# Design and implementation issues of a time-dependent shortest path algorithm for multimodal transportation network

Abdelfettah IDRI[1*], Mariyem OUKARFI[2], Azedine BOULMAKOUL[2] and Karine ZEITOUNI[3]

[1]LIM Lab. IOS, National School of business and Management, Casablanca, Morocco
abdelfattah.id@gmail.com
[2]LIM Lab. IOS, Computer Sciences Department, Faculty of Sciences and Technology Mohammedia, Mohammedia, Morocco
oukarfi.mariyem@gmail.com
azedine.boulmakoul@yahoo.fr
[3]David Lab. University of Versailles Saint-Quentin-en-Yvelines, Versailles, France
karine.Zeitouni@uvsq.fr

**Abstract.** In recent years the structure of transportation networks has become more complex as a result of the fast development of the social and economical infrastructure. When dealing with time dependent multimodal context, transport planning represents a fundamental problem and more specifically for hazardous materials. Several approaches have been proposed that strive to reduce the financial costs, travel time and vehicle operating costs. Our work tends to focus on potential improvements related to resolve the shortest path problem. This paper handles the design and implementation issues of our monolithic approach solving the time dependent multimodal transportation problem aimed at calculating the shortest path from a source node to a destination node. The algorithm on which relies our approach is target oriented and focuses basically on the reduction of the search space by considering a virtual path from the source to the destination as well as a user defined constraint defining a dynamically extendable corridor-like restricted search zone that can be applied on either the Euclidian distance or the travel cost function . The specification details of the algorithm have been already presented in our previous work and will be out the scope of this paper. Especially, this work highlights the aspects related to the different dimensions of the search space, namely the time, the mode and the constraint parameter.

**Keywords.** Time dependent Multimodal transport network, search dimension, constraint based shortest path.

## 1 Introduction

Travelers are often faced to a common route planning problem known in the public transport system as the scheduling decision. It involves the use of different public transport means. The complexity relates in general to the specific schedule of each transport service, the connectivity level of the nodes defining the public transport network and the density of this later. This problem is formulated as the determination of the shortest path between the source and the destination regarding the traveler preferences in terms of travel costs and the transport infrastructure requirements.

A transport network is called multimodal when at least two different means of transport are required to perform a travel between a source and a destination. To find the optimal route manually is not an easy task regarding the constraints mentioned above that's why there is a need to compute the shortest path in a dynamic environment.

## 2 Related works

Several optimizations approaches have been proposed to improve the calculation of the shortest path in a time dependent multimodal transport network varying from classical solution relying on speed up techniques Bast et al. [1], to solutions introducing the time dependency component Cooke and Halsey [2], Pyrga et al. [3] and Bakalov et al. [4]. More developed approaches were introduced to extend a single mode to multimodal transport network Schultes et al. [5] and Pajor et al. [6]. Other relevant solutions relies on the graph techniques like the concept of the hypergraphs Bielli et al. [7], the transfer graph Ayed et al. [8] and the hierarchical graph Zhang et al. [9].

In this paper, we will focus on the design and implementation aspects based specifically on our previous work that expresses a constraint based shortest path algorithm in a time dependent multimodal context Idri et al. [10]. This algorithm assumes a straightforward virtual path from the source to the destination and drives the search process in such a way that only the nodes within the corridor defined by a pre-calculated constraint D and the virtual path will be considered for the final solution. When the algorithm needs extra nodes to converge to a solution, the search space will be expanded progressively by increasing the constraint D until a solution is found if it exists.

## 3 Definitions

In the following, a transport network is considered as a direct graph and is denoted as $G$ = (V, E, M, T), where V = $\{v_1,\dots,v_k\}$ is a set of vertices, E = $\{e_1,\dots,e_m\}$ is a set of edges, M = $\{m_1,\dots,m_k\}$ is a set of modes and T = $\{t_1,\dots,t_n\}$ is a set of travels. Each travel $t_j$ is represented with a couple $(t_{jd}, t_{ja})$ specifying the departure and the arrival time. In a time-dependent multimodal context, an edge $e_l$ can be defined as a tuple $(v_i, v_j, m_k)$ expressing that there is a connection between node $e_i$ and node $e_j$ using mode $m_k$. To highlight time-dependency, an edge is mapped to a timetable that is defined as a set of travels involving all the possibilities of traveling from node $e_i$ to node $e_j$ using mode $m_k$. This defines a cost function that is applied to the edge $e_l$ at instant t $C_{el}(t)$. Note that multiple modes can be applied to the same edge. A path is then a set of connected edges from the start node to the target node P = $\{e_s,\dots, e_t\}$.

