Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 692–697

Procedia Computer Science

www.elsevier.com/locate/procedia

The 8th International Conference on Ambient Systems, Networks and Technologies (ANT 2017)

# A new time-dependent shortest path algorithm for multimodal transportation network

Abdelfattah IDRI*, Mariyem OUKARFI[a], Azedine BOULMAKOUL[a], Karine ZEITOUNI[c] and Ali MASRI[c]

[a]*LIM Lab. IOS, Computer Sciences Department, Faculty of Sciences and Technology Mohammedia, Mohammedia, Morocco*
[b]*National School of business and Management, Casablanca, Morocco*
[c]*David Lab. University of Versailles Saint-Quentin-en-Yvelines, Versailles, France*

**Abstract**

The shortest path problem is a classical combinatorial optimization problem with countless real-life applications. Numerous algorithms have been proposed to solve the SPP since the classical Dijkstra algorithm, as well as new approaches to improve the algorithms' optimality. But according to the real-world transport conditions, researchers began to study variants of this problem which include the time-dependent SPP and the multimodal networks. In this paper, we introduce our new time-dependent shortest path algorithm for multimodal transportation network. The proposed algorithm is a goal-oriented single-source single-destination algorithm. It takes into account the concept of "closeness" to the target node as heuristic to drive the search towards its destination. The optimality of the algorithm is principally based on computing a virtual path which is basically an Euclidean distance from the source to the target aiming at a restriction of the search space.



*Keywords:* time-dependent shortest path algorithm; multimodal transportation network; time-dependent network

## 1. Introduction

The transport field is confronting a number of big challenges over the coming years and decades and is challenged to keep up with the pace of the mobility growth, the development of transport infrastructure and the speed of technological change. Accordingly, intelligent transport systems (ITS) came to revolutionize transport globally by

* Corresponding author. Tel.: +212662038857 ; Fax: 212 (05) 23315353
*E-mail address:* abdelfattah.id@gmail.com


10.1016/j.procs.2017.05.379

offering the best prospects for improvements in safety, efficiency and environmental outcomes. An efficient ITS is based on an efficient spatial planning system that basically uses a network flow optimization solution responding to user itinerary request. Travelers ask for efficient routing methods allowing them to reach their destinations through highly developed transportation network involving different transportation modes. Multimodal transport network is the combination of two or more means of transport to move passengers or goods from one source to a destination. More and more people tend to use public transport or soft means of transport as walking or cycling and could combine different means of transport. However, finding appropriate travel routes manually is complicated mainly because it is difficult to determine proper transfer points between different routes. It is one of the most interesting applications of the shortest path problems and needs to be solved under additional constraints such as dynamic changing traveling data, or on a particular structured graph. Due to the increasing interest in the dynamic management of transportation systems, there are needs to find shortest paths over a large graph (e.g., a road network), where the weights (or delays) associated with edges dynamically change over time (time-dependency). This new perspective based on dynamically changing parameters of the travel time had led the transformation from the static case of multimodal routing problem to the dynamic routing. We study in this paper the dynamic form of route planning to find the optimal path from a source to a destination, by introducing our proposed constraint based shortest path algorithm over a time-dependent multimodal graph. The remainder of this paper is organized as follows. We introduce some related works in Section 2. Section 3 describes the time-dependent shortest path problem (TDSP) in multimodal network. We present in Section 4 our shortest path algorithm accompanied with illustrative example. Experimental results are presented in Section 5. Finally, we conclude this paper, as well as we propose some future works.

