

Algorithms for Itinerary Planning in Multimodal Transportation Networks

Konstantinos G. Zografos, Konstantinos N. Androutsopoulos

Abstract— The itinerary planning problem in an urban public transport system constitutes a common routing and scheduling decision faced by travelers. The objective of this paper is to present a new formulation and an algorithm for solving the itinerary planning problem, i.e., determination of the itinerary that lexicographically optimizes a set of criteria (i.e. total travel time, number of transfers, and total walking and waiting time) while departing from the origin and arriving at the destination within specified time windows. Based on the proposed formulation, the itinerary planning problem is expressed as a shortest path problem in a time schedule multimodal network with time windows and time dependent travel times. A dynamic programming based algorithm has been developed for the solution of the emerging problem. The special case of the problem involving a mandatory visit at an intermediate stop within a given time window is formulated as two nested itinerary planning problems which are solved by the aforementioned algorithm. The proposed algorithm has been integrated in a web based journey planning system while its performance has been assessed by solving real life itinerary planning problems defined on the Athens Urban Public Transport Network providing fast and accurate solutions.

Index Terms— itinerary planning, multimodal transportation network, multi-criteria, shortest path

I. INTRODUCTION

A journey in an urban public transport system usually involves the combined use of the available public transport services. Each conventional urban public transport service is defined by a sequence of stops on a given route and a specified schedule. In this context, any journey may be realized by a path that consists of alternate interconnected route segments of the underlying public transport services.

Manuscript received May 14, 2007. The research reported in this paper has been partially supported by the General Secretariat for Science and Technology of the Greek Ministry of Development, through the project “Integrated Decision Support System for On-line, Real Time Trip Planning in Multimodal Transportation Systems” (ENOSIS).

K. G. Zografos is with the Transportation Systems and Logistics Laboratory, Department Management Science and Technology, Athens University of Economics and Business, Evelpidon 47A & 33 Lefkados, 113 62 Athens, Greece, Tel: +30 210 8203673-5, Fax: +30 210 8203684, Email: translog@aueb.gr.

K. N. Androutsopoulos is a research associate in the Transportation Systems and Logistics Laboratory, Department Management Science and Technology, Athens University of Economics and Business.

Any such path enhanced with a feasible schedule of traversing it, is called itinerary. It is evident that a journey may be realized by several alternative itineraries. A major decision that arises for the traveler relates to the selection of the itinerary that complies with his/her preferences and requirements. The major complexities of this type of decision relate to: i) the lack of information on the schedule and the routes of the public transport services, ii) the difficulty in determining feasible itineraries within the dense urban public transport network, and iii) the intensive task of assessing alternative feasible itineraries in terms of multiple traveling criteria. The provision of real time journey planning services through an online advanced public transport information system may alleviate the aforementioned difficulties [1], [2] and contribute to the enhancement of the usability and accessibility of the public transport services. During the past decade many advanced public transport information systems have been developed providing journey planning services with the objective of determining the shortest itineraries in terms of the travel time or the cost [3], [4]. Nowadays this type of services are directly accessible by the travelers through on-line web-based applications [2], [5], [6], [7]. The existing journey planning services aim to determine the itinerary between an origin-destination pair that optimizes one of the travel time, number of transfers, or walking time, given the preferable departure or arrival time stated by the user and considering constant (average) travel times.

In general the itinerary planning problem constitutes a multi-criteria routing and scheduling problem, providing a traveler with many alternative itineraries for a specific urban journey. However, this implies that the user should take on the tedious task (especially for the user who is unfamiliar with the specific public transport system) of selecting an itinerary from a list of alternative solutions. This issue could be resolved by the incorporation of the lexicographical ordering in the solution process of the itinerary planning problem. Under this approach, more than one criteria are taken into account, while only a single solution is provided to the user.

This paper provides a new formulation of the itinerary planning problem which apart from the lexicographically optimal itinerary aims to determine the optimal departure time of the journey. The proposed formulation enhances the existing trip planning models mentioned above by

simultaneously taking into account time-dependent travel times and multiple traveling criteria. The research in this paper, also addresses a variant of the above problem which includes the additional constraint of visiting an intermediate stop (e.g. a sightseeing) within a given time window. A dynamic programming algorithm is presented that solves the itinerary planning problem while it constitutes the basic procedure for solving the above variant. The computational performance of both algorithms have been tested on a wide range of real life journey planning problems defined on the urban public transport network of Athens, Greece. The emerging computational results indicate that the test problems were solved within a reasonable amount of time, hence both algorithms were integrated in a web-based decision support system for the real-time planning of urban and interurban journeys.

The remainder of the paper consists of five sections. The second section presents the formulation of the itinerary planning problem while the third section discusses the related work on addressing this category of problems. The fourth section presents the proposed algorithms for solving the itinerary planning problems while the fifth section is devoted to the computational performance of the proposed algorithms. Finally the sixth section includes the major concluding remarks.

II. PROBLEM DEFINITION

Assume that N is the set of nodes (vertices) that denote the stops of the urban public transport network. Any service is defined by a sequence of nodes $\{v_1, v_2, \dots, v_k\}$ implicitly specifying the route $R_s = \{(v_1, v_2), (v_2, v_3), \dots, (v_{k-1}, v_k)\}$ while the departure from each node v_i is allowed at specified points in time $ST_{v_i}^s := \{\tau_{v_i}^1, \tau_{v_i}^2, \dots, \tau_{v_i}^r\}$ within a time horizon $[0, T]$. In general, the travel time on any arc $(v_i, v_{i+1}) \in R_s$ is stochastic and dynamic, i.e., the travel time is a random variable and its distribution function depends on the departure time from the upstream node [8]. However, in the present paper this feature is simplified by assuming that the travel time is time dependent, i.e., it depends on the departure time τ from the upstream node v_i and it is denoted by $t_s^\tau(v_i, v_{i+1})$. The set of arcs formed by the routes $R_s, s \in S$ is denoted with A . In addition assume A' as the set of walking arcs $(v_i, v_j), v_i \in s_1, v_j \in s_2, s_1 \neq s_2$ which denote the transfer between any two services s_1, s_2 . The departure for traversing any interchange arc $(v_i, v_j) \in A'$ may occur at any point in time while the corresponding walking time, denoted by $t_w^\tau(v_i, v_j)$, is also assumed dependent on the departure time from the upstream node v_i . Thus, an urban public transport network can be modeled by a multimodal time-schedule

network, denoted by $G(N, A, A', S, ST)$ where $ST := \bigcup_{s \in S} \bigcup_{v_i \in N} ST_{v_i}^s$.

It should be clarified that the term multimodal is used in the sense of multiple fixed scheduled transport services. Note also that the origin and destination of a journey may be two locations with no public transport stops in their vicinity. This type of locations (e.g. points of interest, hotels etc.) are included in the above network as nodes and they are connected with their nearest public transport stops with walking arcs.

Any path $p^\tau(v_0, v_n)$ between two nodes v_0 and v_n enhanced with the schedule of traversing it, i.e., the departure time τ_{v_i} from each node included in the path (apart from the destination), is called itinerary. Note that the superscript τ in the notation of an itinerary denotes the ready time of the path, i.e., the earliest possible start time for traversing the path. Any itinerary may be written as a sequence of arcs enhanced with the associated departure times and the services used for traversing them, i.e.

$$p^\tau(v_0, v_n) := \{[(v_0, v_1); \tau_{v_0}; s_0], [(v_1, v_2); \tau_{v_1}; s_1], \dots, [(v_{n-1}, v_n); \tau_{v_{n-1}}; s_{n-1}]\} \quad (1)$$

The itinerary planning problem relates to the determination of the itinerary that satisfies the scheduling constraints (2)-(4) while optimizing a set of criteria including the total travel time (c_1), the number of transfers (c_2), and the total time transfer time, i.e., walking and waiting time (c_3).

$$\tau_{v_i} + t_s^{\tau_{v_i}}(v_i, v_{i+1}) \leq \tau_{v_{i+1}}, (v_i, v_{i+1}) \in p^l(v_0, v_n), \tau_{v_i} \in ST_{v_i}^s \quad (2)$$

$$d_{v_0}^e \leq \tau_{v_0} \leq d_{v_0}^l \quad (3)$$

$$a_{v_n}^e \leq \tau_{v_{n-1}} + t_s^{\tau_{v_{n-1}}}(v_{n-1}, v_n) \leq a_{v_n}^l \quad (4)$$

Constraint (2) implies that the departure from node v_{i+1} of the itinerary, should occur after the arrival at it from the preceding node v_i . Constraint (3) implies that the departure time from the origin should occur within the time window $[d_{v_0}^e, d_{v_0}^l]$ where $d_{v_0}^e, d_{v_0}^l$ denote the earliest and latest departure times, respectively. On the other hand, constraint (4) expresses that the arrival at the destination node should occur within the time window $[a_{v_n}^e, a_{v_n}^l]$, where $a_{v_n}^e, a_{v_n}^l$ denote the earliest and latest arrival times, respectively.

