Liping, Jing-Quan Li, Kun Zhou, Somak Datta Gupta, Meng Li, Wei-Bin Zhang, Mark A. Miller, y James A. Misener (2011). Traveler Information Tool with Integrated Real-Time Transit Information and Multimodal Trip Planning. Transportation Research Record: Journal of the Transportation Research Board 2215 (-1) (December 1): 1–10.