**Table 1.** Sample multimodal transport network

| *Mode 1* | | | *Mode 2* | | |
|---|---|---|---|---|---|
| Edge | Timetable | Cost function | Edge | Timetable | Cost function |
| a → e | 1 → 3 | 2 | e → d | 1 → 4 | 7 |
| | 3 → 4 | 4 | | 12 → 15 | 3 |
| e → c | 3 → 5 | 1 | a → f | 4 → 6 | 4 |
| | 4 → 8 | 1 | | 7 → 9 | 1 |
| c → b | 6 → 8 | 5 | f → c | 6 → 7 | 1 |
| | 7 → 10 | 2 | | 8 → 9 | 5 |
| g → a | 2 → 3 | 7 | g → a | 3 → 9 | 1 |
| | 9 → 11 | 6 | | 8 → 11 | 10 |
| b → g | 9 → 10 | 1 | b → g | 10 → 11 | 5 |
| | 10 → 11 | 1 | | 13 → 15 | 8 |

A sample transport network is given in Table 1. This network contains two modes and a set of edges related to their timetables and cost function which is considered as the travel time in our case. But it can represent the financial cost as well, or any other preferred variable and the same rules remain valid. Whenever we need to demonstrate specific aspects, we will refer to this sample network.

## 4 Constraint based shortest path algorithm

This section describes formally our approach by specifying the algorithm and its input parameters.

Figure 1 shows a recursive version of the constraint based shortest path algorithm (CBSPA) as introduced in our previous work [10]. When the algorithm is executed, it starts with a restricted search space defined by the user-defined constraint d which defines the threshold distance of the search space: the distance of a given node to the virtual path defined between the source and the target node. The constraint d is increased with a step parameter Δd whenever no nodes can be captured within a given iteration until the overall maximal distance Dmax is reached. The function OneStepMMTDSP(v,t) generates the next iteration candidates based on the timetable of the vertex v assumed $V_s$ and $V_t$ represent the start and the target nodes. The algorithm works on a time-dependent multimodal network G(V,E,M,T) as defined above.

```
Algorithm CBSPA (u, d, path ,Q , t)
  Output: shortest path
  // (VsVt): virtual path which is calculated based
on the coordinates (Euclidean case: straight line
between Vs and Vt) or the minimal/mean value of the
cost function (in our case, the travel time)
  // Q: the set of the neighbors of u satisfying
the constraint control, a parameter needed to for-
ward the intermediate results
  // t: start time; u: start node
  // d = ∑_1^n dist(v_i,(VsVt))/n : represents the mean dis-
tance(cost) from all nodes to the virtual path
  // Dmax: represent the maximum value of d, that's
the distance (cost) to the farthest node of the
network
  If u = Vt then
    Return path
  Else
   Let Q = {v ∈ Neighbors(u)/ dist(v,(VsVt)) ≤ d}
    In
    if Q = Ø then
      If d < Dmax then
        CBSPA (u, d+Δd ,path,Q,t)
      Else
        Let w = predecessor(u) in
        CBSPA(w,d+Δd, path\{u},Q,t)
    Else
      Let Qnew = ∐_{v∈Q} OneStepMMTDSP(v,t) in
      Let Q = {v ∈ Qnew/dist(v,(VsVt))<d} in
      ∀ v ∈ Q, CBSPA(v,d,path ∪ {v},Q\{v},t)
```

**Fig. 1.** CBSPA Algorithm

# 5 Search process

In this section, we present the details of the search process based on the different search dimensions expressed in terms of abstraction layers. It also handles the building process of the complete solution as well as that of the shortest path.

## 5.1 Global view

The search process is target oriented and is driven by three dimensions: the mode, the time and the user-defined constraint d. Each time we explore the possibilities of a given edge $e_{ij}$ connecting the nodes $v_i$ and $v_j$, we deal in fact with a three dimensional search space as illustrated in Figure 2.