It should be noted that total time of transfers (waiting and walking time) constitutes part of the travel time of the itinerary and therefore these two criteria cannot be considered totally conflicting. However, the criterion referring to the number of transfers is usually conflicting with the travel time criterion, thus justifying the multi-criteria feature of the

problem. In this context, the itinerary planning problem constitutes a multi-criteria shortest path problem with time windows in a multimodal time schedule network with time dependent travel times. Due to the incorporation of multiple criteria in the emerging decision making problem, non-dominated rather than optimum solutions may be determined. A non-dominated solution of the itinerary planning problem between nodes v_0 and v_n at time τ , is defined as any feasible itinerary $p^\tau(v_0, v_n)$, i.e., satisfying (2)-(4), such that there does not exist any other $p^\tau(v_0, v_n)' \neq p^\tau(v_0, v_n)$ for which $c_k(p^\tau(v_0, v_n)) \geq c_k(p^\tau(v_0, v_n)')$ for any criterion k among those under consideration and $c_{k_0}(p^\tau(v_0, v_n)) > c_{k_0}(p^\tau(v_0, v_n)')$ for at least one criterion k_0 . Determining the entire set of non-dominated solutions is a tedious task since the complexity of the problem grows exponentially with the size of the network [9]. Alternatively, the incorporation of the lexicographical ordering in the solution process of the itinerary planning problem results to the prompt determination of the lexicographically minimum itinerary, taking into account the ranking of the evaluation criteria. Under this approach, more than one criteria are taken into account for the journey planning, while only a single solution is provided to the user. Given the ranking of the evaluation criteria $\{c_{k_1}, c_{k_2}, c_{k_3}\}$ and any two itineraries $p_1^\tau(v_0, v_n), p_2^\tau(v_0, v_n)$ (or p_1^τ, p_2^τ if no confusion is created about the origin and destination), the lexicographical ordering relationship (\leq_L) is defined as follows: $p_1^\tau \leq_L p_2^\tau$ iff there exists j such that $c_{k_j}(p_1^\tau) < c_{k_j}(p_2^\tau)$ and $c_{k_i}(p_1^\tau) = c_{k_i}(p_2^\tau)$ for $i=1, \dots, j-1$. In this case, we say that itinerary p_1^τ lexicographically dominates p_2^τ . In this context, an itinerary p_0^τ is lexicographically minimal if there does not exist an itinerary p^τ that lexicographically dominates p_0^τ . Note that based on the proposition 1 that follows, any lexicographically minimal itinerary is non-dominated.

Proposition 1. If p_1^τ is a lexicographically minimal itinerary from node v_0 to node v_n of the multimodal time schedule network $G(N, A, A', S, ST)$, then it is non-dominated for time τ .

Proof: Assume that p_1^τ is a lexicographically minimal itinerary, i.e., there does not exist itinerary p^τ that lexicographically dominates it. Let's assume now that $\exists p_2^\tau$ which dominates p_1^τ , i.e., $c_{i_k}(p_2^\tau) \leq c_{i_k}(p_1^\tau) \quad \forall k \in \{1, 2, 3\}$ and $\exists k_0 \in \{1, 2, 3\} : c_{i_{k_0}}(p_2^\tau) < c_{i_{k_0}}(p_1^\tau)$. With no loss of generality assume that i_{k_0} is the index of the first criterion for which

$c_{i_{k_0}}(p_2^\tau) < c_{i_{k_0}}(p_1^\tau)$, implying also that $c_{i_k}(p_2^\tau) = c_{i_k}(p_1^\tau)$ for $k < k_0$. However this last statement contradicts with the hypothesis that p_1^τ is a lexicographically minimal.

An important extension of the itinerary planning problem pertains to the incorporation of an additional mandatory visit to an intermediate node. This type of constraint emerges from the intention of the traveller to temporarily interrupt his/her journey in order to perform a prespecified activity at a given location and then continue his/her way to the destination. The incorporation of the intermediate visit constraint in the itinerary planning problem gives rise to the composite itinerary planning problem which aims to determine the lexicographically minimal itinerary from v_0 to v_n passing through v_q (denoted by $p^\tau(v_0, v_q, v_n)$) satisfying constraints (2)-(4) described earlier and the following additional constraints:

$$a_{v_q}^e \leq \tau_{v_{q-1}} + t_s^{\tau_{v_{q-1}}}(v_{q-1}, v_q) \leq a_{v_q}^l \quad (5)$$

$$\tau_{v_{q-1}} + t_s^{\tau_{v_{q-1}}}(v_{q-1}, v_q) + t_{v_q} \leq \tau_{v_q} \quad (6)$$

$$d_{v_q}^e \leq \tau_{v_q} \leq d_{v_q}^l \quad (7)$$

where $a_{v_q}^e$ and $a_{v_q}^l$ denote the earliest and latest arrival times at the intermediate node v_q respectively, t_{v_q} denotes the duration of the visit at stop v_q , while τ_{v_q} and $\tau_{v_{q-1}}$ denote the departure times from nodes v_q and v_{q-1} respectively.

Constraint (5) implies that the arrival time at the intermediate node v_q should lie within the time window $[a_{v_q}^e, a_{v_q}^l]$. Constraint (6) implies that the mandatory visit at stop v_q should last for at least t_{v_q} time units while constraint (7) expresses that the departure from the intermediate stop should occur within the time window $[d_{v_q}^e, d_{v_q}^l]$.

The remainder of this paper presents an overview of the previous related work on this area and provides the proposed algorithms for solving the itinerary planning problem and its variant mentioned above.

III. PREVIOUS RELATED WORK

The previous related work on this area of research is mainly focused on the path finding problem in a transit network. The major research effort on this problem relates to the determination of optimum itineraries based on a specific departure or arrival time. Two major types of formulations have been proposed up to date: i) the headway based, and ii) the schedule based models. The former approach assumes a constant headway for the transit lines while the interarrival

time at each node is approximated by the average headway of the transit lines visiting that node. In this context, the relevant path finding algorithms search for optimal paths in the underlying static network [10].

On the other hand, the schedule based approach assumes a fixed timetable for the transit lines with deterministic and specified departure times from every stop. Label correcting and label setting algorithms have been proposed for solving this category of problems [2], [11]-[14], while for the formulations with non-additive objective function, branch and bound schemes have been developed [15], [16]. Moreover, heuristic routines have been developed and integrated into transit information systems [17], [18], [19] where the fast solution of the problem is essential.

The formulation of the itinerary planning problem presented in this paper falls under the schedule based approaches while it enhances the existing ones by addressing simultaneously the following aspects: i) incorporation of multiple criteria through lexicographical ordering, ii) allowing the departure and arrival times to lie within specified time windows, iii) assuming time dependent travel times, and iv) taking into account the special case where an intermediate stop at a given time window should be scheduled. According to the knowledge of the authors no research on this problem has been focused on incorporating simultaneously the above features.

