Object modeling and path computation for multimodal travel systems

Maurizio Bielli a,*, Azedine Boulmakoul b, Hicham Mouncif b

a Institute of Systems Analysis and Informatics "Antonio Ruberti", National Research Council, Viale Manzoni 30, 00185 Rome, Italy
b LIST Laboratory, Department of Computer Science, Mohammedia Faculty of Sciences and Technology (FSTM), B.P. 146, Mohammedia 20650, Morocco

Available online 14 April 2005

Abstract

This paper describes a multimodal travel system (MTS) designed to address the needs of a variety of demand-responsive transport. An origin–destination (O–D) trip in transportation network can be accomplished by using multiple modes. In urban network passengers may boarding buses or metros to go from one place to another, and modes as autobus or trains are used by passengers to travel between cities. The work focuses on the network object modeling and multimodal shortest path algorithm. A solution to the problem of long-run planning of transit on multimodal network has been implemented and tested. The work presents the general results found, and the proposed algorithm recognizes the set of constraints related to the time schedule and the sequence of used modes in a O–D trip. The aim is to provide a tool for detecting the facilities of using different travel modes through a transportation network. Routings may include distinct combination of rail, and route. Geographic Information Systems (GIS) were invaluable in the cost-effective construction and maintenance of this work and the subsequent validation of mode sequences and paths selections. Attention is devoted to the multimodal path operator as well as to the use of GIS-transit planning.

Keywords: Object modeling; Transportation network systems; Multimodal routing; Scheduling; GIS-transit itinerary planning

1. Introduction

Multimodal travel system (MTS) is defined as the combination of all traveler modes and kinds of transportation systems operated through various information transport systems. This system focuses on the distribution of transportation related information and the coordination of regional transportation systems for
the benefit of the transportation network users. Travelers need improved means to access information on alternative transport modes and problems affecting their journeys. The goal of MTS is to increase the utilization of high occupancy vehicle modes. By providing travelers, especially regular commuters, with traffic and transportation service information prior to embarking on their trips, travelers can make the most informed choices of modes and routings. Furthermore, trip planning module is a main component of MTS, informing and assisting travelers in choosing the best path to reach their destination in terms of transit modes, transit routes, accessible transit stations, transfers, schedules, travel distances, and more. MTS is the key for significantly raising the quality of services, and increasing investment, to encourage travelers to transfer from car to bus and rail, and thus to reduce congestion and pollution from excessive use of motor vehicles. As its name implies, the trip planning module generates paths for travelers. Each traveler, including transit drivers, itinerant travelers, receives an individual travel plan. Information about each traveler’s activities is used to create trip requests. A trip request consists of several information: the origin and destination of the trip, the ranges for the preferred starting and ending time, and the travel mode choice.

In general, transportation modeling activities is a hard task, especially when considering configuration of several modes of travel. Current and historic patterns of mode choice are used to dictate mode split, or the proportion of trips that will be undertaken by each available mode. In reality, movements of people do not occur solely on the basis of a single mode. The multimodal movements involve the contribution of each of these modes to overall movement patterns of people. Trip chains for passenger travel may include several modes. A work trip commute might involve driving a car to a rail or bus stop, with possibility of additional transfers within or among modes to complete the trip.

The aim of this paper is to provide a framework to address both the algorithmic approaches proposed for solving the multimodal shortest path problem arising frequently in a transit network system, and a transportation network modeling. The work proposes a specific path operator, that is an important component in GIS-transit context, and describes how to adapt transit modalities and schedules to the shortest path approaches. Both minimum time and viable path problems are analyzed. Moreover, how the multimodal routing model can be exploited to design special-purpose trip planning modules for MTS is shown.

More precisely, Section 2 reviews the literature on network data modeling, multimodal routing and GIS-transit itinerary planning applications. Section 3 develops the data model and details the network partitioning structure, used to create service districts based on accessibility, to perform time searching and to evaluate possible facility locations. Section 4 presents the components of the network solver. Section 5 formulates the routing algorithm for multimodal transportation system. Section 6 is devoted to the integration of GIS component within multimodal operators analysis. Section 7 provides conclusions.

2. Literature review

Hierarchical structure is an efficient way to model transit network system levels. Van Nes [22] presents a strategy to design multimodal network based on the concept of hierarchical network levels. In the road networks and the public transport networks a hierarchy of functionally different network levels can easily be distinguished, for example railway, arterials and freeways. Each level of the network is well suited to contribute specific journey functions, e.g. according to travel distances, and has its own quality in terms of travel speed and travel comfort. For private car, network levels as street, arterials and freeways can be distinguished, and for public transport local bus, metro, and long distance train are well known network levels. In a multimodal transport system network, there is presence of different modes, bus, metro, train, that are connected at transfer points, where service routes are fixed and the departure or arrival at certain stations is scheduled in advance and generally not subject to changes. Fig. 1 illustrates the network levels.

The first step for modeling the transit network system is to derive a model which captures all the possible transit modalities and the interconnections among them. Mainguenaud [10] presents a data model to man-
age networks with a Geographical Information System. The data model is based on the merge of graph theory concepts and the object-oriented paradigm. This strategy allows definition of a node and link as an abstraction of a subnetwork. Hence, the network can be structured in a hierarchical way following the importance of the nodes and links. Jung and Pramanik [6] developed a new graph model, called Hierarchical multilevel graph, for very large topographical road maps. This graph model provides a tool to structuring and abstracting a topographic road map in a hierarchical fashion. Jing et al. [5] proposed the HEPV (Hierarchical Encoded Path View). Their idea consists of partitioning large graph into smaller subgraphs and organizing them in a hierarchical fashion by pushing up border nodes. This approach pre-computes the shortest paths between all the member nodes (including the boundary nodes) of each subgraph only within that subgraph. Most of these researches investigated the problem of a very large volume of data they have to search. Thus, we need an efficient database organization method for structuring the multimodal transportation network and to speed up the computation of a minimum cost path. In this regard, we deal with a Path View operator that concerns data organization techniques able to pre-compute and store some partial path information. This operator uses the pre-computed partial path information to prune the search space when computing a minimum cost path. Hence, it represents an important component of our network solver.

