Research Article

Distributed Approach for Solving Time-Dependent Problems in Multimodal Transport Networks

Carlos Galvez-Fernandez,[1] Djamel Khadraoui,[1] Hedi Ayed,[1, 2] Zineb Habbas,[2] and Enrique Alba[3]

[1] CITI, CRP Henri Tudor, 29 avenue John F.Kennedy, 1855 Luxembourg, Luxembourg

[2] LITA, University Paul Verlaine-Metz, Ile du Saulcy, 57045 Metz Cedex 1, France

[3] Departamento de Lenguajes y Ciencias de la Computación, Universidade de Málaga, Campus Teatinos, 29071 Málaga, Spain

Correspondence should be addressed to Carlos Galvez-Fernandez, carlos.galvez@tudor.lu

This paper presents an alternative approach for time-dependent multimodal transport problem. We describe a new graph structure to abstract multimodal networks, called *transfer graph*, which adapts to the distributed nature of real information sources of transportation networks. A decomposition of the Shortest Path Problem in *transfer graph* is proposed to optimize the computation time. This approach was computationally tested in several experimental multimodal networks having different size and complexity. The approach was integrated in the multimodal transport service of the European Carlink platform, where it has been validated in real scenarios. Comparision with other related works is provided.

1. Introduction

Multimodal transport is the combination of two or more means of transport to move passengers or goods from one source to a destination [1]. The traveller can use either public (e.g., bus, taxi, metro, railway, and ship) or private vehicles (e.g., car, motorbike, bicycle, and walking). Multimodal journeys are becoming a real necessity in our society in order to guarantee a high level of mobility both inside cities and at the regional level [2].

From the last decades, many research projects were devoted to develop multimodal transport systems that recommend travellers a combination of transport means for door-to-door journeys [1, 3–5]. These systems must cope with different transport information sources, which are normally distributed in real scenarios, that is, transport data are usually provided by different public and private companies and can be stored in different locations. Another

challenge for these systems is to integrate real-time traffic information (e.g., traffic jams, delays, roadworks) in order to adapt proposed routes to real conditions.

Different abstractions have been proposed to represent the time-dependent multimodal transport problem. Three main categories of approaches can be identified: the multigraph-based approach [2, 6, 7], the constraint satisfaction problem approach [8], and the grid-based approach [9]. In order to adapt multimodal transport abstraction to the distributed nature of real-world information, the following constrains should be fulfilled: (i) the underlying multimodal network is assumed to be flat and to contain no hierarchical structure, like regional hierarchy; (ii) the involved unimodal networks may be distributed, that is, unimodal networks are stored and accessed separately; (iii) if there are multiple network information sources within a single mode, they may be distributed. The above-mentioned approaches relax any of these constraints when modeling and solving the problem.

An alternative representation of the time-dependent multimodal transport problem was partially described in [10]. Here we present a formal description of this abstraction as well as the transfer graph approach, which calculates the best paths in this structure. This approach was tested and also validated in the real context of Carlink Platform (Carlink, Wireless Platform for Linking Cars, is a project of the Celtic Cluster Programme Call, which aims to develop an intelligent wireless traffic service platform between cars) [11]. Transfer graph approach provides a solution for the time-dependent multimodal transport problem which fulfills the three defined constraints and adapts to the distributed nature of real transport information providers.

The outline of this article is the following. Section 2 describes the transfer graph approach for the time-dependent multimodal transport problem. In Section 3 this approach is computationally tested. Section 4 describes Carlink, the real scenario where the approach is validated, mainly the service platform as well as the obtained results. In Section 5 the results are compared with other related works. Finally, this paper ends with some conclusions in Section 6.

2. Transfer Graph Approach for Multimodal Transport Problems

In real scenarios, transport information sources are normally distributed, that is, transport schedules are usually provided by different public and private companies. Besides, this information is frequently modified due to daily unexpected perturbations such as traffic jams, delays, and roadworks. These restrictions complicate the management of transport information and the updates of transport networks.

This section presents the transfer graph approach for solving the Shortest Path Problem (SPP) in discrete time-dependent multimodal network. It is able to compute multimodal routes even if all involved networks are kept structurally separated and distributed. Instead of building a solution limited to the unified view of multimodal transport system, our strategy consists in finding a fundamental network representation which covers multimodal networks and then solves the SPP on this abstraction.

The transfer graph approach adapts to the distributed nature of real world transport information sources and allows updates on each transport network independently without requiring any further modification. Our approach is mainly divided into two steps. In the first step, relevant paths are computed within each unimodal network. In the second step, the graph is reduced to a simple time-dependent unimodal network, where the classical SPP can be solved.

The next subsections first introduce the basic concepts of graphs and a definition of SPP in time-dependent multimodal networks. Then, the *transfer graph* abstraction is formally described, followed by the presentation of the *relevant graph*, a simplified structure calculated from *transfer graph*. Finally, the transfer graph approach to calculate SPP in *transfer graph* is discussed.

2.1. SPP in Discrete Time-Dependent Multimodal Networks

Let $G = (N, M, A)$ denote a directed graph or network, in which $N = \{n_1, \ldots, n_j\}$ is a set of nodes, $M = \{m_1, m_2, \ldots, m_k\}$ is a set of transport modes (e.g., train, bus, and car), and $A = \{a_1, \ldots, a_l\} \subseteq N \times N \times M$ is a set of arcs. An arc $a_i \in A$ can be identified by $(n_i, n_{i'}, m_i)$ or $(n_i, n_{i'})_i$, where $n_i, n_{i'} \in N \wedge m_i \in M$. The a_i means the possibility of going from node n_i to $n_{i'}$ by using transport mode m_i. A value c_{a_i} is associated to each arc a_i, indicating the cost of including the arc in the solution (e.g., distance, time).