## 2. Related works

For solving the time-dependent shortest path problem in the intricate multimodal transportation network, several approaches have been proposed. Some of the researchers used the hypergraph theory[1,2,3], others used multi-label networks to model the multimodal transportation network. Besides, several algorithms to compute the shortest path in such networks are given in the literature. The classical Dijkstra[5] has known many improvements since 1956 for routing in static networks, such as speed up techniques[6,9,12]. However, routing in static networks has become quite easy compared to dynamic networks. Public transit networks such as bus and railway are inherently time-dependent. The travel time from one station to another is not static; it depends on the arrival time of the user at the departure station. Cooke and Halsey[8] transformed the standard algorithm Dijkstra into dynamic algorithm to cope with the time dependency aspect of public transportation modes but in the expense of additional computational efforts such as the networks size[10,11]. Another extensive works have been done to speed up techniques to accommodate with time-dependent networks[13]. Metaheuristics[14,15] such as Genetic algorithms have been also used to solve this routing issue. Therefore, there has been an application need to develop new routing approaches that provide optimal routes in reasonable computational time in large-scale multimodal networks, as well as, that cope with additional problem constraints such as stochastic arcs' weights, multi-criteria optimization etc.

## 3. Problem description

The conventional routing issue of finding the shortest path in a directed graph is defined as follows: Let $G = (V, E)$ denotes a directed graph, where $V = \{v_1, \dots, v_n\}$ a set of vertices, $E = \{e_1, \dots, e_m\}$ a set of edges, $c(e_i)$ the cost associated to the edge $e_i$, a source vertex $s \in E$, a target vertex $t \in E$ and a departure time $t_0$. We ask then for finding a minimum path $\wp = \{e_1, e_2, \dots, e_k\}$ among all the other routes that respect the properties: $\wp$ starts at $s$ in the departure time $t_0$ and ends at $t$. However, in this paper we are interested on defining the variants of the conventional routing issue, which are the multimodal graph and the time-dependent shortest path.

*3.1. Multimodal network*

Let $\mathrm{G} = (\mathrm{V}, \mathrm{E}, \mathrm{M})$ denotes a multimodal directed graph, where $\mathrm{M} = \{\mathrm{m}_1, \dots, \mathrm{m}_\mathrm{k}\}$ is a set of transport mode (e.g., train, bus, metro). An edge $\mathrm{e}_\mathrm{i} \in \mathrm{E}$ can be identified by $(\mathrm{v}_\mathrm{i}, \mathrm{v}_{\mathrm{i}'}, \mathrm{m}_\mathrm{i})$ where $\mathrm{v}_\mathrm{i}, \mathrm{v}_{\mathrm{i}'} \epsilon \mathrm{V} \wedge \mathrm{m}_\mathrm{i} \epsilon \mathrm{M}$. $\mathrm{e}_\mathrm{i}$ means the possibility of going from node $\mathrm{v}_\mathrm{i}$ to $\mathrm{v}_{\mathrm{i}'}$ by using transport mode $\mathrm{m}_\mathrm{i}$. A path $\wp = \{\mathrm{e}_1, \mathrm{e}_2, \dots, \mathrm{e}_\mathrm{k}\}$ is said to be multimodal if $\exists \mathrm{e}_\mathrm{i}, \mathrm{e}_\mathrm{j} \in$ A, $\mathrm{e}_\mathrm{i} = (\mathrm{v}_\mathrm{i}, \mathrm{v}_{\mathrm{i}'}, \mathrm{m}_\mathrm{i}) = \{\mathrm{v}_\mathrm{i}, \mathrm{v}_{\mathrm{i}'}\}_{mi}$ , $\mathrm{e}_\mathrm{j} = (\mathrm{v}_\mathrm{j}, \mathrm{v}_{\mathrm{j}'}, \mathrm{m}_\mathrm{j}) = \{\mathrm{v}_\mathrm{j}, \mathrm{v}_{\mathrm{j}'}\}_{mj}$ , $\mathrm{m}_\mathrm{i} \neq \mathrm{m}_\mathrm{j}, and\ i, j \leq k$. To get to his destination, a traveler in a multimodal transportation network commutes between more than one transport modes. He can drive his car to a metro parking then he can take a metro line …etc. The intersections of these modes are called Transfer nodes. It is a node where the traveler commutes from one mode to another under changing conditions. The representation of this transfer action is treated as one of the key issues in multimodal route planning problem[16]. Firstly, it is assumed in the following context that the transfer time between a stop and its transferable stop is defined as the sum of the walking time between these two stops, the waiting time to take the related transferable mode at the transferable stop and the stopping time of the related transferable mode at the transferable stop. Therefore, it is relevant to insert additional transfer arcs with a cost function $cost_{transfer}(\,)$.