On the other hand, to generate the optimal multimodal routing, the derived transit network model needs to take into account the set of nodes representing pre-scheduled stations, or time points, and a set of transfer links connecting two nodes from different routes. Links that made a change of mode or modal transfer must be added to the network graph, to represent the following activities: waiting for a bus or a train, boarding/alighting a bus or a train, walking between two transit stations for transfer. There have been many research efforts reported in the literature that focused on the shortest multimodal path computation problem [2,4,9,13,14,18,23]. In this class of problem we have two main constraints, the first concerns the set of transit modes used along the O–D trip, and the second is the time constraint. By viable path we mean a path that respects a constraint on its sequences of used modes. Few research introduced the definition of viable path in multimodal transit [2,9]. Fernandez et al. [4] studied the shortest path on bimodal networks. Pallottino and Scutellà [18] considered the number of modal transfers in a path as an attribute in the multicriteria shortest path. Modesti and Sciomachen [14] presented a utility measure for finding multiobjective shortest paths in urban multimodal transportation networks. Miller and Storm [13] created a modal transfer arc for representing each modal change. A characteristic of viable path is the use of distinct modes of transportation. Each user
assigns its personal attribute to $O-D$ paths in multimodal network, and he can give the maximum of the generalised cost, the modes of transport utilized, i.e., a user can choose which modes he prefers to use to reach his destination, and a number of modal transfers that he is able to establish. The multimodal shortest viable path is one of the most important current routing problem, and its solution is very interesting to travel in multimodal networks. The time constraint represents the discontinuities of fixed schedule lines, the delays at transfer points and penalties associated with turning movements. Ziliaskopoulos and Wardell [23] studied this type of constraint and presented the time-dependent least time paths algorithm on multimodal networks. Most of this research presents one routing model that take into account only one of the main constraints. The paper [9] considered only the set of mode constraints, and the routing model in research [23] studied only the time constraint and its optimal path may generate an unviable path. One of the purposes of this research is to combine the multimodal routing and the schedule model to provide an efficient routing in terms of modes and schedule.

One of the objective of many works is to integrate the routing model within Geographic Information System (GIS) technology. Thériault et al. [21] present a modeling and simulation procedure to evaluate optimal routes and to compute travel times for reaching individual trip of an $O-D$ survey database using TransCad GIS software. The procedure finds the best routes through a topological road network. Boummakoul et al. [3] formulated path-finding in terms of fuzzy networks based on semirings, and provided generic algorithmic solutions to these problems supported by Fuzzy Spatial Network Solver. The FSNet-Solver software was developed using MapObject GIS (ESRI Product) and it is an integrated component in the mobile GIS application. Li and Kurt [8] proposed a model called GIS-TIPDSS (design of transit itinerary planning system) based on three modules: input module, transit itinerary module, and output module. The input module includes passenger information and loading transit network. The transit itinerary module performs phase itinerary finding and produces the best path. The output module includes the results from the transit itinerary module and displays the best path.

In this research we presented a software system implemented under a PC Windows 98 in JAVA object-oriented programming language (SUN MICROSYSTEM Product), and the graphical user interface (GUI) was designed using JFC (JAVA Foundation Class. ©SUN Microsystems). The GUI displays a map and the software users may query the map for $O-D$ trip information such as transit modes, schedules, and more. The software users are able to set up a favourite point alias to frequented map locations or to add incidents to model route hazards (congestion, accidents, etc.).

3. Network data model

This paragraph describes the network data model used to structure the transportation network in a hierarchical fashion.

Three levels are defined to manage network oriented-data: physical level (i.e. spatial coordinates), logical level (i.e. graph modeling the transportation network), and applicative level (i.e. the structuring of a graph taking into account application dependent modeling).

Several models have been proposed to model geometrical data and application dependent-data or both. Since graphs are a special concept for representing such relationships, few graph data models [12] have been proposed, particularly in a Geographical Information System context. The application of graph data models can be found, for example, in the definition of EDIGéO or in GDF.

In general, data models have to provide facilities to represent relationships among objects. In many cases, it is helpful to view such relationships as graph structures. Then, many queries can directly be mapped to well-known graph problems for which efficient algorithms exist; e.g. route finding is solved by shortest path problem.

In the case under study the transit network is modeled with a hierarchical structure. We find two main levels: the national network that connects cities between them, and the urban network. The aim is the use of
a data model to manage transit network with Geographical Information System. The data model is described by logical view (see Fig. 5). Nodes and edges may carry geometric information, for example, a point value may be associated with a node and a polygonal line with an edge. A node (for generic topology) is the smallest identified location in space, and it plays many different roles in a transport network (node is not just a location in space). A link is an important component of the multimodal transit system, and it may represent both the type of road and the mode of transport between two nodes. The introduction of the hierarchical graph model provides a strategy of abstracting and structuring a transit network in a hierarchical fashion. The idea is to deal with a special data structure that allows efficient traverses between network levels. Graph operations must be implemented on the basis of efficient graph algorithms. We integrate a proposed multimodal shortest path algorithm within spatially embedded networks as well to obtain more efficient long-run planning of transit on multimodal networks systems.

A transit network can be viewed as a directed graph $G(V, E)$, where each node in $V$ represents network objects (i.e. the bus station, parking). Edges $(x, y)$ in $E$ correspond to the connections between the nodes $x$ and $y$ in $V$. Suppose that $G(V, E)$ is partitioned into a set of subgraphs such that

$$V_1 \cup V_2 \cup \ldots \cup V_m = V; E_1 \cup E_2 \cup \ldots \cup E_m \subseteq E,$$

$$V_i \cap V_j = \emptyset \quad \text{and} \quad E_i \cap E_j = \emptyset,$$

where

$$1 \leq i, \quad j \leq m \quad \text{and} \quad i \neq j.$$

**Definition 3.1.** For subgraphs $SG_i(V_i, E_i)$, $SG_j(V_j, E_j)$, where $i \neq j$, let $CON(SG_i, SG_j)$ denote the set of connections between $V_i$ and $V_j$. $CON(SG_i, SG_j) = \{(x, y)/x \in V_i, y \in V_j\}$.

**Definition 3.2.** Given a collection of subgraphs $SG = \{SG_1, SG_2, \ldots, SG_m\}$ let $OutEdges(SG_i)$ denote the set of edges leaving $SG_i$.

**Definition 3.3.** Given a collection of subgraphs $SG = \{SG_1, SG_2, \ldots, SG_m\}$ we denote by $BN(SG_i)$ the set of vertices of $V_i$ that have at least one incoming or outgoing edges in $\cup_{1 \leq j \neq i \leq m} CON(SG_i, SG_j)$. Then, $BN(SG_i)$ is called boundary nodes.

See example in Fig. 2, and consider subgraphs $SG_1$ and $SG_2$. We have $BN(SG_1) = \{4, 5, 6\}$, $BN(SG_2) = \{9, 7, 8\}$, $OutEdges(SG_1) = \{(5, 7), (6, 8), (4, 9)\}$, $OutEdges(SG_2) = \emptyset$.