Definition 2.1. A graph $G = (N, M, A)$ is said to be multimodal if there is at least two transport modes $m_i, m_j \in M$ where $(n_i, n_{i'}, m_i), (n_j, n_{j'}, m_j) \in A$, $m_i \neq m_j$, and $n_i, n_{i'}, n_j, n_{j'} \in N$. If there is only one transport mode in the graph, the graph is said to be unimodal.

In a graph $G = (N, M, A)$ a path or route p is a sequence of arcs between a pair of nodes n_1 and n_{k+1}, $((n_1, n_2, m_1), (n_2, n_3, m_2), \ldots, (n_k, n_{k+1}, m_k))$, $n_j \in N$ for all $j = 1, \ldots, k+1$, and $(n_i, n_{i+1}, m_i) \in A$, $m_i \in M$ for all $i = 1, \ldots, k$.

Definition 2.2. Given a graph $G = (N, M, A)$, a path $p = (a_1, a_2, \ldots, a_l)$ is said to be multimodal if $\exists a_i, a_j \in A$, $a_i = (n_i, n_{i'}, m_i)$, $a_j = (n_j, n_{j'}, m_j)$, $m_i \neq m_j$, $i \neq j$, and $i, j \leq l$. If there is only one mode involved in the path, then the path is said to be unimodal.

Let P be the set of paths in a graph G and $f : P \rightarrow R$ a function which represents the cost of a path. An SPP is considered to be multiobjective if f maps to a vector of dimension $k \geq 2$, $f : P \rightarrow R^k$. Each component $f_i \in f$, for all $(1 \leq i \leq k)$, follows the definition of single objective path cost function.

Let t denote time in a discrete set $\{t_1, t_2, \ldots, t_l\}$. Unlike nontime-dependent networks where arc costs are fixed, in time-dependent networks the cost of an arc $a_i = (n_i, n_{i'}, m_i)$ varies according to cost function $c_{a_i}(t_k)$ for all $t_k \in \{t_1, \ldots, t_l\}$.

Definition 2.3. Given a graph $G = (N, M, A)$ and a discrete time set $\{t_1, \ldots, t_l\}$, a travel for the arc $a_i = (n_i, n_i', m_i) \in A$ is defined as tuple $(t, t') \in \mathrm{Tr}$, $t, t' \in \{t_1, \ldots, t_l\}$, where t is the departure time from node n_i, and t' is the arrival time at node n_i'. Tr denotes the set of travels in G.

Definition 2.4. Given a graph $G = (N, M, A, \mathrm{Tr})$ and a discrete time set $\{t_1, \ldots, t_l\}$, G is said to be a time-dependent network if for all $a_i \in A$, $T_{a_i} = \{(t_1, t_1'), \ldots, (t_k, t_k')\} \subseteq \mathrm{Tr}$ where $|T_{a_i}| \geq 1$ and for all $j = 1, \ldots, k$, $t_j, t_j' \in \{t_1, \ldots, t_l\}$ and $t_j \leq t_j'$.

We conclude this subsection by formulating the SPP in time-dependent multimodal graphs. The reader can find a wider description of the SPP in [12].

Definition 2.5. Given a time-dependent multimodal graph $G = (N, M, A, \mathrm{Tr})$, two nodes $s, d \in N$, and time $t \in \{t_1, t_2, \ldots, t_l\}$, the SPP is defined as follows: calculate a path p from node s to d that departures from s at time t and $f(p)$ is minimal.
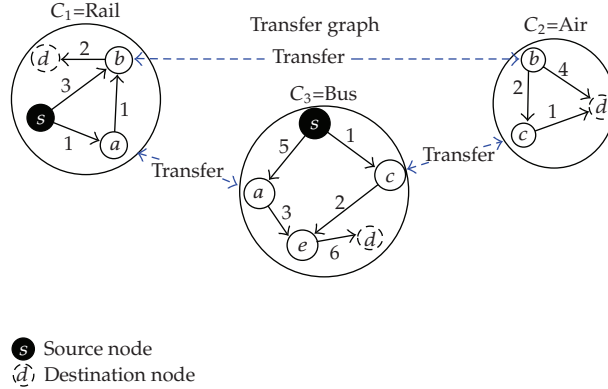
Figure 1: *Transfer graph* illustrative case.

2.2. Transfer Graph

In this subsection, we present the *transfer graph*, a graphical structure that abstracts the time-dependent multimodal transport network.

A *transfer graph* is described by a set of unimodal networks and a set of arcs connecting them, and it is defined by $T = (N, A, C, \text{Tr})$. Let $s, d \in N$ be an origin-destination pair and $C = \{C_1, C_2, \ldots, C_m\}$ a set of unimodal networks called components, where each $C_g = (N_g, A_g)$, $N_g \subseteq N$ and $A_g \subseteq A$. C itself is such that $N = \bigcup_{\forall g} N_g$ and $A = \bigcup_{\forall g} A_g$, for all $g = 1, \ldots, m$, $C_g = (N_g, A_g) \wedge C_g \in C$. Unlike partitioned graph, given two distinct components C_g and $C_{g'}$, having $N_g \bigcap N_{g'} = \emptyset$ is not mandatory; however $A_g \bigcap A_{g'} = \emptyset$ must always hold. Again, let us define two distinct components $C_g, C_{g'} \in C$; a node n is said to be a transfer point when $n \in N_g \bigcap N_{g'}$, that is, any node which belongs to more than one component. Finally, a transfer is any arc which connects two components of the *transfer graph*: (n_i, n_j), $n_i \in N_g \wedge n_j \in N_{g'} \wedge n_i = n_j \wedge g \neq g'$.

