Abstract

Four techniques used to find shortest paths in multimodal transport networks are discussed. The first technique pre-compute of all possible paths between any two points (Jariyasunant et al., 2010). The second one uses a set of rules to build an abstract graph and a relevant graph where the shortest path can be easily found (Ayed et al., 2011). In the third technique, all the topologically similar paths which reach a given node are simultaneously updated (Ziliaskopoulos and Wardell, 2000). Finally, the fourth technique builds a transport network using a database with a hierarchical structure (Wang et al., 2009).

1. Introduction

Globalization turned transport in a complex system, dependent on multiple economic, political actors and common people. Transportation shapes the cities and vice versa; plan it in a sustainable manner is a priority that many countries are adopting, where chaotic growth can harm more than help to the society. The use of tools which provide shortest paths in transport for the users is a way to approach the public transport function to the people. Studies such as Watkins et al. (2011), Dziekan and Kottenhoff (2007) and Tang and Thakuriah (2012) have shown...
that if the user has more accurate information about the transport system, the uncertainty is reduced which can cause stress, increases the willingness to pay and reduces the waiting time perceived by the user.

In this paper four models for multimodal transport systems are reviewed: Jariyasunant et al. (2010), Ayed et al. (2011), Ziliaskopoulos and Wardell (2000) and Wang et al. (2009), each one of these models has a different way of represent multimodal transport systems. For each model a graphical example is presented, all models will represent a four streets network that intersect at four points and where there are two bus lines with two stops, as shown in Fig. 1.

Jariyasunant et al. (2011) use a time expanded graph to model a bus-walk transport system and they solve the problem with the k-shortest path algorithm; although this technique may be slow, they reduce, through different techniques, the size of the graph in order to keep running times low. Ayed et al. (2011) use a time dependent graph and a transfer graph to create an abstract graph which reduces the size of the graph in order to calculate shortest paths with the Dijkstra algorithm. Ziliaskopoulos and Wardell (2000) use a time dependent graph and propose a way to store optimum paths in a data structure which also handles transfers between lines of modes. They propose an algorithm whose complexity is independent of the number of modes and the fixed schedules (Ziliaskopoulos and Wardell, 2000). Finally Wang et al. (2009) construct a graph using a hierarchical structure for model a public transport system in a Geographic Information System; the aim is to find shortest paths through databases queries.

A review of these models, using the same test network, is presented in Sections 2, 3, 4 and 5. Then, a comparison of the models is presented in Section 6. Finally, conclusions and references are included.

2. Time expanded graph model (Jariyasunant et al., 2010)

Jariyasunant et al. (2011) divide the transport system network in two groups: the static graph and dynamic graph. The static graph is a time expanded graph that consists of three sub-graphs.

- Waiting subgraph: formed by the set of nodes and arcs in a time t. It represents the action of waiting at a stop.
- Walk subgraph: formed by the set of nodes and arcs that are not time dependent. It represents the action of walking through the network.
- Boarding subgraph: formed by the set of nodes and arcs that depend on time t. It represents the action of boarding a mode of transport.
To create the *dynamic* graph the model assumes that it is possible to know in real time if a bus is delayed or not and also real time information about traffic is available. If some bus is delayed the *waiting* subgraph is updated by adding a delay $r$ to the waiting times, this action modifies the node and arcs associated to the time $t$ and replace them with nodes and arcs related to the time $t + r$, so there are consequences in the subsequent node and arcs whose times are equal to or greater than $t$. The cost of the arcs in the *boarding* subgraph is updated with the transit real time information.

The authors solve the problem using the $k$-shortest paths techniques, in order to allow the user to choose the option that best suits her/him from a number of possible solutions. When the user makes a query, the first step is pre-compute all feasible paths (those that are possible given a departure time) in the static graph, this pre-computation reduces the size of the graph and limits the number of request to the real-time information services (Jariyasunant et al., 2011). In order to reduce the graph size the authors suppose that the user can only make transfers or boarding to those lines whose stations are within a distance smaller than a given radius. This restriction simulates the behavior where human is only willing to walk a maximum distance for boarding the bus. Moreover the model only accounts those paths that have at most four modal transfers, that’s because the real-time predictions tends to degrade in the future, so far good predictions are achieved in the range of 20 minutes. It is immediate that simulate human behavior and limit the number of possible transfers significantly reduces the size of the graph because the user options to start her/his journey are reduced to a bunch (Bast et al., 2007).