The above definitions can be generalized into multi-levels. By observing the structure of the whole network, we know that all subgraphs are related to each other in a complete balanced tree structure. The root node of the tree is the graph $G$ and each tree’s level represents an abstraction level of the transit network.

Since transportation network is modeled with a hierarchical structure, then, the data model presented is well adapted to manage several levels of abstraction. This network modeling elaborates different levels of
details following the importance of the nodes and the links. The principal classes predefined are: Node, Link and Network. For each of them we give its form using C++ class syntax.

3.1. Main classes

3.1.1. Nodes
The class Node is given as follows:

\[
\text{Template } <\text{class } T> \text{ class Node } \{ \\
T* \text{ Label; } \\
\text{Node}<T>( ) \{ \\
\quad \text{Label} = \text{new } T; \\
\} \\
\};
\]

The use of the template class \( T \) is very interesting in routing problems, for example class \( T \) may represent the set of constraints to be associated with the class Node. On the other hands, the class Node plays an important role in the network modeling; it may represent several points in the different elements of the active network. Fig. 3 gives an example of node representation.

3.1.2. Links
The class Link is defined to model, rail, road and transport mode. It is given as follows:

\[
\text{Template}<\text{class } T> \text{ class Link } \{ \\
\text{Node}<T>* \text{ Initial}_\text{Node}; \\
\text{Node}<T>* \text{ Terminal}_\text{Node}; \\
\text{Link}<T>( ) \{ \\
\quad \text{Initial}_\text{Node} = \text{new } \text{Node}<T>; \\
\quad \text{Terminal}_\text{Node} = \text{new } \text{Node}<T>; \\
\} \\
\};
\]

![Diagram](image-url)  
Fig. 3. Node may represent several elements of the multimodal network.
Fig. 4 illustrates an example of link representation. The link may represent transit modes, road network and rails.

3.1.3. Networks

A graph is an efficient concept to model network oriented data. Then, the class Network is defined as follows:

Fig. 4. Link representation.

Fig. 5. Data model representation.
3.2. Abstraction levels

Transit network is structured in a hierarchical fashion. The abstraction levels are used to take into account the network-oriented data. In particular, the HyperNode class (in respect of the HyperLink class) is considered to allow the definition of a node (in respect of a link) as an abstraction of subnetwork.

3.2.1. Sub_networks

The class Sub_network is a subclass of the class Network. The concept Out_Edges in Definition 3.2 is used to connect this network to the various levels of abstraction. This class is defined by an aggregation of two attributes.

Template<class T> Sub_network: public Network {
    list<Node<T>*>* ListOfBorderNodes;
    list<Link<T>*>* ListOfOutEdges;
    Sub_network<T>() {
        ListOfBorderNodes = new list<Node<T>*>;
        ListOfOutEdges = new list<Link<T>*>;
    }
};

3.2.2. HyperNetwork

The class HyperNetwork introduces in the abstraction level, in which we aggregated a set of HyperNodes and a set of HyperLinks.

Template<class T> class HyperNetwork {
    list<HyperNode<T>*>* ListHyperNodes;
    list<HyperLink<T>*>* ListHyperLinks;
    HyperNetwork<T>( ) {
        ListHyperNodes = new list<HyperNode<T>*>;
        ListHyperLinks = new list<HyperLink<T>*>;
    }
};
3.2.3. HyperNodes
The class HyperNode inherits from the class Node. This class may have several associated subnetworks.

```cpp
Template<class T> class HyperNode: public Node {
    list<Sub_network<T>*>*ListOfSub_networks;
    HyperNode<T>( ) {
        ListOfSub_networks = new list<Sub_network<T>*>;
    }
};
```

3.2.4. HyperLinks
The class HyperLink is a subclass of the class Link. It has a unique associated subnetwork.

```cpp
Template<class T> class HyperLink: public Link {
    Sub_network<T>*Net;
    HyperLink<T>( ) {
        Net = new<Sub_network<T>*>;
    }
};
```

Fig. 5 describes the data model concepts hierarchy, and relationships between classes.

4. Network solver
A Geographical Information System (GIS) provides the opportunity to manage network facilities. Network solver model (Fig. 7) has taken into account the manipulation of the different levels of abstraction. We have basic components directly linked to the concept of abstraction in the data model, elementary operators correspond to the manipulations of graphs and subgraphs, and high level operators correspond to the GIS user interface.

Langou [7] defined two operators directly linked to the concepts of abstraction in the data model: the DEVELOP operator and the UNDEVELOP operator. The DEVELOP operator provides more specific details by merging a subnetwork associated with a HyperNode or with a HyperLink. On the other hands, we have the conversion operator; UNDEVELOP operator provides a more restricted graph by the replacement of subnetwork by the HyperNode or the HyperLink. The main operator considered at high level is the evaluation path operator, between an origin and destination pair. Path operator can be classified into categories of network-oriented operators based on graph manipulation. Many current research

---

**Path View** ($G_o$ ($SG_i$)) $: [G_f]$  
Where $G_o$ : a initial graph with the set of nodes and edges to be developed  
($SG_i$) : a set of subgraphs associated with nodes and edges  
$G_f$ : a result graph constructing by Path View edge sets and OutEdges sets

Fig. 6. The signature of the Path View operator.
[3,11] considers that the path evaluation operator is one of the most important operators in a Geographical Information System. The paper [11] precis[es that the user's queries contain many implicit constraints that are not specified while defining a query (e.g. I would like to go from one place to another one). The work introduces the characterization of database alphanumeric attributes in the following classes: Neutral, Time, Space and Applicative_unit. The required database modeling must take into account these constraints.

The evaluation of path is performed with a path operator. The specification of the path operator that we consider in this work is this evaluation that corresponds to the detection of a multimodal shortest path between an origin node and a destination node, based on our algorithmic approach proposed in the following paragraph. This path can be a direct link (i.e, a successor in a graph) or a more complex path (i.e. a transitive closure of a graph).

In this work, we define a new operator called Path View (PV) operator, used to elaborate the shortest path in abstraction levels and it can be used in routing to dramatically improve performance.

Each subgraph is described and identified by its boundary nodes since they exclusively belong to one subgraph.

**Definition 4.1.** Given a subgraph $SG$, we have

$$PV_{SG} = \{(x,y,f_{x,y})/ (x,y) \in (BN(SG) \times BN(SG))\}$$

is the Path View edge set.

Function $f_{x,y}$ defines the shortest path from boundary node $x$ to boundary node $y$ only within the subgraph $SG$.