Figure 1 illustrates an example of this abstraction, where C_1, C_2, and C_3 are components connected by three transfers, and nodes a, b, and c are transfer points. One can observe that source node s and target node d may belong to more than one component; this is because they can be transfer points. Actually, time-dependence and multiobjectivity are not considered in this example, although the approach is able to work with these restrictions.

Transfer graph abstraction separates all transport modes in unimodal networks. So, when a perturbation (e.g., traffic jam, roadworks) is detected for a transport mode, it does not affect globally to the representation, but locally. In those cases, it is only necessary to update those components related to the incident. According to this characteristic we can state Lemma 2.6.

Lemma 2.6. *Transfer graph abstraction is more robust against modification of transport network data than classical multimodal network representation.*

2.3. Relevant Graph

In the previous subsection we have formally defined the *transfer graph*. We present now a simplified structure derived from this abstraction, called *relevant graph*, from which the Shortest Path Problem can be easier computed.

Table 1: Best intracomponent paths calculated for the *transfer graph* in Figure 1.

	Comp. C_1	Cost	Comp. C_2	Cost	Comp. C_3	Cost
$P_{s \cdot d}^{*g}$	$\{(s,a)_1, (a,b)_1, (b,d)_1\}$	4	—	—	$\{(s,c)_3, (c,e)_3, (e,d)_3\}$	9
$P_{s \cdot -}^{*g}$	$\{(s,a)_1, (a,b)_1\}$	2	—	—	$\{(s,a)_3\}$	5
	$\{(s,a)_1\}$	1	—	—	$\{(s,c)_3\}$	1
$P_{+ \cdot -}^{*g}$	$\{(a,b)_1\}$	1	$\{(b,c)_2\}$	2	—	—
$P_{+ \cdot d}^{*g}$	$\{(a,b)_1, (b,d)_1\}$	3	$\{(b,c)_2, (c,d)_2\}$	3	$\{(a,e)_3, (e,d)_3\}$	9
	$\{(b,d)_1\}$	2	$\{(c,d)_2\}$	1	$\{(c,e)_3, (e,d)_3\}$	8

At a high level, paths in a *transfer graph* can be divided into two groups: *intracomponents paths* and *intercomponent paths*. An *intercomponent path* within a *transfer graph T* is considered as any sequence of arcs which connects two nodes $n_i, n_j \in N$ where at least there are two arcs belonging to two distinct components. On the other hand, an *intracomponent path* within C_g is any sequence of arcs which connects two nodes $n_i, n_j \in N_g$ whose arcs belong only to one component. *Intra component paths* can be divided into the following categories:

(i) *full paths* ($P_{s \cdot d}^{*g}$): the smallest route which starts at source node s and ends at target node d within the component C_g;

(ii) *relevant head paths* ($P_{s \cdot -}^{*g}$): the smallest routes which start at source s and end at a transfer point within the component C_g;

(iii) *relevant intermediate paths* ($P_{+ \cdot -}^{*g}$): the smallest routes which start and end at any transfer point within the component C_g;

(iv) *relevant tail paths* ($P_{+ \cdot d}^{*g}$): the smallest routes which start at any transfer point and end at target d within the component C_g.

Given a *transfer graph* $T = (N, A, C, \mathrm{Tr})$ and an origin-destination pair $s, d \in N$, assume that for all components $C_g \in C$ we have computed the best relevant path sets: $P_{s \cdot d}^{*g}$, $P_{s \cdot -}^{*g}$, $P_{+ \cdot -}^{*g}$, and $P_{+ \cdot d}^{*g}$ (see Table 1). Having these relevant path sets in hand, it is possible to derive a reduced graph from which all possible best *inter component paths* from s to d can be computed. We call this graph the *relevant graph* and use R to denote it. The nodes of R are s, d and a subset of all transfer points in the *transfer graph*. The arc set of R is the best relevant intra component paths viewed as edges, that is, paths that are represented only by the initial and final nodes. In fact, R is a multigraph, but in general, its scale should be much smaller than the equivalent multigraph of the underlying *transfer graph*. This is because any node appearing in R is a transfer point, except the origin and destination nodes. So, the number of nodes in R depends directly on the number of transfer points in the whole *transfer graph*. Figure 2 shows the relevant graph computed out of Figure 1.

In order to formally describe R, let V^g denote the set of relevant nodes for component C_g, and let E^g be the set of relevant paths computed for C_g. From all V^g and E^g we can, respectively, build $V = (\bigcup_{\forall g} V^g) \bigcup \{s, d\}$ for all $C_g \in C$ the set of relevant nodes for all component, and $E = \bigcup_{\forall g} E^g$ for all $C_g \in C$ the set of relevant paths for all component $C_g \in C$. The *relevant graph* is described by the pair $R = (V, E)$. If we consider time-dependent networks with different discrete time $t_i = \{t_1, t_2, \ldots, t_l\}$, then in order to build the *relevant graph*, it is necessary to calculate the best relevant paths at any departure time t_i, that is, $P_{s \cdot d}^{*g}(t_i)$, $P_{s \cdot +}^{*g}(t_i)$, $P_{- \cdot d}^{*g}(t_i)$, $P_{- \cdot +}^{*g}(t_i)$, for all $C_g \in C$ and for all $t_i = \{t_1, t_2, \ldots, t_l\}$.