Once the set of feasible paths is created, the information of the arcs is updated in real time. When a bus is late (the information is updated in real time), the set of pre-computed feasible paths is re-calculated. This update removes all paths that cannot be reached because of the delay and add a cost to all the paths that were affected by the delay of the bus. In order to keep the feasible paths list small enough, the authors propose two constraints:

- All the paths with more than $k$-modal transfers for some fixed origin-destination pair are not calculated.
- All the paths whose historic fastest time is greater than the time of some other path are removed (domination).

In the construction of shortest path, the algorithm searches for the set of stops that are closer to the origin, then searches for the list of feasible paths and only those whose ends at a stop near to the destination are selected. Using this information, shortest paths are found with the $k$-shortest paths algorithm. The authors applied the model to the bus transport network of Washington DC (a pedestrian and bus network), and got a maximum of 3 seconds in the path calculation. However, the pre-computation of feasible paths may take up to 99 minutes. The authors do not specify the complexity of the algorithm.

According to the Jariyasunant et al., (2011) (Fig. 2) the model requires one node per intersection, per bus stop and per time interval (entry nodes and exit nodes). The arcs join intersections (representing streets), join entry nodes and exit nodes and join bus stops with the entry and exit nodes in order to represent modal transfers. It is observed
that for a simple street network a large number of elements are required to model it and hence the efficiency for finding shortest paths lies entirely in the methods of simplifying the graph.

3. Time dependent graph model (Ayed et al., 2011)

The model by Ayed et al. (2011) uses a transfer graph based on a directed multimodal time dependent graph. A multimodal directed graph is defined by a set of nodes, a set of arcs, a set of modes (bike, bus, metro, etc.) and a set of time intervals. An arc connects two nodes if and only if there is a mode of transport that connects them, and each arc is associated with a set of time intervals. It is noted that a multimodal time dependent network can be sectioned into monomodal time dependent graphs. Therefore, a transfer graph is a finite set of monomodal time-dependent graphs and a finite set of virtual arcs connecting the monomodal graphs, the nodes of the virtual arcs are called transfer nodes. The authors propose two approaches to find the shortest path: by relevant graphs and a hybrid approach.

In order to build a relevant graph, it is necessary to divide the transfer graph into two components, the intercomponent and intracomponent (Ayed et al., 2008). The intercomponent consists of all paths in the transfer graph that have at least two edges in two different monomodal graphs, i.e., all the paths that use at least one transfer. The intracomponent is the set of paths in the graph that have no transfers.

For an origin, a destination and a departure time, it is possible to calculate all the shortest paths that contain intercomponents (Ayed et al., 2011), based on the following:

- The shortest paths from the origin to the transfer nodes of the same mode as the origin.
- The shortest paths between transfer nodes of the same mode (for all modes in the network).
- The shortest paths from the transfer nodes (of the same mode as the destination) to the destination.

So, a relevant graph is formed of the origin and destination nodes, all the transfer nodes and a set of arcs representing the set of shortest intracomponents. It is noted that defining a multimodal time-dependent graph as a relevant graph significantly decreases the size of the graph. To construct the relevant graphs the authors compare the Dijkstra algorithm and the Ant Colony algorithm; once the relevant graphs are found, the shortest paths are calculated.

Ayed, et al., (2011) propose another method to calculate shortest paths, they refer to this method as the hybrid approach. This method makes an intermediate step in the construction of the relevant graph; it is the construction of the abstract graph. An abstract graph is formed only of the transfer nodes, and a node is adjacent to another if there is a monomodal path that between them.