It is easy to see that the size of the whole graph depends on the size of the abstraction levels. Thus, for an efficient routing, we need to minimize the space of search for shortest path computation. Then, based on Definition 4.1, each subgraph $SG$ is reduced to a Path View edge set ($PV_{SG}$). A new graph considered is

![Fig. 7. Network solver model.](image-url)
defined in terms of OutEdges sets (Definition 3.2) and Path View edge sets. The optimality of the shortest path cost computed on the new graph is given by the following theorem.

**Theorem 4.1.** Let a collection of subgraphs \( \{SG_1, SG_2, \ldots, SG_m\} \) be a partition of the graph \( G \).

For any pair \( x \in SG_i \) and \( y \in SG_j \), we have \( SP_D(x, y) = SP_D(x, y) \), where

\[
D = SG_i \cup SG_j \cup (\cup_{1 \leq k \neq i,j \leq m} PV_k) \cup (\cup_{1 \leq k \neq i,j \leq m} \text{ OutEdges}_k).
\]

\( SP_D(x, y) \) represents the shortest path within the graph \( G \).

\( SP_D(x, y) \) represents the shortest path on the new graph.

\( PV_k \) represents the Path View edge set within the subgraph \( SG_k \), for \( k \in \{0, 1, \ldots, m\} \).

**Proof.** The proof is given in the paper [6]. \( \square \)

From this theorem, we find how a Path View operator contributes to reduce the search space, issue from the abstraction levels. That is, without using this operator the search space would be the whole graph \( G \) (see Fig. 6).

### 5. Routing algorithm for a multimodal transportation system

Now we are interested in finding the shortest path in large cities that have several travel modes taking passengers from one given place to another. We model transit network as a digraph \( G = (V, E) \), where \( V \) is the set of nodes and \( E \) is the set of arcs. We set \( V = V_P \cup V_B \cup V_M \cup V_A \cup V_T \), and \( E_P \cup E_B \cup E_M \cup E_A \cup E_T \subseteq E \). Where \( V_m \) and \( E_m \) are the nodes and arcs associated with the transit mode \( m \in Md \), where \( Md \) denote the set of transit modes. The main modes considered in the multimodal transit network are presented in the following table:

<table>
<thead>
<tr>
<th>Travel mode in city</th>
<th>Travel mode between cities</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P ): Private vehicle</td>
<td>( P ): Private vehicle</td>
</tr>
<tr>
<td>( M ): Metro</td>
<td>( T ): Train</td>
</tr>
<tr>
<td>( B ): Bus</td>
<td>( A ): Autobus</td>
</tr>
<tr>
<td>( W ): Walking</td>
<td></td>
</tr>
</tbody>
</table>

The private vehicle mode can be common mode between the different level of the network. Otherwise, the change from one network level to another level is transparent for private car, but for public transport travelers such as a transfer is quite noticeable.

In a multimodal network the users have the possibility to commute from one modality to another. Thus, we have two distinct type of arcs, one is called transfer arc, it represents a change of mode or modal transfer, that may be executed by walking or by waiting, and travel arc that connects two nodes by only one travel mode. The set of direct modal transfer arcs is defined as:

\( Ts = \{ (v_i, v_j) : v_i \in V_i, v_j \in V_j \text{ such that } i \in Md, j \in Md \backslash \{P\}, \text{ and } i \neq j \} \). Hence \( Ts \) represents all allowed commutations between the considered transportation modalities. So now \( E = E_P \cup E_B \cup E_M \cup E_A \cup E_T \cup Ts \). Each arc has a cost associated with it given by the time required to travel from node \( v \) to node \( u \). Suc(\( u \)) represents the set of successor nodes of node \( u \), and Pred(\( u \)) the set of predecessor nodes to node \( u \). In the next sections, the formulation of the multimodal viable path and, respectively, the time-constrained are presented.
5.1. Multimodal viable path

A characteristic of viable path is the use of distinct modes of transportation. It is noticeable that no modal transfer allows to transfer from public modes to private mode since it is logic that once a path is not started with the private modality it is not feasible anymore to take it for reaching the destination. For this reason an O–D feasible path that uses the private mode, it is evident to take it at the origin node O. Then, the viable path is a path that respects a set of constraints on its sequences of used modes. Furthermore, a viable path must respect the number of modal transfer that a user is able to establish. In the following, the private mode, metro mode and the train mode are considered as a constraint modes.

An O–D path within an urban network is considered viable if it include only one consecutive sequence of metro mode, or only one consecutive sequence of private mode with initial node O. As the same way, one defines a viable path at the higher level (between cities), the private mode is a common mode between network levels, so we say that a viable path is a path that contains only one consecutive sequence of train mode.

On the other hand, a path composed only by one travel mode is a trivial viable path, then, the viability is directly related to the number of modal transfers established during the trip; more the number of transfers is important more the chance that the path is not viable.

For a precise viable path formulation, some definitions are given in below. Let \((v, u, w)\) denote a node-triplet on the graph \(G\); by the notation \(Transit(v, u, w)\) we mean the transit from node \(v\) through node \(u\) to node \(w\).

**Definition 5.1.1.** Let \((v, u, w)\) be a triplet of nodes on the graph \(G\).

\(Transit(v, u, w)\) is **Monomodal**, if the arc \((v, u)\) and \((u, w)\) are associated with the same transit mode \(m \in Md\).

**Definition 5.1.2.** Let \((v, u, w)\) be a triplet of nodes on the graph \(G\).

\(Transit(v, u, w)\) is **Begin ModalTransfer**, if the arc \((v, u)\) is associated with a transit mode \(m \in Md\), and the arc \((u, w)\) is an element of \(Ts\).

**Definition 5.1.3.** Let \((v, u, w)\) be a triplet nodes on the graph \(G\).

\(Transit(v, u, w)\) is **End ModalTransfer**, if the arc \((v, u)\) is an element of \(Ts\), and the arc \((u, w)\) is associated with a transit mode \(m \in Md\).

Fig. 8 illustrates examples for the above definitions.
Let \((v, u, w)\) be a node-triplet on the graph \(G\). As a consequence from above assumptions, we have the following results: if \(\text{Transit}(v, u, w)\) is \textbf{Monomodal}, then, the \(\text{Transit}(v, u, w)\) is viable. On the other hand, the viability of the transit from node \(v\) through node \(u\) to node \(w\) must be controlled if \(\text{Transit}(v, u, w)\) is \textbf{End ModalTransfer}.

Finally, if \(\text{Transit}(v, u, w)\) is \textbf{Begin ModalTransfer}, the number of modal transfers is incremented and tested if not superior of the maximum transfers given by the user.

Based on the definition above of the viable path, we define the strategy to be used for detecting the viability of the \(\text{Transit}(v, u, w)\) as follows.