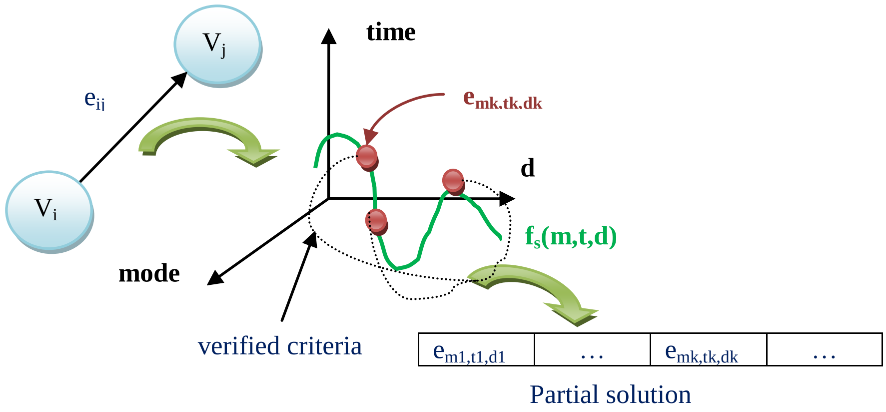



**Fig. 2.** Search dimensions

The function $f_s$ generates instances of the edge $e_{ij}$ from which some will belong the partial solution. Note that at the end of the search process, as the constraint d is increased gradually during the search process, the last reached value will be taken into account and will substitute the intermediate values seen that it is the upper bound value of the verified constraints.

**Remark .** Some criteria might be applied regarding one or more dimensions like a mode is only available up to a departure time t.

## 5.2 Time dependency layer

In a time dependent context, each edge is assigned a timetable indicating the different departure and arrival times relatively to this edge. An edge scenario which is a departure-arrival case derived from the timetable, can be considered then as an instance of the original edge. Therefore, we will obtain at the end a set of edge instances equivalent to the set of the possible edge scenarios. Following this description, if an edge may be viewed as an abstraction, then to process it in this search space, we have to handle each one of its instances but then for every single mode and constraint as shown in Figure 3-(a).

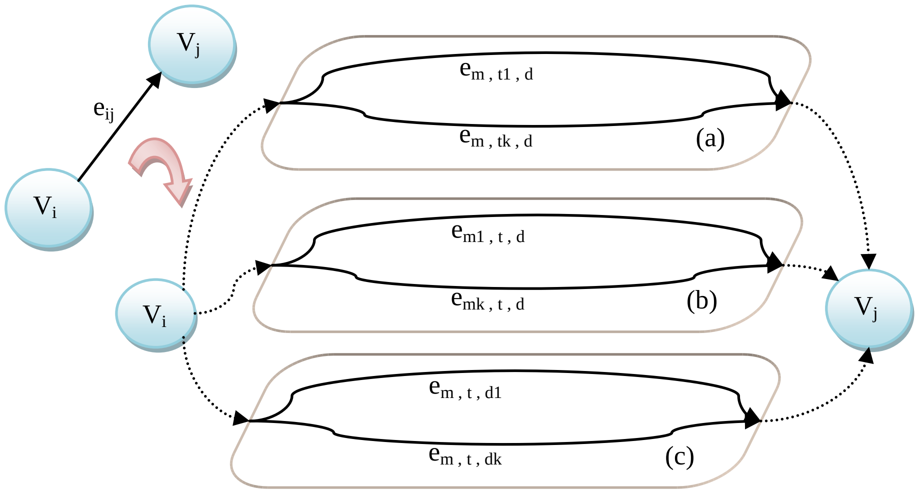



**Fig. 3.** Search layers: (a) time layer (b) multimodality layer (c) constraint layer

### 5.3 Multimodality layer

The same way, a mode is considered as an abstraction having a set of instances represented by the different modes connecting the start and the end nodes of the edge, see Figure 3-(b).

### 5.4 Constraint layer

As for the constraint d (Figure 3-(c)), its instances are generated following the needs of the search process and the size of instances set is unknown in advance.

### 5.5 Global solution

The current version of our algorithm returns as result the first found shortest path if it exists. One can easily alter this algorithm to include all possible shortest paths. The complete search space based on the above reasoning is nothing more than the combinatorial superposition of the three mentioned layers.

### 5.6 Building process of the complete solution

The search process targets the first available solution and that's why we adopted in our algorithm the DFS technique (Depth First Search) which relies on the partial/complete solution approach: the solution is build up progressively from its components mentioned above. Each element is defined by a three dimension edge instance $e_{mi,\ ti\ ,\ di}$ as seen before. Figure 4 describes the building process as the algorithm explores the search space.