The proposed formulation of the itinerary planning problem gives rise to a shortest path problem in a multimodal time-schedule network with time dependent travel times. Ziliaskopoulos and Wardell [20] developed a label correcting algorithm for determining least time paths from all nodes of a multimodal network to a single destination node for every transport mode and departure times assuming time-dependent arc travel times and switching delays between two modes. The complexity of their algorithm is $O(T^2 n^5 |M|^3)$ where $|M|$ denotes the number of modes. Moreover, the most relevant work to the composite itinerary planning problem, has been provided by Bé-rubé, Potvin, and Vaucher [21]. In their work they propose a decomposition scheme for solving the travel planning problem, i.e., optimum path problem in a transportation network with deterministic time-dependent travel times under the constraint of visiting a sequence of specified nodes. The objective of their formulation is to determine the path in the corresponding time-expanded network that minimizes a generalized cost function of the travel time and the waiting time, while passing through a specified set of nodes in a predetermined order. Their algorithm utilize a heap implemented Dijkstra routine for determining the shortest path from intermediate node v_i to the next in sequence intermediate node v_{i+1} for all possible departure times $\tau \in [0, T]$. The shortest paths identified in stage i of the algorithm are taken into account in initializing

the shortest path routine for stage $i+1$, i.e., the shortest path of stage $i+1$ from intermediate node v_{i+1} to intermediate node v_{i+2} at time τ constitutes the expansion of the shortest path of stage i that arrives at node v_i from $i-1$ at time $\tau - d_i$ where d_i is the duration of the visit at node i . The computational complexity of their approach is $O(Km^* \log(n^*))$ where K is the number of intermediate visit nodes and m^* , n^* denote the number of arcs and nodes of the time expanded network.

Special case of the proposed itinerary planning problem relates to the time-dependent shortest path problem on a unimodal network for which many alternative solution methods have been developed. Cooke and Halsey [22] provided a dynamic programming algorithm for solving the fastest path problem between any node of a network with time-dependent travel times and a single destination for any possible departure time in the discrete time horizon $[0, T]$. Each iteration k of their algorithm, compares the temporarily optimal path from each node i and time τ in $[0, T]$ that comprises $k-1$ or less arcs with the emerging path consisting of k arcs. Based on the approach proposed by Cooke and Halsey, Ziliaskopoulos and Mahmassani [23] developed a label correcting algorithm for the fastest and minimum cost path problem.

An alternative category of algorithms for the time dependent shortest path problem has been based on determining efficient techniques in searching for optimum paths in the time-expanded network. Along this line, Cai, Kloks, and Wong [24] proposed an algorithm for the time dependent minimum cost path problem from a single origin to a single destination under the constraint that the total time of the path does not exceed a specified standard value T , i.e., the arrival time at the destination should occur within the time window $[0, T]$. For any time τ (from $\tau=1$ up to T), their algorithm scans every node v_i of the network for which an ingoing arc arrives at time τ , by selecting the minimum value between the cost of the currently optimal path from the origin to node i that arrives at time $\tau-1$ or earlier and the cost of the new paths that traverse the ingoing arcs arriving at v_i at time τ . The backward version of this approach has been proposed by Chabini [25] for the time dependent shortest path problem from all nodes to a single destination for all departure times. This algorithm moves backward in time (from $\tau=T$ up to 0) scanning every node v_i by assessing all new paths from v_i to the destination that depart at time τ and traverse any outgoing arc from v_i .

The multiple criteria time-dependent shortest path problem was initially addressed by Kostreva and Wiecek [26]. The algorithm they proposed is based on modifying the iterative process proposed by Cooke and Halsey for the single criterion case [22]. Hamacher, Ruzika, and Tjandra [27] proposed a

backward label setting algorithm for identifying the entire set of non-dominated solutions for the all to one multiple criteria time-dependent shortest path problem with positive waiting cost values.

The proposed itinerary planning problem is solved by a new backward labeling algorithm based on the algorithmic approach proposed by Chabini [25] for the all to one time-dependent shortest path problem. An enhanced version of the proposed algorithm is also used for solving the composite itinerary planning problem. It is proved that the proposed algorithm outperforms any relevant work in terms of worst case computational performance.

IV. ALGORITHMS FOR THE ITINERARY PLANNING PROBLEMS

A. Solving the Elementary Itinerary Planning Problem

The proposed algorithm moves backward in time starting from the latest arrival time $a_{v_n}^l$ at v_n down to the earliest departure time $d_{v_0}^e$, specifying the lexicographically minimal itineraries from all nodes to node v_n for any possible start time $\tau \in [d_{v_0}^e, a_{v_n}^l]$. The intuition behind this process is that given a departure of service $s \in S$ on the arc $(v_i, v_j) \in A$ at time $\tau_0 - 1$ then the lexicographically minimal itinerary from node v_i to the destination v_n at time $\tau_0 - 1$ can be determined by comparing (in terms of lexicographical ordering) the following alternative solutions: i) the new itinerary departing at time $\tau_0 - 1$ that includes the arc (v_i, v_j) joined with the lexicographically minimal itinerary $p^{\tau'}(v_j, v_n)$ where $\tau' = (\tau_0 - 1) + t_s^{\tau_0 - 1}(v_i, v_j)$, ii) the itinerary that is produced by adding one unit of waiting time at the existing lexicographically minimal itinerary from v_i to v_n departing at time τ_0 , and iii) any other itinerary departing at time $\tau_0 - 1$ that is produced by joining a walking arc (v_i, v_q) with the existing lexicographically minimum itinerary $p^{\tau''}(v_q, v_n)$ where $\tau'' = (\tau_0 - 1) + t_w^{\tau_0 - 1}(v_i, v_q)$. Repeating this process until the earliest departure time $d_{v_0}^e$ is reached, results to the determination of the lexicographically minimal itinerary from every node (including the origin v_0) to the destination v_n for every $\tau \in [d_{v_0}^e, a_{v_n}^l]$. Thus, comparing the emerging itineraries $p^{\tau}(v_0, v_n)$ for $\tau \in [d_{v_0}^e, a_{v_n}^l]$ in terms of lexicographical ordering leads to the lexicographically minimal solution of the itinerary planning problem.

Each itinerary constructed by the above process for node v_r and time τ , is associated with a label $\underline{l}^{\tau}(v_r) := (\lambda_{c_{k_1}}^{\tau}(v_r), \lambda_{c_{k_2}}^{\tau}(v_r), \lambda_{c_{k_3}}^{\tau}(v_r))$. Each attribute of the label $\lambda_{c_{k_i}}^{\tau}(v_r), i = 1, 2, 3$ refers to the criterion value (c_{k_i}) of the

associated itinerary from node v_r to v_n starting at time τ . The order in which the attributes $\lambda_{c_{k_i}}^{\tau}(v_r)$ appear in the above label implies the order of preference of the corresponding criteria. The lexicographically minimal label identified at each step of the algorithm is denoted by $\underline{l}^{\tau}(v_r) := (l_{c_{k_1}}^{\tau}(v_r), l_{c_{k_2}}^{\tau}(v_r), l_{c_{k_3}}^{\tau}(v_r))$ followed by a pointer $\psi^{\tau}(v_r)$ which points to the next node of the corresponding itinerary.

The steps of the proposed algorithm are presented in the logical diagram provided in Fig. 1. Step 0 of the algorithm involves the initialization of the labels $\underline{l}^{\tau}(v_r) \forall \tau \in [d_{v_0}^e, a_{v_n}^l]$, $\forall v_r \in N$. The attributes in $\underline{l}^{\tau}(v_n)$ are set equal to: i) zero for $\tau \in [a_{v_n}^e, a_{v_n}^l]$ and ii) a very large number M for the remaining values of τ . The corresponding pointers $\psi^{\tau}(v_n)$ are set to nil.

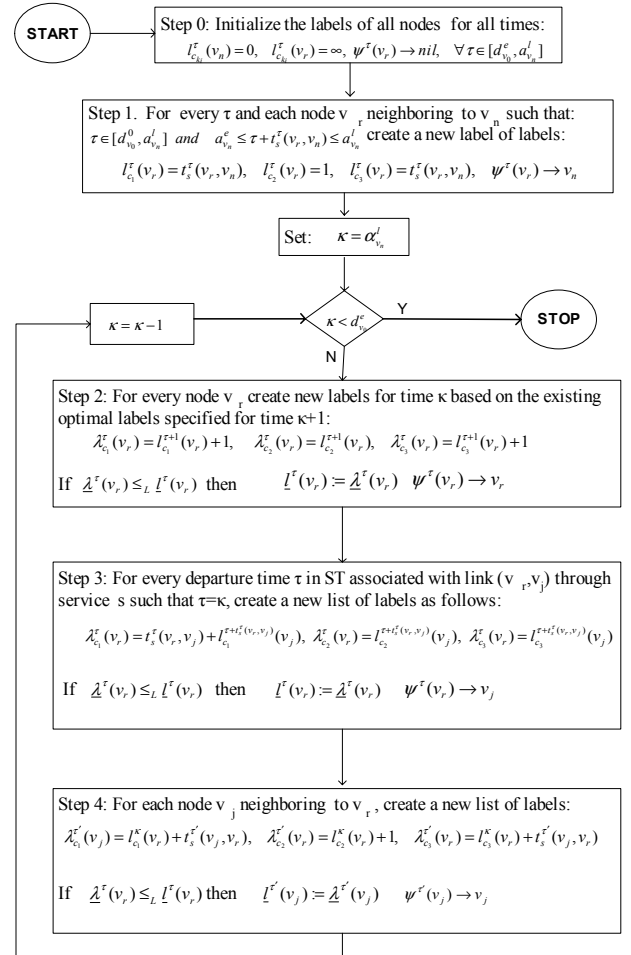


Fig. 1. Logical diagram of the proposed algorithm for solving the (elementary) itinerary planning problem.