One assumption made is that the path \(\Pi\) from origin node \(O\) to destination node \(D\) is a sequence of \(\text{Transit}(v, u, w)\). In the case where the path \(\Pi\) contains only one arc \((O, D)\) it is obviously a viable path. We considered the set of transit modes \(Md = \{\text{Private, Bus, Metro, Autobus, Train}\}\). Each arc \((v, u)\) is associated with a mode, denoted by \(\text{mode}(v, u) \in Md \cup Ts\). \(\text{Mod}(v, u)\) represents the set of modes used in the current path \(\Pi\) from the origin node \(O\) to node \(u\) and \(\text{Transf\_Number}(v, u)\) is the number of modal transfers in the current path from origin \(O\) to node \(u\).

In the case where the \(\text{Transit}(v, u, w)\) is \textbf{End ModalTransfer}, we use the strategy given in Fig. 9.

Note that the arc \((0, O)\) in the beginning of the trip, is associated with \(\text{Mod}(0, O) = O\). Leaving one transit mode \(m\) in the path \(\Pi\) is denoted by \(\text{Um}\), it means that the transit mode \(m\) was used in the current path \(\Pi\).

So now if the \(\text{Transit}(v, u, w)\) is \textbf{Begin ModalTransfer}, we must test if the number of modal transfers associated with the arc \((v, u)\) is lower than the maximum of transfers given by the user (\(\text{Max\_transfer}\)). Then, the number of modal transfers established during the path \(\Pi\) to arrive at node \(w\) is incremented, and marked that the transit mode associated with the arc \((v, u)\) was used to arrive at node \(w\). These steps are defined in Fig. 10.

\[
\text{Switch(mode}(u, w))
\]

\[
\text{Begin}
\]

\[
\text{Case(Private):}
\]

\[
\text{If(\text{Mod}(v, u) = O or \text{Mod}(v, u) = P)}
\]

\[
\text{Transit}(v, u, w) \text{ is viable and Mod}(u, w) = P
\]

\[
\text{Break;}
\]

\[
\text{Case(Metro):}
\]

\[
\text{If(\text{Mod}(v, u) = O, UB, UP, UA, UT, or M)}
\]

\[
\text{Transit}(v, u, w) \text{ is viable and Mod}(u, w) = M
\]

\[
\text{Break;}
\]

\[
\text{Case(Train):}
\]

\[
\text{If(\text{Mod}(v, u) = O, UM, UB, UA, UP or T)}
\]

\[
\text{Transit}(v, u, w) \text{ is viable and Mod}(u, w) = T
\]

\[
\text{Break;}
\]

\[
\text{Default:}
\]

\[
\text{Case(Autobus) resp. Bus:)}
\]

\[
\text{If(\text{Mod}(v, u) = O)}
\]

\[
\text{Transit}(v, u, w) \text{ is viable and Mod}(u, w) = A
\]

\[
\text{(resp. Transit}(v, u, w) \text{ is viable and Mod}(u, w) = B)}
\]

\[
\text{Else}
\]

\[
\text{Transit}(v, u, w) \text{ is viable and Mod}(u, w) = \text{Mod}(v, u)
\]

\[
\text{Break;}
\]

\[
\text{End}
\]

Fig. 9. Viable path procedure for \textbf{End ModalTransfer} transit.
Example ‘a’ in Fig. 11 illustrates a viable $O$–$D$ path, two constrained modes are used to travel from origin node $O$ to destination node $D$, the metro mode and train mode, and the $O$–$D$ path respects the constraint on its sequence of used modes. But the $OD$ path in example $b$ is not a viable path, because the path don’t start at origin node $O$ with the private mode. See Fig. 16 for arcs details.

5.2. Time-constrained

In a multimodal network, scheduled transit mode lines serve several stations, and each one is associated with related scheduled departures. Then, our goal is to deal with an efficient schedule model, where each station has a set of scheduled departures serving next stations. A node on the graph $G$ can model station, parking, light control, and more.

From the above, we assume that each station $u$ is associated with a set of scheduled Departure Lists denoted by

$$DL(u) = \{D_{0,u}, D_{1,u}, \ldots, D_{r,u}\},$$

where $w \in \text{Suc}(u)$ and $D_{i,u}$ represents the $i$th scheduled departure serving the arc $(u, w)$. Let $(v, u, w)$ be a node-triplet on the graph $G$. We denote by $\text{Time}_m(v, u)$ the time required to travel from node $v$ to node $u$ on mode $m \in \text{Md} \cup Ts$. $\text{Delay}(v, u, w)$ denoted the delay at node $u$, when traveling from node $v$ through node $u$ to node $w$.

In the case where $\text{Transit}(v, u, w)$ is Monomodal, $\text{Delay}(v, u, w)$ represents the time penalties associated with the turning movement. If $\text{Transit}(v, u, w)$ is End ModalTransfer, then $\text{Delay}(v, u, w)$ is the waiting time until the coming scheduled departure. Finally, if $\text{Transit}(v, u, w)$ is Begin ModalTransfer, we assume that leaving the transit mode associated with the arc $(v, u)$ is accomplished without delays.

For each node $u$, we associated a list of arrival times at node $u$, $AL_u = \{\text{Arr}_m(v, u)\}_{1 \leq i \leq K, v \in \text{Pred}(u)}$, where $\text{Arr}_m(v, u)$ is the $i$th arrival time at node $u$ through an arc $(v, u)$ on mode $m \in \text{Md} \cup Ts$, and a list of the

![Fig. 10. Procedure of viable path for Begin ModalTransfer transit.](image)

![Fig. 11. Illustration of viable and unviable paths: (a) Viable $O$–$D$ path with 2 modal transfers and (b) Unviable $O$–$D$ path.](image)
departure times from the node \( u \) denoted by \( DT_u = \{ \text{Dep}_{m,n}(v,u,w) \} \) \( 1 \leq i \leq K_u \) \( \in \text{Pred}(u) \), where \( \text{Dep}_{m,n}(v,u,w) \) is the \( i \)th departure time from node \( u \), provided that we visit it through arc \( (v,u) \) on mode \( m \) and the next arc to travel on mode \( n \) is \( (u,w) \), with \( m,n \in Md \cup Ts \). Fig. 12 illustrates the arrival time, the departure time, the delays and the travel time associated with a End ModalTransfer.

Now we define a strategy to calculate \( \text{Dep}_{m,n}(v,u,w) \) and \( \text{Delay}(v,u,w) \), in the case where \( \text{Transit}(v,u,w) \) is End ModalTransfer, where \( m \in Md \setminus \{ P \} \cup Ts \), \( n \in Md \setminus \{ P \} \), and \( \text{Arr}_m(v,u) \) is the arrive time at node \( u \) on mode \( m \). Let \( DL(u) = \{ D_{0,u}, D_{1,u}, \ldots, D_{r,u} \} \) be a list of scheduled departures associated with the transit mode station \( u \) and serving the arc \( (u,w) \). The procedure is done in Fig. 13.