*3.2. Time-dependent shortest path problem*

Shortest path algorithms have been extended to the time dependent problem (time taken on an edge depends on the time) with success. Two approaches are used: the time-dependent model and the time-expanded. The first keeps the existing topology but uses cost functions instead of constant costs. The second model splits the time in a finite number of time intervals and duplicates every node for every time interval. A general trend is to use the time dependent model on road networks and the time expanded in timetable based networks (like public transport). Therefore, we favor the time-expanded model due easier modeling, while the time dependent model has slightly better performances. The realistic version of the time-dependent model[17,18] consists of station nodes (transport station) and route nodes (station platforms connected by transfer arcs to station nodes). The construction of this model proceeds as follows. First, for every station a station node is inserted into the graph. Then the route nodes are inserted: A trip of a vehicle is defined as the sequence of stations it visits according to its elementary connections defined in the timetable. Trips consisting of the exact same sequence of stations are grouped into routes. For every station included in a route, a route node is inserted in the graph and connected by transfer arcs to the corresponding station node. Subsequent route nodes belonging to the same route are connected by travel arcs with time-dependent cost functions. Furthermore, Transfer times between platforms of the same station can be incorporated into the model by inserting additional transfer arcs between route nodes. We chose to define the cost function of an edge $\mathrm{e}_\mathrm{i}$ as a 6-tuple $f_{e_i} = (source, target, departure, arrival, cost, mode)$. The TDSP of a public transportation network can be solved efficiently when the arc cost function is non-negative and FIFO: $\tau \leq \tau' \Rightarrow \tau + f(\tau) \leq \tau' + f(\tau')$.

## 4. Proposed algorithm

The proposed algorithm is a target-oriented algorithm based on the concept of "closeness" to the target node as heuristic to drive the search toward its destination. It is principally based on computing a virtual path which is the Euclidean distance from the source node to the target node, then driving the search toward the connected nodes close to this virtual path by a parameter $d$. The idea of the proposed algorithm came from a logical and evident procedure of searching for the shortest cut to our destination which is obviously staying as close as possible to the destination and avoiding detours that could drive us away from it. As the known algorithms A* and ALT use the concept of lower bounds in triangular inequality to give the priority to nodes that are supposed to be closer to the target, our algorithm uses the virtual path as a conductor of the search space and the parameter d and $\Delta d$ (explained later in this section). This virtual path $(st)$ is considered as an ideal path from the source node $s$ to the target node $t$ and used as a reference path to conduct the search into a restricted space. The algorithm can navigate in backward in the graph as it can go forward, to make sure that it explores all possible paths until constructing the shortest path.

The parameter d defines the width of the search space and can be enlarged by the other parameter $\Delta d$. Let $G = (V, E, M, T)$ denotes the time-dependent multimodal directed graph or network where T is a set of travels in $G$. The edge $\mathrm{e} = \{\mathrm{v}, \mathrm{v}'\}_m$ has a time-dependent distance cost $c_e(t)$. Let $t, t'$ denote time in a discrete set $\{t_1, \dots, t_l\}$, the travel for the edge $e$ is defined as tuple $(t, t') \in T$ where $t$ is the departure time from node $v$ ,and $t'$ is the arrival time at node $v'$. The main idea behind the algorithm approach is to explore the nodes in a restricted search space defined by three parameters and calculated as presented below, knowing that there are other possible methods of calculation:

- Virtual path $(st)$: is the Euclidean distance from $s$ to $t$, represents the reference path
- Distance $d$: is the threshold distance of the search space, represents the mean distance value from all the vertices to the virtual path: $calculate\_threshold(\,)\ \{\, v_i \in V, d = \sum_1^n dist\big(v_i, (st)\big)/n \,\}$ (4.1)
- Distance $d_{max}$: is the maximum value of $d$, represents the distance value from the farthest vertex to $(st)$.
- Distance $\Delta d$: is the elementary step distance of the search space, represents the mean distance value from all the vertices to their neighbors' vertices:$calculate\_step(\,)\ \{\, v_i \in V,\ \Delta d_i = \sum_1^k x_i/k$
  $\Delta d = \sum_1^n \Delta d_i/n\}$ (Where $total_{neighbours}(v_i) = k$ and $x_i$ the distance from $v_i$ to its neighbor) (4.2)