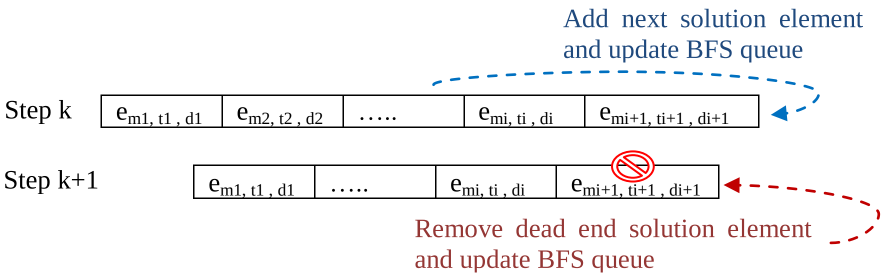



**Fig. 4.** Building process of the solution

To implement this concept, we used a backtracking mechanism, but then we were faced to the optimization process while reducing the search space by the elimination of the visited edge instances in a search scenario.

Again, to mark and unmark an edge instance is an action that has to be performed in the three dimensional space and this is a complex task: a visited edge means in our context a visited three dimensional instance issued from the combined layers described earlier. It is clear that in neither way a whole physical edge can be marked as visited or unvisited like in the classical approach. This leads us to assign the management of this optimization issue to a whole software component accordingly to the backtracking search process.

## 5.7 Building the shortest path

The building process of the solution is based on the partial/complete solution technique that needs a container. In our context it is simply a temporary queue that is build up of solution component similarly to an edge instance satisfying the search criteria: the departure time, the available mode and the constraint d. Such an instance is pushed in the queue to be handled following the algorithm logic. The next iteration, the edge instance at the top of the BFS queue is substituted with its successors verifying the search criteria.

The search process stops when it reaches the end node or when all possibilities are tried out. If a solution is found, it is stored in the temporary queue with some residues edge instances that couldn't be tried out because we are interested in the first found solution. To build up the shortest path we need to reconstruct the path backward starting with the end edge instance. Then the next predecessor is the instance edge having as end node the start node of the end edge instance and an arrival time less or equal to the departure time of the end edge instance and so forth until the start edge instance is reached. Overall, it should be noted that during this process the global cost of the shortest path can be calculated by totaling its elementary edge costs.

## 6 Class diagram of the proposed solution

In this section, a class diagram is given to show how the solution can be implemented based on the concepts introduced earlier.

Figure 5 exposes a class diagram model of our proposal based on the fundamental classes participating in the building process of the solution.