Consider the time intervals ( \( [D_{l-1,u}, D_{l,u}] \) ) \( 1 \leq l \leq r \).
If\( \text{Arr}_m(v,u) < D_{0,u} \)
Begin
\( \text{Dep}_{m,n}(v,u,w) = D_{0,w} \) and \( \text{Delay}(v,u,w) = D_{0,w} - \text{Arr}_m(v,u) \)
End
Else
If\( \text{Arr}_m(v,u) > D_{r,u} \)
Begin
\( \text{Dep}_{m,n}(v,u,w) = \infty \) and \( \text{Delay}(v,u,w) = \infty \)
End
Else
Begin
Find integer \( i \) such that \( \text{Arr}_m(v,u) \in [D_{i-1,u}, D_{i,u}] \).
If\( \text{Arr}_m(v,u) = D_{l-1,u} \) or \( \text{Arr}_m(v,u) = D_{l,u} \)
Begin
\( \text{Dep}_{m,n}(v,u,w) = \text{Arr}_m(v,u) \) and \( \text{Delay}(v,u,w) = 0 \)
End
Else
If\( \text{Arr}_m(v,u) > D_{l-1} \) and \( \text{Arr}_m(v,u) < D_{l,u} \)
Begin
\( \text{Dep}_{m,n}(v,u,w) = D_l \) and \( \text{Delay}(v,u,w) = D_{l,u} - \text{Arr}_m(v,u) \)
End
End

Fig. 12. Representation of labels associated with a End ModalTransfer.

Fig. 13. Procedure to update schedule model.
This procedure can be done in time $O(\log r)$ by binary search, where $r$ is the number of scheduled departures.

Fig. 14 illustrates an example of transit that can be modeled by `Begin ModalTransfer`. The situation represents one user driving its car from a given place to the parking area, and walking to another place. The time required to find one place in the parking is added to the private arc time, and the operation of quitting the private car is made without delays.

Fig. 15 illustrates an example of transit that may be represented by `End ModalTransfer`. It corresponds to the situation when one user walks from a given place to the metro station, and rides a metro to another metro station close to the destination. The metro station is associated with a list of scheduled

---

**Fig. 14.** `Begin ModalTransfer` representation.

**Fig. 15.** `End ModalTransfer` representation.

---

**Fig. 16.** Illustration of modes and stations.
departures, then, the waiting time until the coming scheduled departure is determined by the formulation in Fig. 13.

5.3. Design of the K-multimodal shortest path algorithm

The ranking of shortest paths is considered as a generalisation of the well known shortest path problem [17–19], since several paths must be listed by nondecreasing order of their costs. The algorithm that we present is a modified version of our K-shortest path algorithm [15] to which we integrated the proposed design concerning the multimodal viable path and the time constraints defined above, in order to define an efficient solution for multimodal shortest path problem. For more details about K-shortest paths, the reader can refer to the papers [20,24].

The labels functions are composed of the following elements:

$h$ is the number of modal transfers in the current path.
$X$ is the set of arcs with $h$ modal transfers.
$Y$ is the set of arcs with $h + 1$ modal transfers.

The $k$th component of the $K$-tuple $(\lambda(v, u)^1, \lambda(v, u)^2, \ldots, \lambda(v, u)^K)$ is a predecessor arc of the arc $(v, u)$ in one shortest path (from $K$-shortest path).

The $k$th component of the $K$-tuple $(\Theta(u, w)^1, \Theta(u, w)^2, \ldots, \Theta(u, w)^K)$ is the position of the arc $\lambda(v, u)^k$ in the $K$-tuple $(\lambda(v, u)^1, \lambda(v, u)^2, \ldots, \lambda(v, u)^K)$.

The $k$th component of the $K$-tuple $(LastLabel(v, u)^1, LastLabel(v, u)^2, \ldots, LastLabel(v, u)^K)$ is the last label (arrival time) corresponding to a number of transfers lower than the current $h$.

In consistency with the works of Pallottino et al. [18] and Storchi et al. [9], we introduce the $K$-tuple $(LastLabel(v, u)^1, LastLabel(v, u)^2, \ldots, LastLabel(v, u)^K)$ in order to add more efficiency for the algorithm. We suggest checking dominance when selecting the node $u$ from the set $X$, then comparing the current arrival time to node $u$ with the last label selected relative to node $u$ (i.e., $LastLabel(v, u)^k$). If the arrival time $Arr_m(v, u)^k$ with $h$ modal transfers is greater than or equal to the last arrival with less than $h$ transfers, that is $Arr_m(v, u)^k \geq LastLabel(v, u)^k$, then we discard it since it is a dominated label. Otherwise, that is if $Arr_m(v, u)^k < LastLabel(v, u)^k$, then the label is not dominated, hence $Arr_m(v, u)^k = LastLabel(v, u)^k$.

### Algorithm

1. Mark $Arr_m(v, u)^r$ as unused and set $Arr_m(v, u)^r = \infty$; $\forall r \in \{1, \ldots, K\}$ $\forall u \in V$, $v \in \text{Pred}(u)$, $m \in Md \cup Ts$. Set $Arr_m(0, O)^1 = 0$. Insert $O$ into the set $X$.
2. Take node $u$ from $X$, and set $X = X \setminus \{u\}$.
3. For each $k \in \{1, \ldots, K\}$ such that $Arr_m(v, u)^k$ is unused and finite
   - If ($Arr_m(v, u)^k < LastLabel(v, u)^k$).
     - Begin
       - $LastLabel(v, u)^k = Arr_m(v, u)^k$;
       - For each arc $(u, w)$ emanating from node $u$ do
         - If (Transit$(v, u, w)$ is viable)
           - Begin
             - $Dep_{m,n}(v, u, w)^k = Arr_m(v, u)^k + Delay(v, u, w)$
             - $Temp(u, w) = Dep_{m,n}(v, u, w)^k + Time(u, v)$
             - If ($Temp(u, w) < \max\{Arr_m(u, w)^s, 1 \leq s \leq K\}$)
               - Begin
                 - $Dep_{m,n}(v, u, w)^k = Arr_m(v, u)^k + Delay(v, u, w)$
                 - $Temp(u, w) = Dep_{m,n}(v, u, w)^k + Time(u, v)$
                 - $LastLabel(v, u)^k = Arr_m(v, u)^k$;
                 - For each arc $(u, w)$ emanating from node $u$ do
                   - If (Transit$(v, u, w)$ is viable)
                     - Begin
                       - $Dep_{m,n}(v, u, w)^k = Arr_m(v, u)^k + Delay(v, u, w)$
                       - $Temp(u, w) = Dep_{m,n}(v, u, w)^k + Time(u, v)$
                       - $LastLabel(v, u)^k = Arr_m(v, u)^k$;
                       - For each arc $(u, w)$ emanating from node $u$ do
                         - If (Transit$(v, u, w)$ is viable)
                           - Begin
                             - $Dep_{m,n}(v, u, w)^k = Arr_m(v, u)^k + Delay(v, u, w)$
                             - $Temp(u, w) = Dep_{m,n}(v, u, w)^k + Time(u, v)$
                             - $LastLabel(v, u)^k = Arr_m(v, u)^k$;
                           - End
                         - End
                       - End
                     - End
                   - End
                 - End
               - End
             - End
           - End
         - End
       - End
     - End
   - End
- End
Begin 
\[
    r \leftarrow \text{order of } \max\{\text{Arr}_m(u,w)^s, 1 \leq s \leq K\}
\]
\[
    \text{Arr}(u,w)^r = \text{Temp}(u,w); \lambda(u,w)^r = (v,u);
\]
\[
    \Theta(u,w)^r = k; \text{Arr}(u,w)^r \text{ is unused};
\]
If \( w \notin X \) then \( X = X \cup \{w\} \)
End
End
Else