Before elaborating the algorithm approach, we should mention that the virtual path presented as an Euclidean distance from $s$ to $t$ equal to D is a model to demonstrate our approach and we can adopt the same reasoning to other cost function like travel time. The logic of our algorithm remains applicable when we substitute $dist(w, (st))$ with $f(s, w) + f(w, t)$ which has to be less than $f(s, v) + f(v, t) = \tau$, where $f$ is the cost function and $\tau$ is the threshold value equivalent to $d$. $\tau$ can be calculated the same way as $d$ by means of $calculateThreshold(\,)$, equivalent in this case to the mean value of $f(s, v) + f(v, t)$ such that $v_i \in V, \tau =, \sum_1^n f(s, v_i) + f(v_i, t)/n$ (4.3).

Actually, time-dependence and multimodality aspects are not explicitly shown in this case of study, although the approach is designed to work with these restrictions. The proposed algorithm uses searching constraints to delimit the search space with a lower bound and an upper bound. The lower bound is defined by the virtual path which is the minimum cost function value from $s$ to $t$and the upper bound is defined by the parameter $d$ or in general $f(s, v) + f(v, t) = \tau$. The algorithm proceeds as follow:

- Compute the virtual path $distance(s, t) = D$ and the constraint parameters $d$ and $\Delta d$ ;
- Launch the search process restrained into the reduced search space $(2d, D)$ and start to construct the path with edges that satisfy the condition $\forall v_i \in V, dist(v_i, (st)) \leq d$ ;
- In each searching iteration and after selecting a candidate vertex $v$ inside the search space, apply the elementary step function $OneStepMMTDSP(v, t)$ on $v$. This function applies a multimodal time-dependent shortest path algorithm to find the next candidate vertex for the next iteration. It can be substituted with any algorithm designed to find the SP in a time-dependent multimodal network.
- Increase the parameter d by $\Delta d$ when no vertex satisfies the constraint above, in order to extend the search space and reiterate the search process to find other candidate vertices for the constructed path.
- In case if exploring edges in the search space ($2d_{max}$, D) doesn't deliver a candidate edge for the path, go backward on the graph by removing the candidate edges in the constructed path and launch the search process with an increased parameter $d + \Delta d$.

Constraint Based SP Algorithm

**Input:** $G = (V, E)$, s, t **Output:** $\wp$ath
$CBSP(u, d, \wp\text{ath}, Q)$ {
$d \leftarrow calculate\_threshold(G, s, t)$
$\Delta d \leftarrow caluclate\_step(G)$
$Q = \{v \in Neighbours(u) \mid dist(u, (st)) \leq d\}$
**While** $t \notin Q$ **Do**
  **If** $Q = \emptyset$ **Then**
    **If** $d < d_{max}$ **Then** $d \leftarrow d + \Delta\mathrm{d}$
    **Else** $w \leftarrow predecessor(u)$; $d \leftarrow d + \Delta\mathrm{d}$; $\wp$ath ← $\wp$ath \{u}
**Else** **For each** $v \in Q$ **Do** $NextV \leftarrow OneStepMMTDSP(v, t)$
  **For each** $w \in NextV$ **Do**
    **If** $dist(w, (st)) \leq d$ **Then** $Q \leftarrow Q \cup \{w\}$; addTopath(w) $Q \leftarrow Q \setminus \{v\}$)**Return** $\wp$ath }