The attributes of any other label $\underline{l}^{\tau}(v_r)$ are set equal to a very large number M apart from any node v_r that is neighboring to the destination v_n , i.e., there exists a walking arc $(v_r, v_n) \in A'$. These nodes are passed to step 1, where for

each time $\tau \in [d_{v_0}^e, a_{v_n}^l]$, the attributes $l_{c_1}^\tau(v_r)$ and $l_{c_3}^\tau(v_r)$ expressing the total travel time and transfer time respectively, are set equal to the corresponding walking time $t_w^\tau(v_r, v_n)$ while the attribute $l_{c_2}^\tau(v_r)$ is set equal to 1 since this is the first transfer confronted so far. The corresponding pointers $\psi^\tau(v_r)$ point to the node v_n .

Moving to steps 2-4, the variable κ is a counter that counts time backwards. Initially, κ is set equal to the latest arrival time $a_{v_n}^l$ at the destination. Then, a sequence of $T(= a_{v_n}^l - d_{v_0}^e + 1)$ iterations of steps 2-4 are executed. In Step 2, a new label $\underline{\lambda}^\kappa(v_r)$ is created for each node v_r based on the existing label $\underline{l}^{\kappa+1}(v_r)$. The attributes of the new label are presented by (8)-(10).

$$\lambda_{c_1}^\kappa(v_r) = l_{c_1}^{\kappa+1}(v_r) + 1 \quad (8)$$

$$\lambda_{c_2}^\kappa(v_r) = l_{c_2}^{\kappa+1}(v_r) \quad (9)$$

$$\lambda_{c_3}^\kappa(v_r) = l_{c_3}^{\kappa+1}(v_r) + 1 \quad (10)$$

This label is associated with the new itinerary which involves adding one unit of waiting time to the lexicographically minimal itinerary from v_r to v_n for time $\kappa+1$. If the label $\underline{l}^\kappa(v_r)$ is lexicographically dominated by $\underline{\lambda}^\kappa(v_r)$ then $\underline{l}^\kappa(v_r) := \underline{\lambda}^\kappa(v_r)$ and $\psi^\kappa(v_r) := v_r$ else the values of $\underline{l}^\kappa(v_r)$ are retained.

Step 3 scans the list of departure times of all services for time $\tau = \kappa$. For each departure time found related to service s and arc (v_r, v_j) , a new label $\underline{\lambda}^\kappa(v_r)$ is created based the existing label $\underline{l}^{\kappa+t_s^\kappa(v_r, v_j)}(v_j)$ as indicated by (11)-(13).

$$\lambda_{c_1}^\kappa(v_r) = l_{c_1}^{\kappa+t_s^\kappa(v_r, v_j)}(v_j) + t_s^\kappa(v_r, v_j) \quad (11)$$

$$\lambda_{c_2}^\kappa(v_r) = l_{c_2}^{\kappa+t_s^\kappa(v_r, v_j)}(v_j) \quad (12)$$

$$\lambda_{c_3}^\kappa(v_r) = l_{c_3}^{\kappa+t_s^\kappa(v_r, v_j)}(v_j) \quad (13)$$

This new label refers to the itinerary departing at time κ produced by joining arc (v_r, v_j) , arriving at v_j at time $\tau' = \kappa + t_s^\kappa(v_r, v_j)$ with the lexicographically minimal itinerary from v_j to v_n starting at τ' . If label $\underline{l}^\tau(v_r)$ is lexicographically dominated by $\underline{\lambda}^\kappa(v_r)$ then $\underline{l}^\tau(v_r) := \underline{\lambda}^\kappa(v_r)$ and $\psi^\tau(v_r) := v_j$ else the attributes of $\underline{l}^\tau(v_r)$ are retained.

Finally, if there exists a node v_r from the previous step (step 3), for which the corresponding label $\underline{l}^\tau(v_r)$ has been updated, step 4 is performed. In this step, for any node v_i neighboring to v_r (i.e., $(v_i, v_r) \in A'$), a new label $\underline{\lambda}^{\tau'}(v_i)$ is created as presented in (14)-(16) with τ'' denoting the

earliest departure time from v_i so that v_r is reached at time τ .

$$\lambda_{c_1}^{\tau'}(v_i) = l_{c_1}^\tau(v_r) + t_w^{\tau'}(v_i, v_r) \quad (14)$$

$$\lambda_{c_2}^{\tau'}(v_i) = l_{c_2}^\tau(v_r) + 1 \quad (15)$$

$$\lambda_{c_3}^{\tau'}(v_i) = l_{c_3}^\tau(v_r) + t_w^{\tau'}(v_i, v_r) \quad (16)$$

If label $\underline{l}^{\tau'}(v_i)$ is lexicographically dominated by $\underline{\lambda}^{\tau'}(v_i)$ then $\underline{l}^{\tau'}(v_i) := \underline{\lambda}^{\tau'}(v_i)$ and $\psi^{\tau'}(v_i) := v_r$ else the attributes of $\underline{l}^{\tau'}(v_i)$ are retained.

When the time counter κ reaches $d_{v_0}^e - 1$ then the above set of iterations terminates and the emerging labels for stop v_0 are processed as follows: i) selection of the labels $\underline{l}^\tau(v_0)$ with $\tau \in [d_{v_0}^e, a_{v_0}^l]$, and ii) determination of the lexicographically minimal among the selected labels.

A major feature of the proposed algorithm is that given the κ_0 first iterations described above, the label of any node at time κ , later than the current time κ_0 (i.e., $\kappa > \kappa_0$) is not affected by the remaining iterations of the solution process. This observation implies that there is no need to assume the well known FIFO condition for the underlying network. The definition and lemmas presented below lead to the proof of correctness of the proposed algorithm.

Definition 1. Assume an itinerary $p^\tau(v_0, v_n) := \{[(v_0, v_1); \tau_{v_0}, s_0], (v_1, v_2); \tau_{v_1}, s_1, \dots, [(v_{n-1}, v_n); \tau_{v_{n-1}}, s_{n-1}]\}$ in $G(N, A, A', S, ST)$. If node v_r is reached by $p^\tau(v_0, v_n)$, then the part of the itinerary $p^\tau(v_0, v_n)$ from v_r to v_n starting at time $\tau' \in [\tau_{v_{r-1}} + t_{s_{v_{r-1}}}^\tau(v_{r-1}, v_r), \tau_{v_r}]$ is called sub-itinerary of $p^\tau(v_0, v_n)$ from v_r to v_n .

Definition 2. A label $\lambda^\tau(v_r)$ is lexicographically minimal iff its associated itinerary $p^\tau(v_r, v_n)$ is lexicographically minimal.

Lemma 1. If an itinerary $p^\tau(v_r, v_n)$ is lexicographically minimal for time τ then any of its sub-itineraries $p^{\tau'}(v_k, v_n)$, $\tau' > \tau$ from v_k to v_n is also lexicographically minimal for τ' .