Relevant graph

— Edge from C_1
······ Edge from C_2
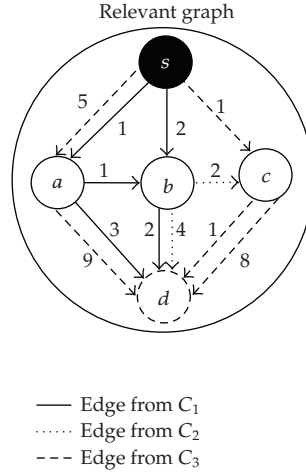- - - Edge from C_3

Figure 2: Relevant graph calculated for the transfer graph in Figure 1.

When a *relevant graph* is built from a *transfer graph* for nodes s and d and time t, we have a time-dependent unimodal multigraph with a reduced number of nodes where all the shortest paths between transfer points, s and d, are calculated. The computation of the shortest path from s to d in R is basically the classical origin-destination SPP on a reduced time-dependent unimodal network. This allows us to establish the following lemma.

Lemma 2.7. *Let $T = (N, A, C, Tr)$ be a transfer graph and $R = (V, E)$ its associated relevant graph, then R is a less complex structure than T in terms of the shortest path computational time.*

2.4. Transfer Graph Approach

Once the *transfer graph* and the *relevant graph* are formally defined, we present the decomposition of the SPP in these abstractions as well as the transfer graph approach.

Every time a request is fulfilled for obtaining the shortest path from a source s and a destination d at departure time t, we need to calculate the best relevant paths for $P_{s \cdot d}^{*g}(t)$, $P_{s \cdot -}^{*g}(t)$, and $P_{+ \cdot d}^{*g}(t_i)$ for all $C_g \in C$, for all $t_i = \{t_1, \ldots, t_l\}$. However, $P_{+ \cdot -}^{*g}(t_i)$ for all $t_i = \{t_1, \ldots, t_l\}$ is calculated once, since it does not depend on s and d. Taking advantage of this feature we can decompose the SPP in transfer graphs as follows.

(1) The best paths for $P_{+ \cdot -}^{*g}(t_i)$, for all $C_g \in C$, for all $t_i = \{t_1, \ldots, t_l\}$ are calculated and stored. This is called *precalculations*.

(2) When a request is fulfilled for two specific nodes s and d, and time t, then $P_{s \cdot d}^{*g}(t)$, $P_{s \cdot -}^{*g}(t)$, and $P_{+ \cdot d}^{*g}(t_i)$ for all $t_i = \{t_1, \ldots, t_l\}$ can be calculated.

(3) Finally, the *relevant graph* R is constructed and the shortest path is computed.

This decomposition reduces time complexity of the problem considerably because *precalculations* compute the most difficult set of paths in terms of computation time. *Precalculations* provide an important advantage if multiple and frequent requests are fulfilled (e.g., in a real web service). On the contrary, *precalculation* must be partly recomputed when a perturbation is detected. Table 2 shows an example of recalculating the *relevant graph* for

Table 2: Best intra component paths recalculated for graph in **Figure 1** if the cost of arc $(b,c)_2$ of C_2 increases from two to four. Recalculated paths are shown in bold.

	Comp. C_1	Cost	Comp. C_2	Cost	Comp. C_3	Cost
$P_{s \cdot d}^{*g}$	$\{(s,a)_1, (a,b)_1, (b,d)_1\}$	4	—	—	$\{(s,c)_3, (c,e)_3, (e,d)_3\}$	9
$P_{s \cdot -}^{*g}$	$\{(s,a)_1, (a,b)_1\}$	2	—	—	$\{(s,a)_3\}$	5
	$\{(s,a)_1\}$	1	—	—	$\{(s,c)_3\}$	1
$P_{+ \cdot -}^{*g}$	$\{(a,b)_1\}$	1	$\{(b,c)_2\}$	**4**	—	—
$P_{+ \cdot d}^{*g}$	$\{(a,b)_1, (b,d)_1\}$	3	$\{(b,d)_2\}$	**4**	$\{(a,e)_3, (e,d)_3\}$	9
	$\{(b,d)_1\}$	2	$\{(c,d)_2\}$	**1**	$\{(c,e)_3, (e,d)_3\}$	8

Input: Transfer graph T with m modes and n possible departure times
for $g = 1$ to m **do**
   for $i = 1$ to n **do**
      $P_{+-} \leftarrow Compute(P_{+-}^{*g}(t_i), T)$
      $StoreDataBase(P_{+-}, DB)$
   end for
end for
while $Request(s,d,t)$ **do**
   for $g = 1$ to m **do**
      $P_{sd}[g] \leftarrow Compute(P_{s \cdot d}^{*g}(t), T)$
      $P_{s-}[g] \leftarrow Compute(P_{s \cdot -}^{*g}(t), T)$
      for $i = 1$ to n **do**
         $P_{+d}[g] \leftarrow Compute(P_{+ \cdot d}^{*g}(t_i), T)$
      end for
   end for
   $R \leftarrow RelevantGraph(P_{sd}, P_{s-}, P_{+d}, DB)$
   $ShortestPath(R, s, d, t)$
end while

Algorithm 1: Transfer graph approach.

Figure 1 if the cost of the arc between nodes b and c for component C_2 increases from two to four. As it can be seen, only few paths are necessary to be recalculated, instead of considering all transport network. This is specially efficient for *transfer graphs* with multiple components.

