Finding the K-best paths in evacuation network

H. Alaeddine K. Serrhini M. Maïzia E. Néron

18 novembre 2014

Abstract

The problem of K-shortest paths between two nodes in network has been extensively studied, offering several efficient algorithms for different applications. These algorithms, so-called ranking methods, aim to compute the first shortest path then the second one etc. It's well known that the computation in these methods is based only on one criterion which is generally the distance or the time. Often a single criterion is not sufficient in some real-world problems (road network, internet, etc.) where one or several supplementary criteria (such as cost, capacity, security, etc.) must also be taken into consideration. This multi-criteria optimization, so-called labeling algorithm, aims to compute a Pareto front that represents a set of non-dominated pathways. However it is very difficult to make a decision on which K−best paths to choose among this set, and especially when this sub-set of paths serves as a data input for a sensitive problem such as the evacuation. In the event of this, buildings located in dangerous area must leave to a safety zone via an established evacuation network. We present in this paper a ranking methods enabling the computation of a set of K-best paths between an origin-destination pair. The free-flow travel time, the capacity and the number of lanes of roads are the criteria which are considered when computing the K-best paths among buildings and shelters associated. The strategy applied here and the optimization criteria used in these methods aim both to convert the mono-criteria aspect to a multi-criteria one combining between time and traffic.

Keywords accessibility, ranking path, evacuation, multi-criteria optimization, K-best paths

1 Introduction

Computing the K-shortest paths between a given pair of nodes in a graph was the subject of many research works in different disciplines such as computer science, operations research, engineering, etc. Their applications include network and electrical routing, transportation, etc. [12].

The ranking K-shortest path is a more generalized problem of the basic problem of finding the shortest path between an origin-destination pair [13]. It aims to compute at each iteration $k \in \{1,\ldots,K\}$ the k-th shortest path that has axiomatically a cost (time or distance) higher or equals to that of the $(k-1)$th path. For different reasons the computation of the second (third, etc.) shortest path is important. For example, in emergency situations the first shortest path between an origin-destination cannot be the only escape route for the concerned people. Moreover in route planner web applications, a computed set of available itineraries between places provides flexibility for applications and gives users the choice to select their paths [21]. It should be noted that those applications are generally not based on ranking algorithms but on labeling ones (label correcting, label setting) that enable to compute a Pareto front representing all non-dominated paths according to several criteria such as distance, security, etc. [22][24]. Chen et al. [5] developed a polynomial algorithm for finding the first K minimum cost simple paths in a time-schedule network where each node has a list of departure times. This type of network is interesting for modeling a transportation problem where, for example, a departure time is the moment when a train leaves from a station to another one. The travel time on arcs added to the waiting time on nodes are the criterion on which an algorithm of computation of shortest path is based.

Hershberger and al. [11] presented a new algorithm for ranking the K-shortest paths in a directed graph. Their algorithm is based on a replacement paths algorithm proposed by Hershberger and Suri [12]. Let $p = \{a_1,\ldots,a_r\}$ be a shortest path from an origin s to a destination t. The replacement paths problem aims to find at each iteration $1 \le i \le r$ the shortest path from s to t that does not pass through the arc a_i. Eppstein [9] addressed the problem of K shortest paths with minimum total length using heap-ordered tree and where cycles of repeated nodes are allowed. The method is to compute the shortest path tree and then to build a graph representing all possible deviations from the shortest path. According to Jimenez and Marzal [16], the experimental results showed that the time required to build this graph is considerable. Thereby they proposed a modified version of Eppstein's algorithm where only some deviations from the shortest path are built. In other words, those are the promising deviations for the selection of the K shortest paths. Hamed [10] developed an approach by applying a genetic algorithm for computing the K-shortest paths based on a real world system (links bandwidth of the network). Repeated nodes are also allowed in that paper. A comparative study of K−shortest paths algorithms was studied by [4], where four methods [27, 18, 17, 13] were selected for more detailed study from over seventy papers focusing on this problem. AlNasr et al. [2] developed an approach combining a dynamic programming algorithm with a K-shortest path algorithm to rank the topologies of the protein secondary structure elements detected in CryoEM volume maps. To reduce congestion problem on streets in modern cities by encouraging private-vehicle drivers to use public transportation, Wu and Hartley [26] presented a

K-Shortest paths algorithms to accommodate user preference in the optimization of public transport travel. Jimenez and Marzal [15] addressed the computation of the K− shortest paths between a source and a sink by developing a recursive enumeration algorithm to solve a set of equations generalizing the equations of Bellman [3] for the single shortest path problem. Stepanov et al. [25] have considered the time criterion to compute the K-best paths for the evacuation problem.

Beside the ranking algorithms listed below that address the problem of K− shortest paths on a mono-criterion basis, several authors addressed this problem on a multi-criteria basis. Lee et al. [19] have developed an algorithm that calculates the K-best paths with a minimum total cost and a minimum number of common nodes. Pangilinan and Janssens [23] have addressed the problem of multi-objectives path aiming to find all the effective ways (non-dominated or Pareto-optimal) from a source node to a destination node with multiple objectives. Sauvanet et al. [24] also addressed the multi-objective paths problem for the design and the implementation of a mobility center for cyclists, where several criteria (distance, insecurity, stress, etc..) are taken into account [22].