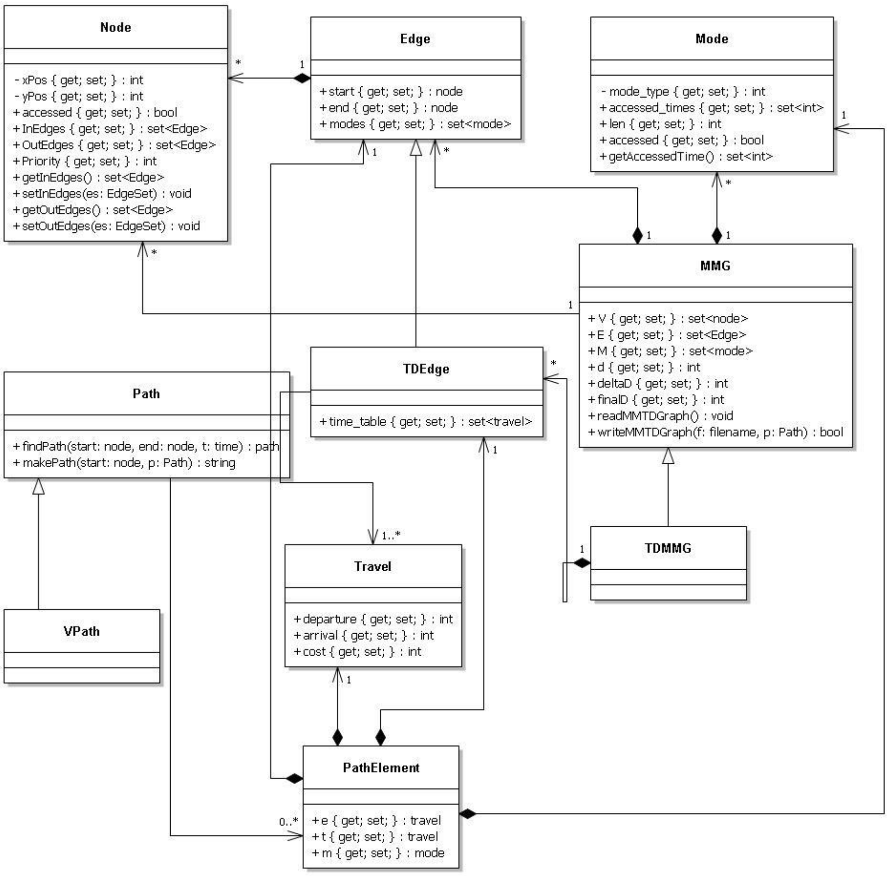


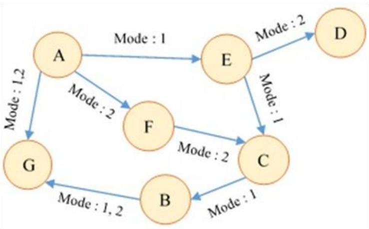

**Fig. 5.** MMTDSP Class diagram

This model supports either classical multimodal transport graphs based on the Euclidian distance (MMG) or time-dependent multimodal graphs that manipulate timetables (TDMMG). In this context, we are using the second model. The time dimension is captured within the Travel class and a path is modeled as a list of PathElement which is composed of travel (dimension time), mode and Edge (or TDEdge). The virtual path class VPath serves as a reference container of virtual paths.

# 7 Example

This section explains the whole process from the specification step to the generation of the shortest path by mean of an example based on the same data given in Table 1.

From the point of view of the concepts handled in the previous sections, we will focus on the performance of our implementation techniques instead of presenting the profiling and the benchmarking related to the business aspects of the approach which were evaluated in our previous work for both the monolithic and the distributed versions. We adopted in this experimentation a cost function defined as travel time and we measured then the behavior of the constraint fluctuation in terms of the necessary iteration to converge to a solution. Also we give hereafter in Figure 6 a result model based on the example of Table 1 to show how we performed our tests. We calculate the proper travel cost and the waiting delay is ignored in these tests. The label “Final D” is the final value of the constraint reached to find the shortest path. In our implementation, the nodes are expressed in numbers respecting in this example the alphabetical order: node A has number 1 and so forth.




Path to find :[ [ Node id: 1 ]======>[ Node id: 2 ] ]<br>
[<br>
[ start node: [ Node id: 1 ]]<br>
[ end node: [ Node id: 5 ]]<br>
[ travel : 1 ===> 3 ===> 2]<br>
[ mode : 1]<br>
[ start node: [ Node id: 5 ]]<br>
[ end node: [ Node id: 3 ]]<br>
[ travel : 3 ===> 5 ===> 1]<br>
[ mode : 1]<br>
[ start node: [ Node id: 3 ]]<br>
[ end node: [ Node id: 2 ]]<br>
[ travel : 6 ===> 8 ===> 5]<br>
[ mode : 1]<br>
[ Total cost : 8]<br>
]<br>
time :0 s<br>
Final D: 10

Path to find :[ [ Node id: 3 ]======>[ Node id: 7 ] ]<br>
[<br>
[ start node: [ Node id: 3 ]]<br>
[ end node: [ Node id: 2 ]]<br>
[ travel : 6 ===> 8 ===> 5]<br>
[ mode : 1]<br>
[ start node: [ Node id: 2 ]]<br>
[ end node: [ Node id: 7 ]]<br>
[ travel : 9 ===> 10 ===> 1]<br>
[ mode : 1]<br>
[ Total cost : 6]<br>
]<br>
time :0 s<br>
Final D: 5

**Fig. 6.** CBSPA result models

Figure 7 shows how the search process behaves regarding the constraint parameter d and the network density expressed in the total nodes number. The final constraint value reached within a search process reflects the iteration number performed to achieve the next node. It is clear that the iteration number refers to the level of the search space reduction. That’s, as long as the maximal value of the constraint parame-

AMI Routerter is not reached; this means the search space is restricted accordingly to the number of iterations. Figure 7 shows also that there is an impact of the network size on the iteration number at least with the samples used in these tests.