Finally, we present the transfer graph approach in **Algorithm 1**. The procedure *Compute* calculates the shortest path for a specific component and departure time. This step can be calculated independently for each component, due to the characteristics of *transfer graph* and the decomposition of the SPP in this abstraction. The procedure *StoreDataBase* saves in the data base *DB* the previous calculated set of the shortest paths. The procedure *Compute* is called three times to calculate all the specific sets of shortest paths when the user request is submitted. *RelevantGraph* is the procedure reducing the initial graph to a more compact structure corresponding to a given user request. Finally, the procedure *ShortestPath* gives the smallest route requested by the user. Obviously all these procedures can be implemented in different ways and giving rise to different complexities. Here, we show the feasibility of our approach by using a variant of Dijsktra's algorithm. Therefore, the transfer graph approach can be applied to scenarios with distributed transport information sources.

Table 3: Performance and graph reduction for different instances of the SPP with variable transport modes, number of nodes and arcs.

SPP Instance (N, M)	Transfer graph					Relevant graph			CPU time	
	Nodes	Arcs	Transfers	Travels	Precalcula. time (second)	Nodes	Arcs	% Node reduction	Avg.	Max.
SPP(50, 3)	50	200	19	2000	0.72	15	210	70	0.001	0.001
SPP(100, 3)	100	300	33	3000	1.47	25	570	75	0.012	0.108
SPP(200, 3)	200	600	62	6000	2.13	52	2170	74	0.025	0.656
SPP(500, 3)	500	1500	152	15000	6.33	126	13214	75	0.054	0.342
SPP(1000, 3)	1000	3000	376	30000	26.11	304	64376	70	0.248	1.295
SPP(2000, 3)	2000	6000	850	60000	102.23	652	243982	67	1.053	3.454
SPP(50, 4)	50	200	26	2000	0.71	16	202	68	0.016	0.176
SPP(100, 4)	100	300	50	3000	1.43	30	711	70	0.013	0.250
SPP(200, 4)	200	600	103	6000	2.62	60	3050	70	0.028	0.193
SPP(500, 4)	500	1500	264	15000	6.87	152	19449	70	0.071	0.624
SPP(1000, 4)	1000	3000	493	30000	25.87	307	70617	69	0.178	0.975
SPP(2000, 4)	2000	6000	904	60000	100.54	657	247018	67	1.141	4.129
SPP(50, 5)	50	200	35	2000	0.69	17	216	66	0.010	0.160
SPP(100, 5)	100	300	52	3000	1.26	33	824	67	0.014	0.387
SPP(200, 5)	200	600	133	6000	2.35	66	3434	67	0.019	0.278
SPP(500, 5)	500	1500	314	15000	5.80	164	19418	67	0.049	0.677
SPP(1000, 5)	1000	3000	654	30000	22.12	320	73765	68	0.149	0.774
SPP(2000, 5)	2000	6000	926	60000	99.04	662	250643	67	1.005	3.705

3. Simulation Results

So far, we have introduced the *transfer graph* abstraction and the transfer graph approach. In this section we present a computational test of this approach with several instances of the SPP in time-dependent multimodal networks.

We have selected eighteen instances of the SPP with three, four, and five transport modes and increasing number of nodes from 50 to 2000, named $SPP(nodes, modes)$. A *transfer graph* is built for each instance with variable number of arcs and transfers, but keeping high number of transfers and 10 possible random travel times per each arc, in order to simulate real scenarios. Table 3 presents for each problem instance a description of the *transfer graph*, the characteristic of the *relevant graph* built from the *transfer graph*, and the computation time. Instances are grouped by number of nodes.

A variant of Dijkstra's algorithm [13] is used to implement the function $ShortestPath(R, s, d, t)$ of the transfer graph approach (Algorithm 1), which calculates the shortest path in terms of travel time. In these simulation tests we consider that $P_{+-}^{*g}(t_i)$ for all $C_g \in C$, for all $t_i = \{t_1, \dots, t_l\}$ are available, that is, the *precalculations* are already computed. The computation time of *precalculation*s is presented in Table 3.

The *transfer graph* approach was tested in a computer equipped with an Intel Core2 Duo T7300, 2 GHz, and 2 GB RAM. 100 tests have been done for each instance with random nodes and departure time, calculating the average and maximum CPU time for them all (see Table 3).
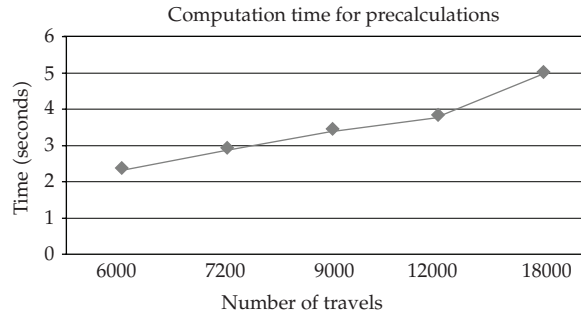
Computation time for precalculations

Figure 3: Computation time for *precalculation* in the instance SPP(200, 3) with different numbers of *travels*.

When building the *relevant graph* from the *transfer graph*, the number of nodes in all instances is reduced around 70%, although the number of arcs increases considerably (see Table 3). Since the complexity of Dijkstra's algorithm depends on the number of nodes $O(N^2)$, this is an appropriate algorithm to calculate the shortest path in a *relevant graph*.

Transfer graph approach has been tested in eighteen instances of the problem, providing a solution in a second approximately in average, even for instances with 2000 nodes and 60.000 *travels*. Precalculation time increases according to the network size and number of travels. Nevertheless, this *precalculations* must be done once, and considerably reduced the computation time for requests.

Finally, we present the computation cost of *precalculation*s on medium-sized networks with different levels of complexity, according to number of *travels*. Five transfer graphs are built for the instance $SPP(200, 3)$ with variable number of *travels*. Figure 3 shows that the computation cost of *precalculation* is not considerably affected by the number of travels.