Lemma 1 expresses a modified version of the well known Bellman's optimality condition for dynamic programming. Note that Lemma 1 implies that the problem at hand could be solved through a backward dynamic programming algorithm. On the other hand, a similar optimality condition cannot be

TABLE I
RESULTS OF THE NUMERICAL EXAMPLE.

	s	A1	B1	A2	B2	A3	B3	t
14	(M,M,M)	(M,M,M)	(M,M,M)	(M,M,M)	(M,M,M)	(M,M,M)	(M,M,M)	(0,0,0);-
13	(M,M,M)	(M,M,M)	(M,M,M)	(M,M,M)	(M,M,M)	(M,M,M)	(M,M,M)	(0,0,0);-
12	(M,M,M)	(M,M,M)	(M,M,M)	(M,M,M)	(M,M,M)	(2,2,1);-	(2,2,1);-	(0,0,0);-
11	(M,M,M)	(M,M,M)	(M,M,M)	(M,M,M)	(M,M,M)	(2,2,1);-	(2,2,1);-	(0,0,0);-
10	(M,M,M)	(M,M,M)	(M,M,M)	(M,M,M) (4,2,1);A3	(M,M,M)	(2,2,1);-	(2,2,1);-	(0,0,0);-
9	(M,M,M)	(M,M,M)	(M,M,M)	(5,3,1);A2	(M,M,M) (5,3,2);A2	(2,2,1);-	(2,2,1);-	(0,0,0);-
8	(M,M,M)	(M,M,M) (6,2,1);A2	(M,M,M)	(6,4,1);A2	(6,4,2);B2	(2,2,1);-	(2,2,1);-	(0,0,0);-
7	(M,M,M) (7,3,2);A1	(7,3,1);A1	(M,M,M)	(7,5,1);A2	(7,5,2);B2 (3,2,1);B3	(2,2,1);-	(2,2,1);-	(0,0,0);-
6	(8,4,2);s	(8,4,1);A1	(M,M,M)	(4,3,2);B2 (8,6,1);A2 (4,2,1);A3	(4,3,1);B2	(2,2,1);-	(2,2,1);-	(0,0,0);-
5	(9,5,2);s	(9,5,1)A1	(M,M,M)	(5,3,1);A2	(5,3,2);A2 (4,2,1);B3	(2,2,1);-	(2,2,1);-	(0,0,0);-
4	(10,6,2);s	(10,6,1);A1 (6,2,1);A2	(M,M,M) (6,3,1);B2	(5,3,2);B2	(5,3,1);B2	(2,2,1);-	(2,2,1);-	(0,0,0);-
3	(11,7,2);s (7,3,2);A1	(7,3,1);A2	(7,7,2);A1 (7,4,1);B1	(6,4,2);A2	(6,4,1);B2	(2,2,1);-	(2,2,1);-	(0,0,0);-
2	(8,4,2);s	(8,4,1);A1	(8,5,1);B1 (7,3,1);B2	(7,5,2);A2	(7,5,1);B2	(2,2,1);-	(2,2,1);-	(0,0,0);-
1	(9,5,2);s	(9,5,1);A1	(8,4,1);B1	(8,6,2);A2	(8,6,1);B2	(2,2,1);-	(2,2,1);-	(0,0,0);-
0	(10,6,2);s (9,5,2);B1	(10,6,1);A1	(9,5,1);B1	(9,7,2);A2	(9,7,1);B2	(2,2,1);-	(2,2,1);-	(0,0,0);-

sustained for the forward case, i.e. if an itinerary $p^\tau(v_r, v_n)$ is lexicographically minimal for time τ then it cannot be proved that any of its sub-itineraries $p^{\tau'}(v_0, v_k)$ from v_0 to v_k is also lexicographically minimal for τ' . See also in [26] for more details on this issue.

Lemma 2. If labels $\underline{\lambda}^\tau(v_r)$, $\tau \geq \tau_0 + 1$ where $\tau_0 \in [d_{v_0}^e, a_{v_n}^l - 1]$ for every node $v_r \in N$ are lexicographically minimal for time τ , then any label $\underline{l}^{\tau_0}(v_r)$ determined by the proposed algorithm is also lexicographically minimal.

Proof: Assume that for some node v_r the corresponding label created by the proposed algorithm is lexicographically dominated, i.e., there exists an alternative itinerary with label $\underline{\lambda}^{\tau_0}(v_r) := (\lambda_{c_{k_1}}^{\tau_0}(v_r), \lambda_{c_{k_2}}^{\tau_0}(v_r), \lambda_{c_{k_3}}^{\tau_0}(v_r))$ such that $\exists i \in \{1, 2, 3\}$: $\lambda_{c_{k_j}}^{\tau_0} \leq \lambda_{c_{k_j}}^{\tau_0}$, $j < i$ and $\lambda_{c_{k_i}}^{\tau_0} < \lambda_{c_{k_i}}^{\tau_0}$. However, any itinerary from v_r to v_n at time τ_0 (including the itinerary associated with $\underline{\lambda}^{\tau_0}(v_r)$) could be analyzed in one of the following ways: i) one unit of waiting time at v_r and then follow the lexicographically minimal sub-itinerary (see Lemma 1) from v_r to v_n at time $\tau_0 + 1$, ii) proceed from v_r to v_j through a walking arc $(v_r, v_j) \in \mathcal{A}'$ and then follow the lexicographically minimal sub-itinerary from v_j to v_n at time $\tau' = \tau_0 + t_w^{\tau_0}(v_r, v_j)$, or iii) depart from v_r to v_k through service s and continue from v_k at time $\tau'' = \tau_0 + t_s^{\tau_0}(v_r, v_k)$

through a lexicographically minimal sub-itinerary leading to v_n . In any of the above cases the labels of the itineraries $p_\lambda^{\tau_0}(v_r, v_n)$, $p_\lambda^{\tau'}(v_j, v_n)$, $p_\lambda^{\tau''}(v_k, v_n)$ are identical to the labels associated with the corresponding lexicographically minimum itineraries $p_i^{\tau_0}(v_r, v_n)$, $p_i^{\tau'}(v_j, v_n)$, $p_i^{\tau''}(v_k, v_n)$ produced by the proposed algorithm. However, by the completion of $(\tau_0)^{th}$ iteration of the proposed algorithm, the above cases have already been specified and assessed and only the lexicographically minimal is held. Thus $\underline{l}^{\tau_0}(v_r, v_n)$ coincides with $\underline{\lambda}^{\tau_0}(v_r, v_n)$.

Proposition 2 (Proof of Correctness). The labels for each $\tau \in [d_{v_0}^e, d_{v_0}^l]$ determined by the proposed algorithm for the origin v_0 are lexicographically minimal for time τ .

Proof. The proof may be derived by induction making use of lemma 2.

Proposition 3. The worst case computational complexity of the proposed algorithm is $O(|N|^3 |S| T)$ where $|N|$ denotes the number of nodes of the time-schedule network, $|S|$ is the number of services under consideration and T denotes the time steps between the latest arrival time at the destination and the earliest departure time from the origin (i.e., $T := a_{v_n}^l - d_{v_0}^e + 1$).

Proof: Steps 0-1 are executed only once at the beginning of the algorithm. Step 0 involves at most $O(|N|T)$ operations while step 1 $O(|N|^2T)$ operations. Steps 2-4 are iterated at most T times. Step 2 includes $O(|N|)$ operations, step 3 $O(|N||S|)$ and step 4 $O(|N|^3|S|)$ operations at most. Therefore, the iterations of steps 2-4 require at most $O(|N|^3|S|T)$ operations.

It is evident that the proposed algorithm depends only linearly on the width of the time horizon T as opposed to the complexity of Ziliaskopoulos and Wardell's algorithm [20] which depends on T^2 .

B. Numerical Example

The steps of the proposed algorithm are illustrated and explained through the example that follows. Fig. 2 presents the routes of two transport services A and B and the associated timetable of their operation. Nodes s and t denote the origin and destination points respectively. The stops of transport service A (A1, A2, A3) are denoted by octagons while circles are used for the stops of transport service B (B1, B2, B3). Any possible transfer between two nodes is represented by a dimmed line while the corresponding walking time is written beside each arc. The objective of the itinerary planning problem under consideration is to determine the itinerary from s to t that lexicographically minimizes the total travel time (c_1), the total walking and waiting time (c_3), and the number of transfers (c_2) in that order, under the following scheduling constraints: i) the earliest departure time is 0, i.e., $d_s^e := 0$, and ii) the latest arrival time is 14, i.e., $a_t^l := 14$. No latest departure and earliest arrival times are taken into account, i.e., $d_s^l := 14$ and $a_t^e := 0$.