From the transfer graph and the abstract graph, the relevant intergraph is build which is composed of: all nodes in the abstract graph, the origin node, the destination node and the following arcs: the arcs of the shortest paths in the transfer graph, the arcs of the shortest paths from the origin node to all the transfer nodes (of the same mode as the origin) and the arcs of the shortest paths from all the transfer nodes (of the same mode as the destination) to the destination node. In the relevant intergraph those dominated paths are eliminated, i.e., if $p$ and $q$ are shortest paths in the relevant intergraph with times $t$ and $s$, and number of transfers $k$ and $h$, respectively, and $t \leq s \leq k \leq h$, then $p$ dominates $q$, so $q$ is eliminated from the relevant intergraph. From the relevant intergraph shortest paths are calculated by the Dijkstra algorithm. The authors obtain the best results using the hybrid approach with a computation time of 636 seconds for graphs with 4000 nodes, 15000 arcs and 5 modes. The authors do not specify the algorithm complexity and the considered modes.
Fig. 3 shows the Ayed et al., (2011) model for the test network (Fig. 1). This model needs a graph for each transport mode. The figure shows the street network (black nodes and continued-line arcs) and the bus network (pattern nodes and doted arcs), both graphs are linked by the nodes which represent bus stops. Since the time dependent cost of the arcs is a function associated to the arcs, this model requires fewer elements than the Jariyasunant, et al., (2011) model; however, this graph can also be quite large for representing a real transport system.

4. Time dependent network model (Ziliaskopoulos and Wardell, 2000)

The Ziliaskopoulos and Wardell (2000) model is a graph formed by sets of nodes, arcs, modes and discretized time intervals. Each arc has a different travel for each time interval and for each mode. Also the nodes have associated a delay time which is related to the entry mode and the exit mode of the node, i.e., if the entry mode to node \( i \) is different to the exit mode, the delay represent the time required to make the transfer between modes, if instead the entry and exit modes are the same, then, the delay represents the turning delay.

To each node \( i \), the model associates the time of all the paths arrive at \( i \) from each one of the arcs that enter \( i \). In order to find the optimum time from among all the paths entering \( i \), the authors develop the Time-Dependent Intermodal Least Time Path (TDILTP) algorithm (Ziliaskopoulos and Wardell, 2000). The TDILTP algorithm starts from the destination node and scans in a backward direction all the arcs entering at a node and that could improve the path-time of such node. That is, if there is an optimum path to node \( j \), the algorithm scans all its predecessors, i.e., all the nodes \( i \) which form the backward star of \( j \); if the least time to node \( i \) is improved by the path entering from node \( j \), then node \( i \) is added to a queue to be scanned in a other iteration of the algorithm. When the set of nodes in the queue is empty the algorithm ends. The main drawback of the algorithm is that the optimum path may contain cycles however the authors observe that this behavior is not common in real transport networks. The execution time of the algorithm is 13 seconds for a graph whit 1,000 nodes, 2,747 arcs, 30 lines and 100 time intervals. The complexity is \( O(T^2V^2) \), where \( T \) is the number of time intervals and \( V \) is the number of nodes.

The Ziliaskopoulos and Wardell (2000) model for the test network is shown in Fig. 4, where bus stops and intersections are represented as nodes. The arcs represent streets (linking intersections) and bus lines (linking bus stops). In this model there are not transfer nodes and transfer arcs, so the network size is lower in comparison with the models previously presented. However the information concerning the possible transfers and time intervals is stored in data structures, which require to stores a large amount of information (Jariyasunant, et al. (2011) and Ayed et al. (2011)). The authors try to reduce the size of the data structures by preprocessing all the possible modal transfers for any given time interval, then, only the information for the permitted transfers at some time interval remains.
5. Hierarchical structure network model (Wang et al., 2009)

Wang et al., (2009) use a hierarchical structure network divided in tree levels:

- **Physic level**: represents the roads (sidewalks, streets, rails) where the transport modes transit.
- **Logical level**: represents the roadway (directions and restrictions of the roads).
- **Applicative level**: represents the lines and the stops of the transportation mode. Because all the considered modes use the street network (physical level), this level inherited all the characteristics of the roadway (logical level).

Wang et al., (2009) do not consider the metro within these three levels as this mode flows on its own physical layer (rails), however metro mode could be associated with the street network through the stops located on the street (physical level). The authors define a multimodal graph \( G(V,E) \) where \( V \) is the set of nodes and \( E \) is the set of arcs. The streets (physical level) are defined by the set of arcs \( E \), and a set of nodes \( V \), these nodes are call end points and represent the ends of the streets. Let \( S_i, S_j \in E \) be the arcs that represents two streets, for all \( i, j \), \( S_i \cap S_j = \emptyset \) or \( S_i \cap S_j = EP_k \), such that \( EP_k \in V \), which means that streets intersects only at the end points, i.e., the streets are not divided in the middle by other streets.