4. Experimental Validation

In previous sections the *transfer graph* approach has been described. In this section, we first present the Carlink Platform (http://carlink.lcc.uma.es/) as well as its services and constraints. Then, it is described how the *transfer graph* was validated in Carlink Platform for real scenarios.

The Carlink Wireless Traffic Service Platform is designed to provide a basis for a wide range of commercial and governmental traffic, safety, and transport services (see Figure 4). The platform is a wireless ad hoc type communication entity with connectivity to the backbone network via base stations. The communication platform itself is the key element of Carlink, but the various services created to the platform have an important role. The three most important services are the traffic local road weather service (RWS), the traffic management service (TMS), and multimodal transport service (MTS).

The local RWS is derived from the FMI (Finnish Meteorological Institute) road weather model [14] and provides automatic traffic safety enhancement operations by gathering safety related information from vehicles and delivering consolidated data back, automatically without requiring user initiation. The TMS provides the traffic status by using static traffic information offered by the authorities and the dynamic traffic information given by the Carlink floating car application. This application is composed of several cars collecting information about the vehicle density in different locations of the road network
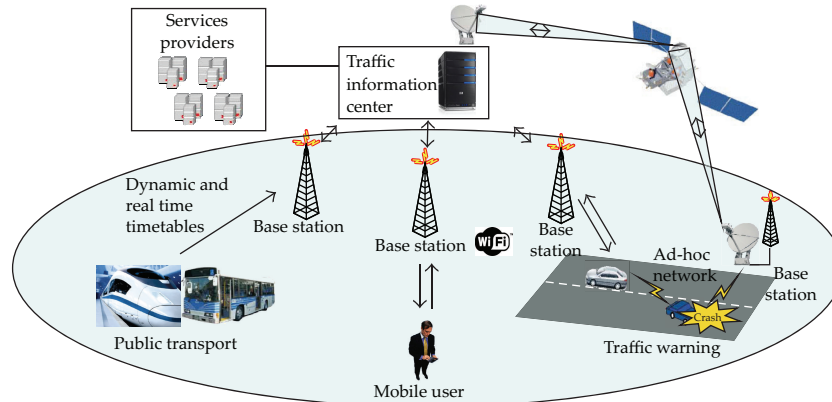
Figure 4: Carlink platform structure.

and providing information of the traffic status. Finally, the MTS calculates multimodal itineraries according to a travel requested and a set of constraints specified by the user. This service is based in the transfer graph approach which has been extensively described in Section 2.

The platform exploits information of the RWS and TMS, as well as of other services, in order to generate accident or incident warning from traffic and weather conditions (see Figure 4). The MTS uses this information to calculate best routes avoiding bad condition roads and accidental areas to reduce the traveling time, or even recommending to change the transport mode if a blocked road is detected (e.g., to park the car near a train station and taking the train instead).

4.1. Experimental Environment

The Wireless Traffic Service Platform is divided in three different parts: Traffic Service Central Unit (TSCU), Traffic Service Base Station (TSBS), and Mobile End User (MEU) with ad hoc connectivity and (noncontinuous) backbone network connectivity. The whole platform can be seen as data exchange architecture between TSCU and MEU, while TSBS acts as bidirectional data transceivers. MEUs communicate between them in an ad hoc manner when encountering and can also send critical data directly to the TSCU by using a lower-capacity communication method (GRPS). The service cores are external systems connected to TSCU.

TSCU achieves and processes up-to-date traffic platform information from the service cores. The TSCU updates regularly TSBS with the more recent traffic platform information related to their area, through the fixed network. Each TSBS disseminates the traffic platform information to every MEU that passes close to it by using a wireless ad hoc connection. Finally, MEUs disseminate this information when encountering (see Figure 5).

MEU and TSBS are equipped with acquisition system that collect their own information (speed, GPS-data, airbag blast, road temperature, etc.) and transmit regularly to TSCU. This data is processed by the TSCU, and if a critical event is detected in an area (e.g., accident, bad road condition), a warning message is delivered in real-time directly to MEUs known to be in the warning area by using GPRS or similar communication method.
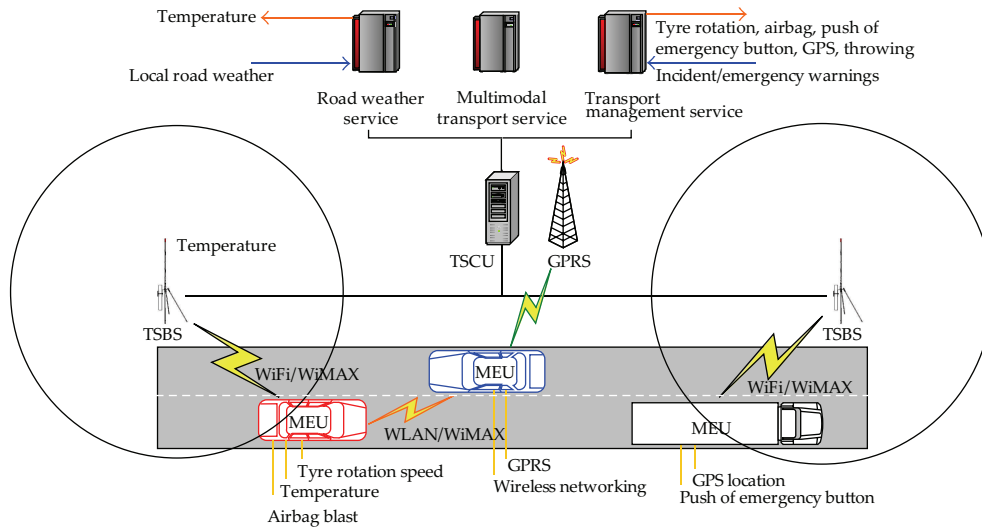
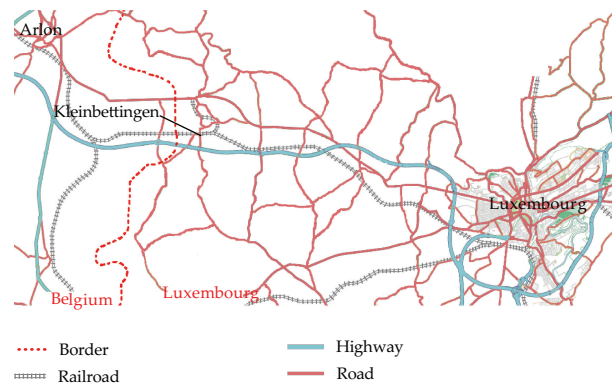**Figure 5:** Carlink platform architecture.