Table I presents the labels $(\lambda_{c_1}^r(v_r), \lambda_{c_3}^r(v_r), \lambda_{c_2}^r(v_r))$ that were calculated for every node v_r and time $\tau \in [0,14]$ during the solution of the problem, followed by the corresponding pointer $\psi^r(v_r)$ (separated from the label with a semicolon). The first column of the table presents the time steps of the problem while the first row contains the nodes of the network under consideration. Each of the remaining cells of the table includes the labels created throughout the entire execution of the solution process written in the order that they were created. A strikethrough is placed on the vectors which were found lexicographically dominated.

The completion of the first row is equivalent to initializing the labels of all nodes at time 14 (step 0), i.e., node t gets the label (0,0,0) while the remaining nodes get the label (M,M,M) (where M is a very large number). Given the label

of node t at time 14, nodes A3 and B3 are assigned the label (2,2,1) for times 0-12 (step 1). These labels refer to the trivial itineraries from nodes A3 and B3 to node t through the corresponding walking arcs (B3,t) and (A3,t).

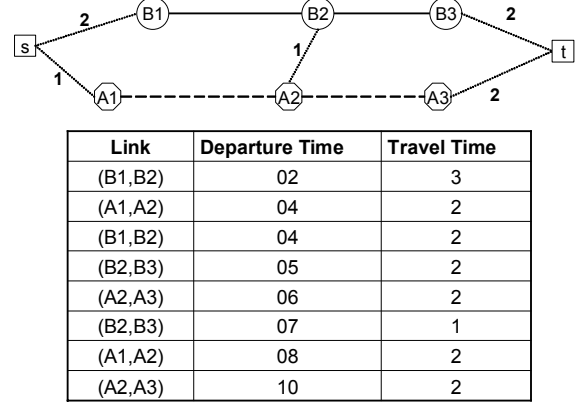


Fig. 2. The network and the timetable of the numerical example.

The remaining cells of the table get the label (M,M,M). Each iteration of steps 2-4 from time 13 down to 0, is equivalent in completing the corresponding row of the table starting from the cell referring to B2 up to the cell referring to s. By the end of the execution of these iterations, the second column of the table (referring to node s) includes the lexicographically minimal labels of the problem for each point in time $\tau \in [0,14]$. Label (7,3,2) is the lexicographically minimal among them which is realized by two alternative itineraries:

$$p_1^3(s,t) := \{[(s,A1);3;w], [(A1,A2);4;A], [(A2,A3);6;A], [(A3,t);8;w]\} \quad (17)$$

$$p_2^7(s,t) := \{[(s,A1);7;w], [(A1,A2);8;A], [(A2,A3);10;A], [(A3,t);12;w]\} \quad (18)$$

C. Solving the Composite Itinerary Planning Problem

Any solution of the composite itinerary planning problem consists of two elementary (i.e., with no intermediate stop) itineraries: i) $p^r(v_0, v_q)$ which departs from v_0 within $[d_{v_0}^e, d_{v_0}^l]$ and terminates at the intermediate node v_q within $[a_{v_q}^e, a_{v_q}^l]$, and ii) $p^r(v_q, v_n)$ which departs from the intermediate node within $[d_{v_q}^e, d_{v_q}^l]$ but later than the time $(a_{v_q}^e + t_v)$ while it arrives at the destination v_n within $[a_{v_n}^e, a_{v_n}^l]$. Thus, the composite itinerary planning problem may be decomposed to the following nested optimization problems: (P1) Determine the lexicographically minimal

itineraries from the intermediate node v_q to node v_n satisfying the above time windows constraints, (P2) Determine the lexicographically minimal among the itineraries from node v_0 to node v_n created by appending to any itinerary $p^\tau(v_0, v_q)$ arriving in v_q at time τ'' , the lexicographically optimal itinerary from v_q to v_n for time $\tau'' + t_{v_q}$ determined in (P1).

Given this intuition, the proposed solution process for the composite itinerary planning problem consists of two sequential stages, i.e., stage I and II. The objective of stage I is to determine the lexicographically minimal itineraries from the intermediate node v_q to the destination v_n for each τ in $[d_{v_q}^e, a_{v_q}^l]$. Stage II aims to determine the itinerary from the origin v_0 at time $\tau \in [d_{v_0}^e, d_{v_0}^l]$ to v_q arriving at $\tau' \in [a_{v_q}^e, a_{v_q}^l]$ which when attached to the lexicographically optimal itinerary from v_q to v_n for departure time $\tau' + t_v$, provides a lexicographically minimal solution for time τ . The algorithm for solving the elementary itinerary planning problem enhanced with some modifications in step 0, is used in both stages in order to determine the lexicographically optimal solution to the composite problem.

The presentation of the steps of the proposed solution approach requires the introduction of $\underline{m}^\tau(v_r) = (m_{c_{k_1}}^\tau(v_r), m_{c_{k_2}}^\tau(v_r), m_{c_{k_3}}^\tau(v_r))$ which denotes the lexicographically minimal label for any stop v_r involved in the stage I of the solution process. Each such label refers to the lexicographically minimal itinerary from stop v_r to v_n for $\tau \in [d_{v_q}^e, a_{v_n}^l]$. Any label produced during the execution of stage I of the solution process is denoted by $\underline{\mu}^\tau(v_r) = (\mu_{c_{k_1}}^\tau(v_r), \mu_{c_{k_2}}^\tau(v_r), \mu_{c_{k_3}}^\tau(v_r))$. For stage II, $\underline{l}^\tau(v_r) = (l_{c_{k_1}}^\tau(v_r), l_{c_{k_2}}^\tau(v_r), l_{c_{k_3}}^\tau(v_r))$ denotes the label for any stop v_r while it refers to the composite itinerary from any node v_r to node v_n for $\tau \in [d_{v_0}^e, a_{v_n}^l]$. Any label produced during the execution of stage II of the solution process is denoted by $\underline{\lambda}^\tau(v_r) = (\lambda_{c_{k_1}}^\tau(v_r), \lambda_{c_{k_2}}^\tau(v_r), \lambda_{c_{k_3}}^\tau(v_r))$. During stage II, it is assumed that once the itinerary from v_r reaches node v_q at time τ' , then the corresponding optimal itinerary from stop v_q to v_n (determined in stage I) starting at time $\tau' + t_v$, is attached to it.

Stage I of the proposed algorithm for determining the lexicographically minimal composite itinerary is identical to the algorithm for the elementary itinerary planning problem. The output of stage I includes the lexicographically minimum labels $\underline{m}^\tau(v_q)$ from v_q to v_n for $\tau \in [d_{v_q}^e, a_{v_q}^l]$.

On the other hand, Stage II involves the following steps:

1. Initialize the labels ($\underline{l}^\tau(v_r)$) which refer to the itineraries from any stop v_r to the intermediate stop v_q . In particular, the attributes of any label of the intermediate stop v_q are set as follows: $\underline{l}^\tau(v_r) := \underline{m}^{\tau+t_v}(v_r)$ for $\tau \in [d_{v_q}^e, a_{v_q}^l]$ and $\tau + t_v \in [d_{v_q}^e, a_{v_q}^l]$. For any node v_x that has an outgoing walking arc to v_q , then its attributes of its labels for $\tau \in [d_{v_0}^e, a_{v_q}^l]$ such that $\tau + t_w^\tau(v_j, v_q) = \tau'' \in [a_{v_q}^e, a_{v_q}^l]$, are set as follows:

$$l_{c_1}^\tau(v_x) = l_{c_1}^{\tau''}(v_q) + t_w^\tau(v_x, v_q) \quad (19)$$

$$l_{c_2}^\tau(v_x) = l_{c_2}^{\tau''}(v_q) + 1 \quad (20)$$

$$l_{c_3}^\tau(v_x) = l_{c_3}^{\tau''}(v_q) + t_w^\tau(v_x, v_q) \quad (21)$$
2. The attributes of the label of any other node are set equal to a very large number.
3. Perform $T' (= a_{v_q}^l - d_{v_0}^e + 1)$ iterations of steps 2-4 of the algorithm for the itinerary planning problem in order to determine the lexicographically minimal labels for every time $\tau \in [d_{v_0}^e, a_{v_0}^l]$.
4. Determine the lexicographically minimal composite itinerary by comparing the emerging itineraries (step 3 above) from v_0 to v_n for any time $\tau \in [d_{v_0}^e, a_{v_0}^l]$.