*4.1. Algorithm possible scenarios*

We try to explain the approach adopted by our constrained shortest path algorithm by following the search phase for different cases. In the first case, the algorithm tries to go forward to the target exploring the neighbor nodes inside of the search space that will compose the resulted shortest path. The algorithm succeeds to find the nodes inside the search space (2d, D) so then constructs the shortest path. The application in this case is straight forward because of the satisfaction of the condition $dist(v_i,(st)) \leq d$ with $v_i$ the neighbor node connected to the candidate node. In the second case, the algorithm applies the distance constraint d for each visited node and its neighbors. When exploring the edges inside the search space doesn't deliver a complete path to the target $t$, we increase d by Δd until a next edge is found and continue the search until we reach our destination node. In the third case, when the algorithm has explored all the possible nodes in the search space and yet the shortest path couldn't be built, we go backward on the graph and start the search on a larger search space.

## 5. Experimental results

An important number of trials with randomly generated instances have been used as a test bed for verifying both the goodness of the obtained solutions and the corresponding CPU time. The CBSP algorithm in the implemented prototype is a variant of A* algorithm in a way that the algorithm use A* for the elementary step function$OneStepMMTDSP$. Therefore, CBSP is compared in the first place with A* to point out the notable reduction of computational time (fig.1). The comparison shows how CBSP reduces the CPU time efficiently in a dense network in comparison with A*. The table in fig.2 measures in percentage the time reduction. The algorithm searching approach is based on the parameters $d$ and $\Delta d$ calculated in (4.1) and (4.3). We noted that this method of calculating the parameters is one of many possible methods that could be used in the algorithm.

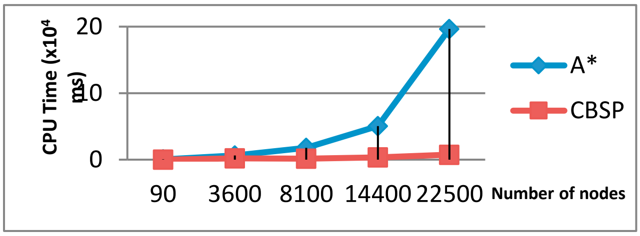


Figure 1. The comparison of A* and CBSP

| Number of nodes | 90 | 3600 | 8100 | 14400 | 22500 |
|---|---|---|---|---|---|
| Time reduction | 17% | 29% | 7.7% | 7% | 3.8% |

Figure 2. Reduction of CPU time

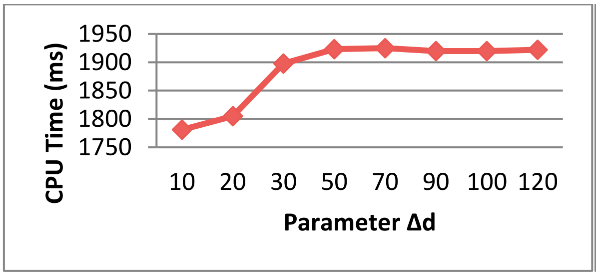


Figure 3. Effect of $\Delta d$ on CPU time (d fix)

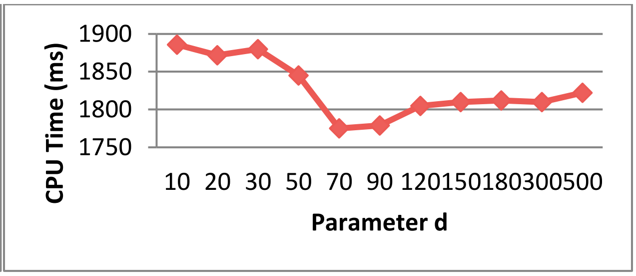


Figure 4. Effect of d on CPU time ($\Delta d$ fix)

However, the algorithm performance highly depends on the values of $d$ and $\Delta d$ to minimize the CPU time. So, we try to give different values to those parameters and calculate the corresponding execution time (fig.3 and fig.4). The performance of the algorithm is more significant in some values of $d$ and $\Delta d$ which includes the importance of the choice of the calculation method. If we chose a minimal value of $d$ which is the upper bound that conducts the search process, the CPU time becomes higher.