Figure 6: Validation scenario.

The advantage of this architecture is that both users (MEU) and disseminating systems (TSBS) send regularly information of the road condition to the TSCU. This is specially important when an accident occurs, because the TSCU is informed in real-time and can warn drivers in the surrounding of the accident area. Besides, since cars communicate each other when encountering, the number of TSBS necessary in roads can be considerably reduced, and also the system cost.

4.2. Experimental Results in Real Scenario

The validation of the developed system was performed in a real scenario (see Figure 6). The route origin is the Belgian city of Arlon, and the destination is Luxembourg city (Luxembourg). Several thousands of commuters do daily this trip by car, train, bus, or by a combination of these transport modes.
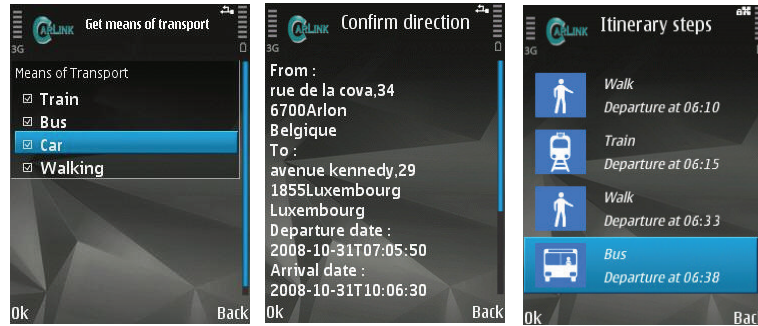
Figure 7: Screenshots of the application for mobile devices. Selecting the preferences for the traveling, confirming and showing the itinerary steps.

Table 4: Routes calculated for the validation scenario.

Route	Transport Mode	Time (min.)	Arlon, 34 cova st.	Highway E25/A4	Klein-bettingen parking station	Klein-bettingen train station	Luxem-bourg train station	Luxem-bourg bus station	Luxem-bourg 29 JFK ave.
				Itinerary steps (departure and arrival time)					
1	Car	40′	5:50						6:30
1 (updated)	Car	80′		6:01					7:21
2	Car	9′		6:01	6:10				
	Walk	5′			6:10	6:15			
	Train	18′				6:15	6:33		
	Walk	5′					6:33	6:38	
	Bus	18′						6:38	6:56

A *transfer graph* abstracting this scenario was deployed in the Carlink Platform and made available for the MTS. The *transfer graph* has 73 nodes, 49 transfer points, and 226 arcs and contains three components, associated to train, car, and bus. We consider that changing between two transport modes is done by walking, and this cost is represented in transfers.

The transfer graph approach is implemented in the MTS of the Carlink Platform. The user requests are sent to the MTS through a client application for mobile devices. In this application, the user can select the travel preferences (e.g., transport modes, departure time, origin, and destination), send the request to the MTS, and get the optimal path from the Carlink Platform (Figure 7).

In our scenario, the selected preferences are optimizing travel time to go from Arlon to Luxembourg by using any kind of transport mode and departing at 5:50. The first route recommended by the MTS consists in going directly by car using the highway (Table 4). While we are traveling by car, the TMS is reported at 6:00 with a perturbation in the current itinerary. Since this perturbation modifies the transport data for road transports, *precalculations* are recomputed for components bus and car, but not for rail. This perturbation increases the travel time for route 1 up to 80 minutes. Automatically, the MTS recalculates a second route taking into account this perturbation and the current GPS position of the traveller. This information is reflected to the user by the client application. The second route

Table 5: Comparative performances of approaches on different instances of the problem.

Number Nodes	Number Arcs	Average CPU time (second)			Normalized average CPU time (second)		
		Appr. 1 [2]	Appr. 2 [5]	Transfer. G	Appr. 1 [2]	Appr. 2 [5]	Transfer. G
50	200	0.33	0.54	0.001	0.33	0.71	**0.01**
100	300	0.53	0.81	0.012	0.53	1.07	**0.16**
200	600	1.27	—	0.025	1.27	—	**0.33**
500	1500	4.31	6.19	0.054	4.31	8.17	**0.72**
1000	3000	6.55	13.59	0.248	6.55	17.94	**3.30**

is expected to arrive 25 minutes before the first route and its itinerary steps are the following: park near the next train station in Kleinbettingen, take the next train to Luxembourg, and finally the bus to the final destination, as indicated in Table 4.

5. Related Work

The multimodal transport problem under the time-dependence constraint is not often studied in literature. As far as we know there are two works related to time-dependent multimodal transport problem which present the performance of their approach on different instances of the problem [2, 5].