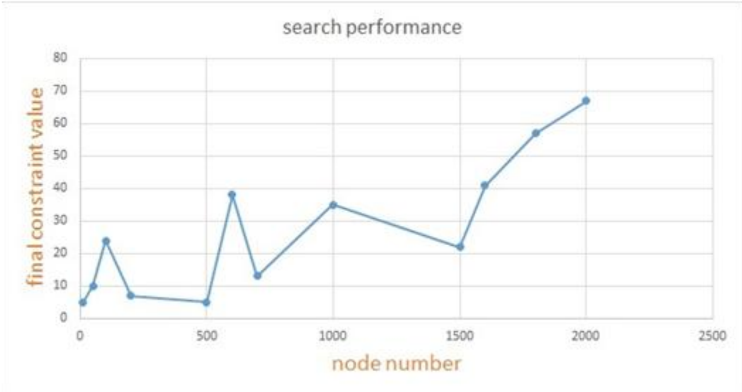



**Fig. 7.** Impact of the network density on the constraint parameter iterations

## 8 Large scale networks

Our approach to deal with large scale networks is based on parallel distributed architecture that relies on a Manager/Agent model. This model fits well with our needs as it is scalable and each agent runs in an independent way (parallel). Multiple agents cooperate to deliver partial solutions which are gathered by the manager that builds up the complete solution. Figure 8 exposes a distributed architecture relying on a CORBA framework as it offers object-oriented facilities and distributed event management that supports Asynchronous Method Invocation (AMI). A model based on Map/Reduce architecture may also give similar results.

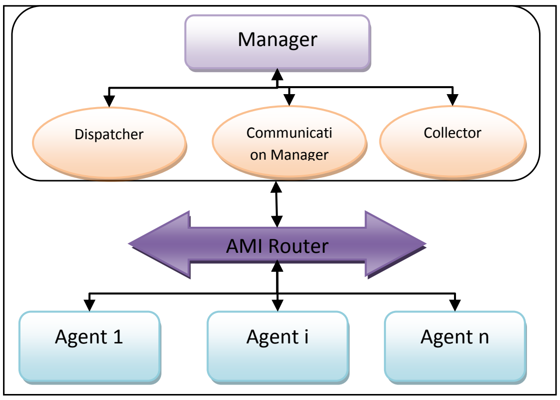



**Fig. 8.** CORBA Architecture

Each agent is assigned the task to compute elementary paths starting from a given intermediate node $V_i$ (initially the start node). These paths are represented by the set of the next nodes directly connected to $V_i$ and satisfying the algorithm constraints. The manager is responsible for dispatching the tasks, collecting the elementary results, building the complete solution as well as covering the communication issues with the agents.

## 9 Conclusion

In this paper we treated some design and implementation aspects regarding our approach dealing with the time-dependant multimodal transport problem expressed as finding the shortest path algorithm. Especially, we focused on the design techniques to solve the problem in the context of the different search dimensions including the user defined constraint that influences the search process and the final results. To address the big data issue in this context, we adopted a parallel distributed architecture that guarantees the scalability and improves the performance of the algorithm.

We intend in our future work to investigate deeply the behavior of existing algorithms when embedding this approach within these algorithms in a parallel distributed architecture.

## References


1. Bast, H. et al., “Route planning in transportation networks.” Technical Report MSR-TR-2014-4. Microsoft Research, Microsoft Co. (2009).
2. Cooke, K. & Halsey, E., “The Shortest Route through a Network with Time-Dependent Intermodal Transit Times”. Journal of Mathematical Analysis and Applications, 14, 493-498 (1966).
3. Pyrga, E., Schulz, F., Wagner, D., &Zaroliagis, C. 2008.“Efficient models for timetable information in public transportation systems”. Journal of Experimental Algorithmics (JEA), 12, 2-4, (2008).
4. Bakalov, P., Hoel, E., &Heng, W. L., Time dependent transportation network models. In Data Engineering (ICDE), 31st International Conference on (pp. 1364-1375). IEEE (2015).
5. Schultes, D., “Route planning in road networks”. Ph.D Thesis of Karlsruhe Institute of Technology (2008).
6. Pajor, T., “Multi-modal route planning”.Master thesis of Karlsruhe Institute of Technology (2009).
7. Bielli, M., Boulmakoul, A., and Mouncif, H., “Object modeling and path computation for multimodal travel systems”. European Journal of Operational Researchpp. 175, 1705–1730 (2006).
8. Ayed, H., &Khadraoui, D., “Transfer graph approach for multimodal transport problems”.Second International Conference MCO, Metz, France – Luxembourg (2008).
9. Zhang, L., Yang, H., Wu, D., & Wang, D., “Solving a discrete multimodal transportation network design problem”. Transportation Research Part C: Emerging Technologies, 49, 73-86 (2014).
10. Idri, A., Oukarfi, M., Boulmakoul, A., Zeitouni, K. &Masri, A., “A new time-dependent shortest path algorithm for multimodal transportation network”.Procedia Computer Science.volume 109, Pages 692–697 (2017).