If \( (\text{Transit}(v,u,w) \text{ is } \text{Begin ModalTransfer}) \) Begin 
\[
    \text{Arr}(u,w)^k \text{ is used}
\]
\[
    \text{Dep}_m(v,u,w)^k = \text{Arr}_m(v,u)^k
\]
\[
    \text{Temp}(u,w) = \text{Dep}_m(v,u,w)^k + \text{Time}(u,v)
\]
If \( (\text{Temp}(u,w) < \max\{\text{Arr}_m(u,w)^s, 1 \leq s \leq K\}) \) Begin 
\[
    r \leftarrow \text{order of } \max\{\text{Arr}_m(u,w)^s, 1 \leq s \leq K\}
\]
\[
    \text{Arr}(u,w)^r = \text{Temp}(u,w); \lambda(u,w)^r = (v,u);
\]
\[
    \Theta(u,w)^r = k; \text{Arr}(u,w)^r \text{ is unused};
\]
If \( w \notin Y \) then \( Y = Y \cup \{w\} \)
End
End

4- If \( X \) is empty then insert all element of the set \( Y \) in \( X \), and set \( h = h + 1 \).
5- If \( X \) is not empty and \( h \) inferior or equal to \( \text{Max TRANSF} \) then go to step 2, else stop the algorithm.

Note that the viability of the transit from node \( v \) to node \( w \) through node \( u \) is tested in respect to \( \text{Transit}(v,u,w) \). If \( \text{Transit}(v,u,w) \) is \( \text{End ModalTransfer} \), we use the formulation in Fig. 9. The formulation in Fig. 10 is used if \( \text{Transit}(v,u,w) \) is \( \text{Begin ModalTransfer} \).

5.4. Illustration and computational experience of the algorithm

The example shown in Fig. 17 illustrates a practical performance of the algorithm. This network is modeled by nodes, and links. A node represents station, and link is a transport mode between two stations. In this case the transfer links are considered as a mode. The modes set of urban network include metro, bus. The private vehicle is common between network levels. A train and autobus modes are used to travel between cities (see Fig. 16). Station is associated with a list of scheduled departures (see Fig. 18). The private nodes are associated with delays. The number along each arc is the arc’s travel time.

In Fig. 18 the \( DL(v) \) list of scheduled departures associated with node \( v \) is reported. Private nodes 4 and 7 are associated with delays, \( \text{Delay}(O, 4, 10) = 0.04 \) unit, and \( \text{Delay}(4, 7, D) = 0.05 \) unit. The execution of the algorithm determines the \( K \)-best paths following the number of modal transfers given by the user and respecting the scheduled departures associated with each transit mode station.

The algorithm finds one shortest path accomplished by the private mode from the origin node \( O \) to the destination node \( D \), and has a travel time equal to 5.29 units, when no modal transfer is admitted by the user.
Two shortest paths are determined by the algorithm, that having the same travel time, these paths differs by the upper limit of modal transfer and the waiting time until the coming train (in node 13). One path has two modal transfers and a waiting time equal to 0.1 unit, and the other has three modal transfers and a waiting time equal to 0.21 unit.

For two modal transfers the shortest path given by the algorithm has the travel time equal to 5.10 units.
When three modal transfers are admitted, the following is the shortest path with travel time of 5.10 units.

To obtain the time complexity of the algorithm, recall that it involves operations if initialization (step 1), take one node and remove it from the scanning list (step 2). Three operations: viability tests, decrease-values, and computing delays (step 3). Finally, insertion of all elements that are visiting by one modal transfer into the scanning list, and respecting the constraint of the maximum number of modal transfer given by the user (steps 4–5).

It is clear that the step 3 determines the major complexity of the algorithm. Two main operations, namely, decrease-values, and computing delays jointly given the complexity of this step. We need to examine $N$ nodes each with at most $M$ arcs to update at most $K$-values. Therefore, the complexity of this part is $O(KMN)$. Next, for each arc $(u, w)$ we need to compute the delay time by calling the procedure in Fig. 13. Since there are $O(KMN)$ examinations and each call is done by $O(\log r)$, then the complexity of this step is $O(KMN \log r)$. Furthermore, in the start of the algorithm at most $N - 1$ nodes are inserted into the scanning list, the source node will not re-inserted into this set. Of the $N - 1$ nodes initially inserted, the one with

<table>
<thead>
<tr>
<th>$D_L(v)$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>1</td>
<td>1.00</td>
<td>1.15</td>
<td>1.30</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1.00</td>
<td>1.15</td>
<td>1.30</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.85</td>
<td>1.05</td>
<td>1.25</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>1.35</td>
<td>1.55</td>
<td>1.75</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>1.70</td>
<td>2.20</td>
<td>2.70</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>2.95</td>
<td>3.45</td>
<td>3.95</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>2.00</td>
<td>2.50</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>2.05</td>
<td>2.55</td>
<td>3.05</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>3.15</td>
<td>3.65</td>
<td>4.15</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>3.00</td>
<td>3.50</td>
<td>4.00</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>4.05</td>
<td>4.25</td>
<td>4.45</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>4.15</td>
<td>4.30</td>
<td>4.45</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>4.17</td>
<td>4.32</td>
<td>4.47</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>4.00</td>
<td>4.20</td>
<td>4.40</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>4.50</td>
<td>4.70</td>
<td>4.90</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>4.75</td>
<td>4.90</td>
<td>5.05</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>5.35</td>
<td>5.50</td>
<td>5.65</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>5.85</td>
<td>6.00</td>
<td>6.15</td>
</tr>
</tbody>
</table>

Fig. 18. Representation of scheduled departures of transit mode stations.
the least label will be updated permanently. On the other hands, by scanning all \( N - 1 \) nodes of the scanning list, in \( N - 1 \) repetitions of the step 3, at least one label will be set permanently. Since initially we have at most \( KM \) labels that can be improved, then the total time is \( O(K^2 M^2 N \log r) = O(M^2 N) \).