The logical level is composed of roadways, intersections and turning tables. Let \( E_r \) be the set of roadways and \( V_r \) the set of intersections. If \( R_i \in E_r \) and \( I_i \in V_r \), then for all \( R_i \in E_r \), exists \( S_j \in E \), so, \( R_i \) is a part \( S_j \), in other words, the roadway is a part of the street Each roadway divides the street according to the features of the network (way, mode restrictions, intersections, etc), i.e., \( R_i \) gives the rules to \( S_j \) for the transportation modes that exist on the street.

Finally, the application level considers two groups modes: public transport modes and private transport mode. A public transport mode is ruled by stops, timetables and lines, while private mode doesn’t have schedules and can be reached at any place in the network at any time. It is noted that at this level, all lines of public transport modes are completely defined. The relationship between public transport modes is defined by a transfer matrix; walking paths are used for connecting a mode to another.

To find shortest paths the authors use a search based algorithm on the hierarchical graph structure. The user defines an origin, a destination and a maximum number of transfers. When the algorithm starts an array stores all paths starting at the origin and another one stores all paths starting at the destination. The search of these paths is easy as they are defined in the application level. When the origin array and the destination array intersect, the algorithm finishes. Otherwise, a search is performed in the nodes of the origin array that have a transfer; each result of this search is a path from the origin that has one transfer and is stored in an auxiliary array containing: the number of transfers, the mode of the path before the transfer and its mode after the transfer. If one of the results from this search contains the destination, the algorithm finishes, otherwise the process is repeated in the auxiliary arrays until the number of transfers exceeds the limit given by the user or a path to the destination is founded.
The Wang et al. (2009) model (Fig. 5) is a non-standard model like the ones where the nodes represent physical points (or time intervals) and the arcs join these nodes; instead Wang et al. (2009) use a geospatial database to model the transport system. The physical network is modeled using: complete streets (shown in strip lines in Fig. 5), roadways (shown in dotted lines in Fig. 5), intersections and end points. The stops (shown in pattern nodes in Fig. 5) are not linked to the graph as they are referenced in the applicative level. In addition the authors create a referential data base with a hierarchical structure where the modes characteristics are defined, such as bus lines, speeds, turning restrictions, etc. The main advantage of this model is the simplicity in terms of representation and maintenance. In the other three models, if for example a bus line changes, disappears or is created, for update the model, it is needed rebuilt almost the whole network but, in the Wang et al. (2009) model is only needed to update part of the database related to the change of the mode. The physical part of the network (streets) is represented in a different level to the modes definition level, i.e., modes are not directly linked to the streets (or physical part), as in the other three models. The main disadvantage of this model is that the authors do not consider (at least not explicitly) the use of time intervals, so that the model is incomplete for real time applications. An interesting research could be try to apply this hierarchical network structure to any of the three models, in order to get the update properties.

6. Models comparison

The Table 1 shows the main characteristics of the four models reviewed in this paper. The table presents in the first row, the authors who develop the model. The second row describes the type of network used to construct the model. The third row lists the elements needed to model the physical network; each cell is divided in two sections: the first section (letter “a”) lists the types of nodes needed to represent the physical points of the network, and the second section (letter “b”) lists the types of arcs needed to represent the roads of the network (such as streets, bus lines, bicycle paths, etc). The Jariyasunant et al. (2011) model is a time expanded graph, hence in addition it needs nodes for representing spatiotemporal events in the network.

The fourth row presents the preprocessing needed in the different models, for finding shortest paths through algorithms, and also the complexity of each algorithm. Both the Jariyasunant et al. (2011) and Ayed et al. (2011) models don’t specify the complexity of the algorithms. So it’s assumed that the Jariyasunant et al. (2011) k-shortest path algorithm has a complexity of at least $O(|A|+|V|\log|V|+k)$ (according Eppstein, D., 1998). Analogously the Dijkstra algorithm used in the Ayed et al. (2011) has at least a complexity of $O(|A|+|V|\log|V|)$ (according Fredman and Tarjan, 1987). The Wang et al. (2009) algorithm uses database searches for finding shortest paths, these searches are repeated at most the number of transfers made by the user, this may see quiet fast however depends on the complexity of the queries that could take some time to execute.