In the other hand, $\Delta d$ is an elementary step distance added to $d$, so the more it is minimal the better it is. The proposed algorithm was able to find the exact shortest path in reduced CPU time. However, an extensive computational effort devoted at validating the proposed algorithm is in progress.

## 6. Conclusion

Solving the multimodal route planning is a highly important issue to develop seamless advanced traveler information systems. For this purpose, we proposed a new approach for finding the time-dependent shortest path in multimodal network. In this work we present an overview of the existing solutions for shortest path problem in multimodal networks and a description of the problem to finally propose the algorithm with illustrative examples and point out the effectiveness of its searching process. In the future, the implementation of our proposed algorithm will be held in a distributed parallel architecture and evaluated by comparing the calculated computational time.

## Acknowledgment

This work was partially funded by the CNRST project in the priority areas of scientific research and technological development "Spatio-temporal data warehouse and strategic transport of dangerous goods»

## References

1. Nguyen, S. and Pallottino, S. “Equilibrium traffic assignment for large scale transit network.” European Journal of Operational Research, 37(2), p. 176-186, 1988.
2. Di Febbraro, A. and Sacone, S. “An on-line information system to balance traffic flows in urban areas.” Porceeding of the 36th IEEE conference on decision and control, 5, p. 4772-4773, 1995.
3. Lozano, A. and Storchi, G. “Shortest viable hyperpath in multimodal networks.” Transportation Research Part B 36, p. 853–874, 2002.
4. Pallottino, S. and Scutellà, M. G. “Shortest path algorithms in transportation models: classical and innovative aspects.” In Equilibrium and Advanced Transportation Modeling, Kluwer, p. 1245-281, 1998.
5. Ziliaskopoulos, A. and Wardell, W. “An intermodal optimum path algorithm for multimodal networks with dynamic arc travel times and switching delays.” Elsevier Science, European Journal Of Operational Research, 125, p. 486–502, 2000.
6. Dijkstra, E. W. “A note on two problems in connexion with graphs.” Numerische mathematik 1.1: 269-271. 1959.
7. Schultes, D. “Route Planning in Road Networks.” Ausgezeichnete Informatikdissertationen. 2008.
8. Cooke, K. and Halsey, E. “The Shortest Route through a Network with Time- Dependent Intermodal Transit Times.” Journal of Mathematical Analysis and Applications, (14):493{498, 1966.
9. Delling, D. and Wagner, D. “Time-dependent route planning.” Robust and Online Large-Scale Optimization. Springer Berlin Heidelberg (p. 207-230), 2009.
10. Delling, D. “Engineering and augmenting route planning algorithms”. Diss. Karlsruhe Institute of Technology, 2009.
11. Delling, D., Pajor, T., and Wagner, D. “Engineering time-expanded graphs for faster timetable information.” Robust and Online Large-Scale Optimization. Springer Berlin Heidelberg, 182-206. 2009.
12. Bast, H. et al. “Route planning in transportation networks.” Technical Report MSR-TR-2014-4. Microsoft Research, Microsoft Co., 2014.
13. Bauer, R., Delling, D. and Wagner, D. “Experimental study of speed up techniques for timetable information systems.” Networks 57.1 38-52. 2011.
14. Abeysundara, S., Giritharan, B., and Kodithuwakku, S. “A genetic algorithm approach to solve the shortest path problem for road maps.” Proceedings of the International Conference on Information and Automation. 2005.
15. Yu, H., and Lu, F. “A multi-modal route planning approach with an improved genetic algorithm.” Advances in Geo-Spatial Information Science: 193. 2012.
16. Ayed, H., Galvez-Fernandez, C., Habbas, Z., & Khadraoui, D. “Solving Time-Dependent Multimodal Transport Problems Using a Transfer Graph Model.” Computers & Industrial Engineering, 2010.
17. Pyrga, E., Schulz, F., Wagner, D. and Zaroliagis, C. “Efficient models for timetable information in public transportation systems.” Journal of Experimental Algorithmics (JEA), 12: Article 2.4, 2007.
18. Schulz, F. “Timetable Information and Shortest Paths.” Phd thesis, University of Karlsruhe (TH), 2005.