To demonstrate the computational efficiency of the proposed algorithm and to provide some evidence on the benefit of the adaptive routing strategy, the algorithm was coded in C++, and we use STL (Standard Template Library) data structure for maintaining the scanning sets. We tested on a set of networks with 10 \( \times \) 5, 10 \( \times \) 10, 10 \( \times \) 20, 20 \( \times \) 25, 50 \( \times \) 20 nodes. For reason of efficient computation, the links in the networks are associated with appropriate modes to given more viable paths, in the problem encountered with random grid networks. All tests were performed under Microsoft Windows operating environment on a Pentium II with 64 MB RAM.

Table 1 illustrates the results of scanning time associated with each class of network, time intervals associated with each transit mode stations is fixed. The results presented linear increase of the scanning time with the number of nodes.

6. GIS-Trip planning application

Many researchers [1,16,26] have identified several advantages of using GIS for transportation modeling. The primary advantages include analytical capabilities, visual power, efficiency of data storage, integration of spatial databases, and capabilities for spatial analysis. Transportation demand analysis has been greatly enhanced by the use of GIS. The graphical, map-based interface provided by GIS enhances data input and management capabilities. GIS data aggregation functions can be used to easily assign demand characteristics to nodes on a transportation network. Once transportation demand indices have been associated with nodes, the data can be ported to an urban transportation planning system (UTPS) package such as TransModel.

In this paragraph the specification and implementation of the software environment are presented. The main functionality provided by this system is the ability to support the traffic planning throughout several phases and tools that are demanded to analyze traffic systems. The software has been implemented under a PC Windows 98 in JAVA object-oriented programming language (©SUN MICROSYSTEM). A common user interface, a data model that is integrated into a general object model frame for a broad range of objects dealing with the topological network, as well as powerful inter-tool communication facility intended to support scheduling model, are a few of the distinct characteristics of the software environment. Furthermore, the software environment model supports both microscopic as well as macroscopic traffic planning tools.

In multimodal traffic systems planning, one has to deal with models of the entire city, which include road network, transit mode stations, transit mode itineraries, parking area, demand profile from traffic sources to sinks, intersection control. The data model (Fig. 3) is implemented in a database and becomes available to the integrated tools in an organized way. The data interface, at the intermediate level, manages the communication between the database level and the user interface levels that are either the edition tools or the external application tools. The interactive tools constitute the environment graphical user interface (GUI),
developed using JFC (JAVA Foundation Class). They are the required resources to aid the user in the traffic planning. By using the software features the user can rapidly and consistently perform the main steps in the

Fig. 19. Multimodal transportation network design.

Fig. 20. Procedure to update node's attributes.
traffic planning. The environment allows the user to browse the system entities, to access their attributes, to edit them—changing the system state—to prepare, execute and integrate new tools, as simulation process and analysis tools.

Fig. 21. Procedure to update link's labels.

Fig. 22. Public transport modeling. Detail of a Metro Line: graphical presentation of the mode itinerary, location of metro stations. Description of a timetable (departures schedule and Stop Times).
The modeling procedure builds the multimodal transportation network (Fig. 19) using TransObject components. Transit network includes roads, rails, and road connections (stops, light control, etc.). Each link must contain appropriate identification and associate labeling.

To simulate the real transit, and to make the multimodal routing very efficient to deal with optimal $O-D$ paths, transportation networks include several details such as: turn delays, overpasses, underpasses, link delays. 

![Diagram](image1)

**Fig. 23.** Finding optimal path from source to destination using multimodal routing algorithm: information of the number of paths, transfers and total time.

![Diagram](image2)

**Fig. 24.** (a) Design of an abstraction level and (b) associated network creation.
classifications, performance functions, one-way links, intersection and junction attributes, transfer points, delay functions, transit access, egress, and walk transfer links (Figs. 20 and 21).

The software uses the proposed multimodal routing algorithm to find the optimal path for each O–D trip, with the number of the modal transfers given by the traveler. It provides a very efficient user-interface to model trips directly on the screen (Fig. 23). Scheduling model must take into account mode itinerary, schedule departures, and mode stations (Fig. 22).

Network partitioning or network management is performed with a data model defined above. In this data model a node can represent the abstraction of one (or more) subnetwork (e.g. modeling a local public transportation network), in the same way the edge can also represent a subnetwork. This hierarchical structure is used to create service districts based on accessibility, perform drive-time analysis, or evaluate possible facility locations. When we perform the network partitioning (Fig. 24a), we may calculate the travel time from specific locations in the transit system by the new operator (Path View operator). The Undevelop and Develop operators (Fig. 24b) are used to browse the hierarchical levels of the network.

7. Conclusions

This paper has examined the adaptive multimodal routing model. Trip planning component of MTS uses this model to inform and assist travelers to choose the best paths in terms of modalities, transfers and schedules. Furthermore, we have pointed out the advantages of merging the graph concepts and the object oriented paradigm to describe data models. Nodes and links are efficient concepts to represent several elements of the multimodal transit networks. Abstraction paradigm facilitates the description of the network levels. The proposed algorithm adds more aspects of the integration of path evaluation operator into GIS-analysis. The user can define the number of modal transfers, the number of the best paths between the origin and destination places, the corresponding schedules, and the modes of transport preferred to reach his destination.

On the other hand, our software answers growing need of transit planning. The concepts of the object-oriented approach, systems integration, and visual interactive design make our software environment a powerful analysis tool for multimodal traffic systems studies. The software conception makes the implementation of new tools easier and increases its applicability as a growing number of tools are integrated into the environment.

Finally, we briefly mention that the futures investigations will take into account the real-time multimodal trips-requests. Information about modes, such as arrivals, departures and vehicle locations are issued by automated vehicle location (AVL) systems. Hence, the waiting time and the travelling time must be estimated based on real-time information some applications of the new information technologies in transportation are illustrated in [27] and some new contributions in transportation systems analysis are presented in [25].

References