Authors in [5] abstract the problem with nonhierarchical multimodal graph and propose several data structures and tables, previously precalculated, in order to optimize the computation. The problem is simplified as well as the space search by considering that the optimum paths between two nodes at any departure time are almost always topologically the same. Nevertheless, this constraint cannot be considered in realistic multimodal networks, because perturbations modify the conditions of the networks at different times of the day. Bielli et al. [2] consider the problem of multimodal transport between towns. The problem is geographically divided and abstracted with hierarchical graph: towns are represented by subgraphs, which are interconnected by direct links. The shortest paths inside subgraphs are previously precomputed, reducing the search space. The main goal of this work is the proposed abstraction of the problem, which can be partially parallelized. The proposed approach is tested in networks composed on many small-sized subgraphs.

Table 5 presents a comparison between these works and transfer graph approach on the same instances of the problem. Since different machines are used in the experiments (Pentium II, UltraSPARC III, and T7300), computation time has been normalized by two benchmarks [15, 16]. According to these benchmarks, T7300 and UltraSPARC III are 13.3 and 1.32 times faster than Pentium II, respectively. These ratios are applied to computation time and included in Table 5. After these corrections, our results are still faster and advantageous, even if they were performed in an old Pentium II. The overall results show that transfer graph approach performs better with respect to the normalized computation time. The possible reason of this performance is due to *precalculations*, and the decomposition of the SPP in *transfer graph* simplified the problem.

6. Conclusion and Future Work

The main contribution of this research is an approach for the multimodal transport problem which deals with distributed nature of real transport information providers.

The *transfer graph* can abstract scenarios with distributed transport information sources, since it separates and keeps all transport modes in different unimodal networks. Besides, each unimodal network can be easily updated when unexpected perturbations are detected. This characteristic is very important because the RWS and TMS provide in real-time information about weather and traffic condition, and if an incident or delay is detected, only those unimodal networks related to the warning area must be recalculated, instead of the total network.

The decomposition of the SPP in *transfer graphs* provides an adequate solution for systems which receive multiple and frequent requests. Nevertheless, *precalculations* limited the scalability of the *transfer graph* abstraction in terms of space memory.

The *transfer graph* approach has been tested with several instances of the SPP, calculating the solution in less than one second in average. This approach provides better performance in terms of computation time than other related works.

The transfer graph approach has been developed as a multimodal transport service for the Carlink Platform and validated in real conditions using several transport means. This service will be deployed and made active and will provide an important solution for thousand of daily travellers in Luxembourg.

Our future work is to take advantage of the inherent parallel and distributed structure exhibited by *transfer graph*. For this purpose, relevant paths will be computed by using multiobjective Evolutionary Algorithm within each component separately and concurrently and will be also experimented and tested in large real multimodal network spanning many countries and involving several transportation service providers.

Acknowledgments

References

[1] K. G. Zografos and K. N. Androutsopoulos, "Algorithms for itinerary planning in multimodal transportation networks," *IEEE Transactions on Intelligent Transportation System*, vol. 9, pp. 175–184, 2008.

[2] M. Bielli, A. Boulmakoul, and H. Mouncif, "Object modeling and path computation for multimodal travel systems," *European Journal of Operational Research*, vol. 175, pp. 1705–1730, 2006.

[3] T.-S. Chang, "Best routes selection in international intermodal networks," *Computers and Operations Research Archive*, vol. 35, pp. 2877–2891, 2008.

[4] Q. Li and C. E. Kurt, "GIS-based itinerary planning system for multimodal and fixed-route transit network," in *Proceedings of the MID-Continent Transportation Symposium*, pp. 47–50, Iowa State University, Ames, Iowa, USA, May 2000.

[5] A. Ziliaskopoulos and W. Wardell, "An intermodal optimum path algorithm for multimodal networks with dynamic arc travel times and switching delays," *European Journal of Operational Research*, vol. 125, no. 3, pp. 486–502, 2000.

[6] H. M. Foo, H. W. Leong, Y. Lao, and H. C. Lau, "A multi-criteria, multi-modal passenger route advisory system," in *Proceedings of the IES-CTR*, Singapore, July 1999.

[7] A. Lozano and G. Storchi, "Shortest viable hyperpath in multimodal networks," *Transportation Research Part B*, vol. 36, no. 10, pp. 853–874, 2002.

[8] D. Chiu, O. Lee, H.-F. Leung, E. Au, and M. Wong, "A multi-modal agent based mobile route advisory system for public transport network," in *Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS '05)*, p. 92, Big Island, Hawaii, USA, January 2005.

[9] M. Fragouli and A. Delis, "Easytransport: an effective navigation and transportation guide for wide geographic areas," in *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '02)*, pp. 107–113, Washington, DC, USA, November 2002.

[10] H. Ayed, D. Khadraoui, Z. Habbas, P. Bouvry, and J. F. Merche, "Transfer graph approach for multimodal transport problems," in *Proceedings of the 2nd International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences (MCO '08)*, pp. 538–547, Springer, September 2008.

[11] "CARLINK: Wireless Platform for Linking Cars," 2006–2008, http://carlink.lcc.uma.es/.

[12] S. Pallottino and M. Scutella, "Shortest path algorithms in transportation models: classical and innovative aspects," in *Proceedings of the Equilibrium and Advanced Transportation Modelling Colloquium*, Kluwer Academic Publishers, 1997.

[13] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[14] M. Kangas, M. Hippi, J. Ruotsalainen, et al., "The FMI road weather model," *HIRLAM Newsletter*, vol. 51, pp. 117–123, October 2006.

[15] "Standard performance evaluation corporation (SPEC)," http://www.spec.org/.

[16] "Geekbench benchmark," http://www.primatelabs.ca/geekbench/.