Nowadays the use of real time information in transport systems has become a necessity around the world and several studies (Watkins et al. (2011), Dziekan and Kottenhoff (2007) and Tang and Thakuriah (2012), Zhong et al. (2012) and Ben-Elia et al, (2007)) describe the benefits of using real time information. The models presented in this...
paper could be modify for be used whit real-time information. The fifth row accounts this. The Ayed et al. (2011) and Ziliaskopoulos and Wardell (2000) models are time dependent graphs, so the use of real time information in the functions associated to the time travels is a possibility. The model of Wang et al. (2009) could use real time traffic information through dynamic databases, however, as this model does not consider time scheduling in public transport, it may be not possible the use of real time information for departure times.

The sixth row indicates if the models have been applied in real life instances. Finally the seventh row shows the modes used in the real life implementation.

Table 1. Main characteristics of the reviewed models.

<table>
<thead>
<tr>
<th>Author</th>
<th>Network model</th>
<th>Elements modeled in the physical network</th>
<th>Algorithm and complexity</th>
<th>Real time implementation</th>
<th>Real life implementation</th>
<th>Number of implemented modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jariyasuntant et al. (2011)</td>
<td>Time expanded network, grouped in subgraphs according to actions</td>
<td>a) (Stops × Time intervals + 1) + Intersections + Bus stops + Transfers nodes. b) Streets + Mode Lines + Transfers nodes.</td>
<td>Precomputed viable paths. k-shortest Paths $O(</td>
<td>A</td>
<td>+</td>
<td>V</td>
</tr>
<tr>
<td>Ayed et al. (2011)</td>
<td>Mono modals Time dependent graphs, transfer graph and abstract graph</td>
<td>a) Stops + Intersections + transfers nodes b) Lines of transportation modes + streets + Transfers</td>
<td>Precomputed dominant paths Djikstra $O(</td>
<td>A</td>
<td>+</td>
<td>V</td>
</tr>
<tr>
<td>Ziliaskopoulos and Wardell (2000)</td>
<td>Time dependent graph</td>
<td>a) Intersections + Stops b) Lines of transportation modes</td>
<td>TDILTP $O(</td>
<td>T</td>
<td>^2</td>
<td>V</td>
</tr>
<tr>
<td>Wang et al. (2009)</td>
<td>Hierarchical structure graph. One graph for de street network. Other graph which defines ways and intersections</td>
<td>a) Intersections + street end points b) Streets + roadways</td>
<td>Databases Search Not mentioned</td>
<td>Possible implementation</td>
<td>No application</td>
<td>No application</td>
</tr>
</tbody>
</table>

7. Conclusions

The four models are different approaches to the same problem and is up to the researcher to adopt the one that he considers the best for be implemented in real life instances. Nowadays, transport systems tend to grow and change fast, because the population trend is to live in cities (The World Bank Group, 2013), so the models which are easy adaptable may be preferred over the others in order to save time, money and keep the consistency with the transport systems changes.

In many parts of the world, public transport is based on frequencies, however, in some cities the modes that once were frequency based are starting to use real time information, hence efficient algorithms which use real time information are required.

In this paper, some models for multimodal networks were reviewed in order to understand if they could be implemented using real time information. Although, it may be needed to program the four models and compare execution times to select the fastest, we must also consider which model is easier to maintain with the constant changes in the transportation systems, so the selection of the model requires an analysis beyond the execution times.

Store a transport network in a Geodatabase as in the work of Wang et al. (2009) makes relatively easy the maintenance and modification, as their hierarchical structure allows modifying any of its parts without disrupting the
rest, so the network in subsequent works may have this structure. The Ziliaskopoulos and Wardell (2000) model does not preprocess the network for calculating shortest paths, which is an advantage over the other models to cope the constant changes in the transportation systems, especially if it is intended to use real-time information. However a deeper analysis for the selection of a network model is still needed.

In the Institute of Engineering (UNAM) we develop a platform for finding shortest in multimodal public transport systems where the buses are frequency based. This platform is based on the paper of Lozano and Storchi (2002) and is available at http://hiperpuma.iingen.unam.mx. In future works real time information is going to be added to the platform. So, this is just the beginning of a research whose objective is to propose a model that merges the use of real time information and frequencies for a multimodal transport network in order to develop an Advanced Traveler Information System.

References