The first stage of the proposed algorithm ends up with a set of labels $\underline{m}^\tau(v_q)$, $\tau \in [d_{v_q}^e, a_{v_q}^l]$. Each label $\underline{m}^\tau(v_q)$ refers to the lexicographically minimal itinerary from v_q to v_n when departing at time τ . Step 2 of stage II, terminates with a set of labels \underline{l}^τ , $\tau \in [d_{v_0}^e, a_{v_0}^l]$ which refer to the lexicographically minimal itineraries from v_0 to v_n departing at any time $\tau \in [d_{v_0}^e, a_{v_0}^l]$. Selecting the lexicographically minimal among the emerging composite itineraries provides the solution to the problem.

V. COMPUTATIONAL PERFORMANCE

Given the fact that the increased time in determining the optimal itinerary decreases the utility of the journey planning service provided to the traveler, the computational time of the proposed algorithms constitutes a critical success factor for their integration in an on-line journey planning decision support system.

Both algorithms were implemented in Pascal (Delphi 7.0 Architect) while the assessment of their computational performance was based on their application for solving real life itinerary planning problems defined on the Athens Public Transport Network. Given that the computational complexity of the algorithms depends on the width (T) of the time

window $[d_{v_0}^e, a_{v_0}^l]$, a wide range of test problems has been developed for various values of the aforementioned parameter. In particular, the test problems were built on the Athens Public Transportation Network covering the whole geographical urban area of Athens. The underlying network, consists of 890 stops, 45 public transport services (including metro, bus, tram, and trolley bus lines), operated based on a timetable with approximately 42000 vehicle departures occurring on a daily basis. Seven alternative values for the time window width (T), i.e., $T_1=120$ min, $T_2=180$ min, $T_3=240$ min, $T_4=300$ min, $T_5=360$ min, $T_6=420$ min, and $T_7=480$ min were used. The criteria and their order of preference were randomly determined (two out of the three considered in this set of test problems) for each problem separately. In particular, 200 different itinerary planning problems were generated for every alternative value of the parameter (T). Moreover, another 200 test problems were developed for the composite itinerary planning problem (i.e., including a mandatory intermediate visit), with T ranging from 180 min to 480 min as above. Note that the alternative value $T=120$ min was omitted since it led to many infeasible test problems.

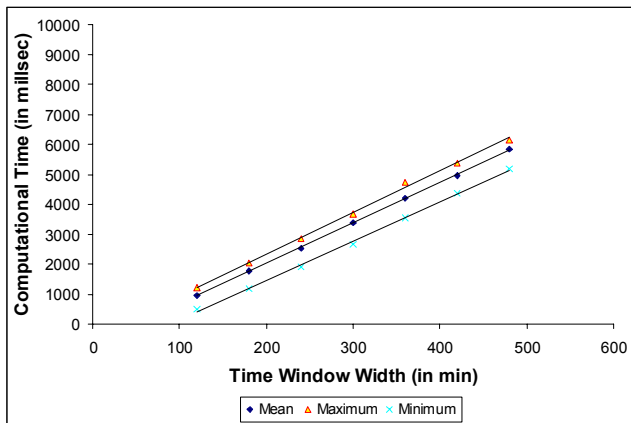


Fig. 3. Graphical representation of the mean, maximum, and minimum values of the computational time for the itinerary selection problems for each alternative problem category (indicated by the corresponding T).

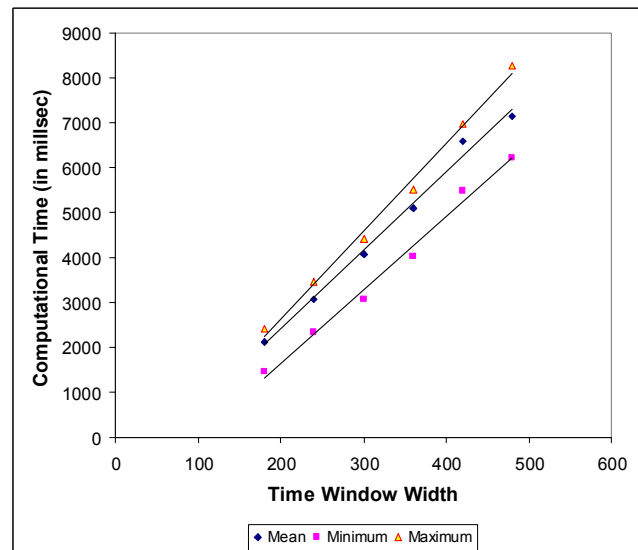


Fig. 4. Graphical representation of the mean, maximum, and minimum values of the computational time for the composite itinerary planning problems for each alternative problem category (indicated by the corresponding T)

The performance of the algorithms was assessed in terms of the following statistical measures of the CPU computational time: i) the mean, ii) the minimum value and iii) the maximum value. Additional statistical measures were calculated (i.e., standard error, skewness, kurtosis, median, and range) in order to explore further the central tendency of the computational time.

The associated runs for solving the test problems were performed on a Pentium IV PC with 2 GB of RAM and a 2.6 GHz Intel processor. Fig. 3 and 4 present the graphs of the mean, maximum, and minimum values for each category of test problems. Based on the results presented in these two graphs, the computational time of the proposed algorithms increases gradually as a function of the parameter (T). Furthermore, it is realized that the solution of the real life itinerary planning problems under consideration can be attained within reasonable computational time (i.e., the worst case computational times are less than 7sec for the elementary itinerary planning problems and less than 9sec for the composite test problems).

Table II presents additional statistical measures of the computational time measurements collected for the elementary itinerary planning test problems. Based on these results, the following observations emerge:

- The largest value of the range of the measurements across the problem categories does not exceed 1sec
- The kurtosis is positive under every problem category. Given that the corresponding kurtosis for the normal distribution is equal to 0, this observation implies that the corresponding histograms of the measurements are highly peaked.

TABLE II
STATISTICAL MEASURES OF THE COMPUTATIONAL TIME FOR THE ITINERARY
PLANNING TEST PROBLEMS.

	120	180	240	300	360	420	480
	min	min	min	min	min	min	min
Mean	959	1777	2531	3377	4193	4951	5826
Standard Error	9	10	12	12	12	13	12
Median	969	1796	2547	3406	4203	4999	5858
Standard Deviation	129	145	170	173	168	179	163
Kurtosis	0,07	1,69	1,25	0,81	1,33	0,45	1,31
Skewness	-0,5	-0,9	-0,9	-0,8	-0,5	-0,7	-0,9
Range	704	859	937	1001	1202	999	968

All times are in millsec.

The above observations imply that the vast majority of the measurements under each problem set lie very close to the corresponding mean value.

Table III presents the corresponding additional statistical measures for the measurements referring to the composite itinerary planning test problems. Although the worst case range reaches 2,67 sec the kurtosis under each problem category is positive, intuitively implying that the corresponding measurements lie close to the mean value.

TABLE III
STATISTICAL MEASURES OF THE COMPUTATIONAL TIME FOR THE
COMPOSITE ITINERARY PLANNING TEST PROBLEMS.

	180	240	300	360	420	480
	min	min	min	min	min	min
Mean	2134	3041	4079	5116	6591	7105
Standard Error	12,6	16,1	17,2	18,7	18,6	17,3
Median	2148	3078	4116	5141	6641	7140
Standard Deviation	159	220	244	264	263	244
Kurtosis	2,84	1,81	2,52	8,19	2,59	9,55
Skewness	-1,22	-1,11	-1,06	0,16	-1,48	0,18
Range	952	1267	1672	2673	1533	2439

All times are in millsec.

VI. CONCLUSION

The journey planning problem in an urban public transport network has been formulated as a shortest path problem with time windows on a multimodal time schedule network that optimizes lexicographically the en-route time, the number of interchanges and the total walking and waiting time (not necessarily in this order). This new formulation reflects the traveler's decision taking problem in selecting the point in time to start his/her urban journey within a specified time window and the sequence of transport legs in order to reach destination in time. The incorporation of this flexibility in departing from the origin enhances the solution space of the problem providing efficient journey planning decisions. Lexicographical ordering is utilized for evaluating the alternative itineraries instead of applying the time consuming task of determining the efficient frontier of the problem. This

limitation ensures the fast determination of a single itinerary and relieves the traveler from the task of selecting on his own the most preferred itinerary among the set of the non-dominated solutions. The examination of the efficient frontier of the itinerary selection problem is highly motivated for the case of interurban journey planning decisions (i.e., itinerary selection on a fixed scheduled network including interurban and urban transport services) where the cost of travel should also be taken into account apart from the criteria considered in this paper.

A dynamic programming based algorithm was developed for determining the lexicographically optimal itinerary. Furthermore, the algorithm was enhanced in order to cover the variant of the problem that included the visit of an intermediate stop within a specified time window. The proposed algorithms were tested on a wide range of real life itinerary planning problems (with and without intermediate stop) defined on the Urban Public Transportation System of Athens, Greece. The width of the time window defined by the earliest departure time from the origin and the latest arrival time at the destination ranged from two to eight hours. The scope of these tests was to verify that the computational time of the algorithms is not prohibitive in integrating them within an on-line journey planning decision support system. The worst case mean value of the computational time of the algorithm on the simple itinerary selection problems was less than 7 sec while the corresponding performance on the problems with intermediate stops was 9 sec. These results justified the integration of both algorithms in an on-line advanced public transport information system that provided urban and interurban journey planning services via alternative communication channels (i.e., the web and mobile phones).

REFERENCES

- [1] J. Adler, V. Blue, "Toward the design of intelligent traveller information systems", *Transportation Research C*, vol. 6, pp. 157-172, 1998.
- [2] Z.R Peng, R. Huang, "Design and Development of interactive trip planning for web-based transit information systems", *Transportation Research C*, vol. 8, pp. 409-425, 2000.
- [3] Z.R. Peng, "A methodology for design of GIS-based automatic transit traveller information systems", *Computers Environment and Urban Systems*, vol. 21, no. 5, pp. 359-372, 1997.
- [4] R.F. Casey, L.N. Labell, S.P. Prenskey, C.L. Schweiger, "Advanced Public Transportation Systems: The State of the Art Update '98". Federal Transit Administration, Washington, 1998.
- [5] H. K. Lo, C. W. Yip and B. Mak, "Passenger route guidance system for multi-modal transit networks," *Journal of Advanced Transportation*, vol. 39, pp. 271-288, 2005.
- [6] B. Caulfield and M. O'Mahony, "An examination of the public transport information requirements of users," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, pp. 21-30, 2007.
- [7] K. Rehr, S. Bruntsch and H. J. Mentz, "Assisting multimodal travelers: Design and prototypical implementation of a personal travel companion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, pp. 31-42, 2007.
- [8] P. Pattanamekar, D. Park, L. Rilett, J. Lee, C. Lee, "Dynamic and stochastic shortest path in transportation networks with two components of travel time uncertainty", *Transportation Research C*, vol. 11, no. 5, pp. 331-354, 2003.

- [9] P. Hansen, "Bicriterion path problems. In Multiple Criteria Decision Making Theory and Application", *Lecture Notes in Economics and Mathematical Systems*, vol. 177, Springer, Berlin Heidelberg New York, 1978, pp. 109-127.
- [10] S.C. Wong, C.O. Tong, "Estimation of time-dependent origin-destination matrices for transit networks", *Transportation Research B*, vol. 32, no. 1, pp. 35-48, 1998.
- [11] P. Modesti, A. Siomachen, "A utility measure for finding multiobjective shortest paths in urban multimodal transportation networks", *European Journal of Operational Research*, vol. 111, no. 3, pp. 495-508, 1998.
- [12] R. Huang, Z.R. Peng, "An Integration of Network Data Model and Routing Algorithms for Online Transit Trip Planning", in *Proc. of the 80th Annual Meeting of the Transportation Research Board (CD-ROM)*, Washington D.C., 2001.
- [13] R. Huang, Z.R. Peng, "Schedule-Based Path Finding Algorithms for Online Transit Trip Planning", in *Proc. of the 81st Annual Meeting of the Transportation Research Board (CD-ROM)*, Washington D.C., 2002.
- [14] M. Horn, "An extended model and procedural framework for planning multi-modal passenger journeys", *Transportation Research Part B*, vol. 37, pp. 641-660, 2003.
- [15] C.O. Tong, A.J. Richardson, "A computer model for finding the time-dependent minimum path in transit systems with fixed schedules", *Journal of Advanced Transportation*, vol. 18, pp. 145-161, 1984.
- [16] C.O. Tong, S.C. Wong, "A stochastic transit assignment model using a dynamic schedule-based network", *Transportation Research B*, vol. 33, no. 2, pp. 107-121, 1999.
- [17] J. Bander, C.C. White, "A New Route Optimization Algorithm for Rapid Decision Support", in *Proc. of the Vehicle Navigation and Information Systems Conference Part 2*, vol. P-253, 1991, pp. 709-728.
- [18] N. Koncz, J. Greenfeld, K. Mouskos, "A Strategy for solving static Multiple-Optimal-Path Transit Network Problems", *ASCE Journal of Transportation Engineering*, vol. 122, no. 3, pp. 218-225, 1996.
- [19] M. C. Tan, C. O. Tong, S. C. Wong and J. Xu, "An algorithm for finding reasonable paths in transit networks," *Journal of Advanced Transportation*, vol. 41, pp. 285-305, 2007.
- [20] A. Ziliaskopoulos, 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.
- [21] J. Bérubé, J. Potvin, J. Vaucher, "Time-dependent shortest paths through fixed sequence of nodes: application to a travel planning problem", *Computers and operations Research*, vol.33, pp. 1838-1856, 2006.
- [22] K.L. Cooke, E. Halsey, "The Shortest Route Through a Network with Time-Dependent Intermodal Transit Times", *Journal of Mathematical Analysis and Applications*, vol. 14, pp. 493-498, 1966.
- [23] A. Ziliaskopoulos, H. Mahmassani, "Time-dependent, Shortest-Path Algorithm for Real-Time Intelligent Vehicle Highway System Applications", *Transportation Research Record*, vol. 1403, pp. 94-100, 1993.
- [24] X. Cai, T. Kloks, C.K. Wong, "Time-Varying Shortest Path Problems with Constraints", *Networks*, vol. 29, no. 3, pp. 141-149, 1997.
- [25] I. Chabini, "Discrete dynamic shortest path problems in transportation applications", *Transportation Research Record: Journal of the Transportation Research Board*, no. 1645, pp. 170-175, 1998.
- [26] M. Kostreva, M. Wiecek, "Time Dependency in Multiple Objective Dynamic Programming", *Journal of Mathematical Analysis and Application*, vol. 173, pp. 289-307, 1993.
- [27] H. Hamacher, S. Ruzika, S. Tjandra, "Algorithms for Time-Dependent Bicriteria Shortest Path Problems", *Discrete Optimization*, vol. 3, pp. 238-254, 2006.



Konstantinos G. Zografos is Professor and Chairman of the Department of Management Science and Technology, Athens University of Economics and Business (AUEB), Director of the Transportation Systems and Logistics Laboratory (TRANSLOG), and Coordinator of the Transportation and Logistics Specialization of the MBA International Programme at the Athens University of Economics and Business. His research, teaching, and professional interests are focused on the applications of operations research and information systems in transportation systems and supply chain management. He has published more than 100 papers, and has been a principal investigator for

more than 60 R&D projects funded by the European Commission, and the Greek Government.

Professor Zografos is a member of the Editorial Board of *Transportation Research – Part C*, and *Simulation Modelling Practice and Theory Journal*. He is a recipient of the ENO foundation for transportation fellowship, the 2005 Franz Edelman Finalist Award of The Institute For Operations Research, and Management Sciences (INFORMS) for Achievement in Management Science and Operations Research, and of the 2005 President's Medal Award of the British Operational Research Society.



Konstantinos N. Androutopoulos received the B.Sc. degree in Mathematics from the University of Athens, Athens, Greece, in 1995, the M.Sc. degree in Statistics and Operational Research from the University of Essex, Colchester, UK in 1996, and the Ph.D. degree in Management Science and Technology from the Athens University of Economics and Business, Athens Greece, in 2003. He is currently a research associate in the Transportation Systems and Logistics Laboratory (TRANSLOG).

His research interests include: Applications of Operations Research and Information Systems in Transportation Systems and Logistics.