We address in this paper the construction of a specific transportation network dedicated for the evacuation of population exposed to natural disasters. The behavior of the two raking algorithms presented in this paper based on the criterion (free-flow travel time or free-flow travel time divided by number of lanes of roads) enables to ensure both the minimization of travel time and the maximization of the capacity of K− best paths computed. In other words these algorithms aim to convert the mono-criteria aspect to a muti-criteria one combining between travel time and traffic. The selection of K-best paths between buildings and shelters is crucial here. The incidence list is used to represent the transport network, the Dijkstra's algorithm [7] is implemented here with a Fibonacci heap [8] to accelerate the calculations. It should be noted that Dijkstra's algorithm can be replaced by any other algorithm allowing to compute the shortest path between a source node and a destination node.

2 Ranking methods

Let us consider $\mathbf{B} = \{\mathscr{B}_1, \dots, \mathscr{B}_n\}$ a set of n buildings to be evacuated towards a set of m shelters $\mathbf{S} = \{\mathscr{S}_1, \dots, \mathscr{S}_m\}$. We assume that one or more shelters are assigned to each building, and we focus in this paper on constructing a transportation evacuation network $\mathscr{G} = (\mathscr{N}, \mathscr{A})$ by computing the K-best paths between each building and each safety point associated.

2.1 Method K-min

The first algorithm developed aims to compute all the shortest paths in terms of time between each building and each safe point associated. At each iteration, the algorithm determines the shortest path and then subtracts the capacity of this path from that of all its arcs. Finally, the iteration is terminated by removing from graph the arc(s) with zero capacity.

Thus, this method enables to calculate only the useful paths for each building. Indeed, from the same building, two paths having at least one common arc with a minimum capacity are of little interest.

The pseudo-algorithm 1 describes the K−min method and a sample transport graph (figure 1) shows the behavior of this algorithm.

Algorithm 1 K-min

```
procedure FIND THE K−BEST PATHS
    Input : Network, Buildings, Shelters
    for each building do
        for each safety point associated do
            repeat
                Compute the shortest path between the building and the safety point
                Reduce the capacity of arcs belonging to this path
                    ▷ New capacity of arc = current capacity of arc - capacity of minimum arc in terms of capacity
                Deactivate all arcs with zero capacity
            until there is no path
            Reactivate the arcs and reset their capacities
        end for
    end for
end procedure
```

2.2 Method K-all

The second algorithm K-all was proposed by Martins [20] known under the name of removing path method. This algorithm also seeks to calculate all shortest paths in terms of time between each building and each shelter associated. However, the strategy followed here is somewhat different since at each iteration of seeking the shortest path (between a building and a safety point), the method does not take into account the arcs belonging to the shortest paths computed in previous iterations (see algorithm 2 and figure 2). The paths thus calculated are pairwise disjoint.

Algorithm 2 K-all

```
procedure FIND THE K−BEST PATHS
    Input : Network, Buildings, Shelters
    for each building do
        for each safety point associated do
            repeat
                Compute the shortest path between the building and the safety point
                Deactivate all arcs belonging to this path
            until there is no path
            Reactivate the arcs
        end for
    end for
end procedure
```

2.3 Selection of K−best paths

The K−min algorithm can provide a large number of paths between each origin-destination pair. Three complementary

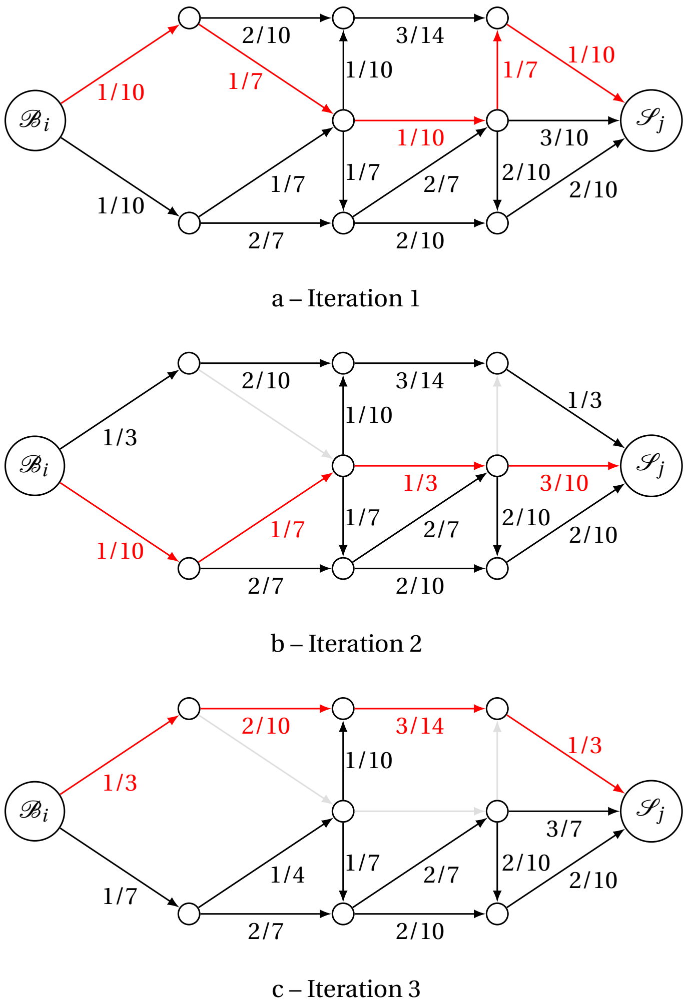


FIGURE 1 – Behavior of the algorithm K−min. weight on arcs : free-flow travel time/capacity

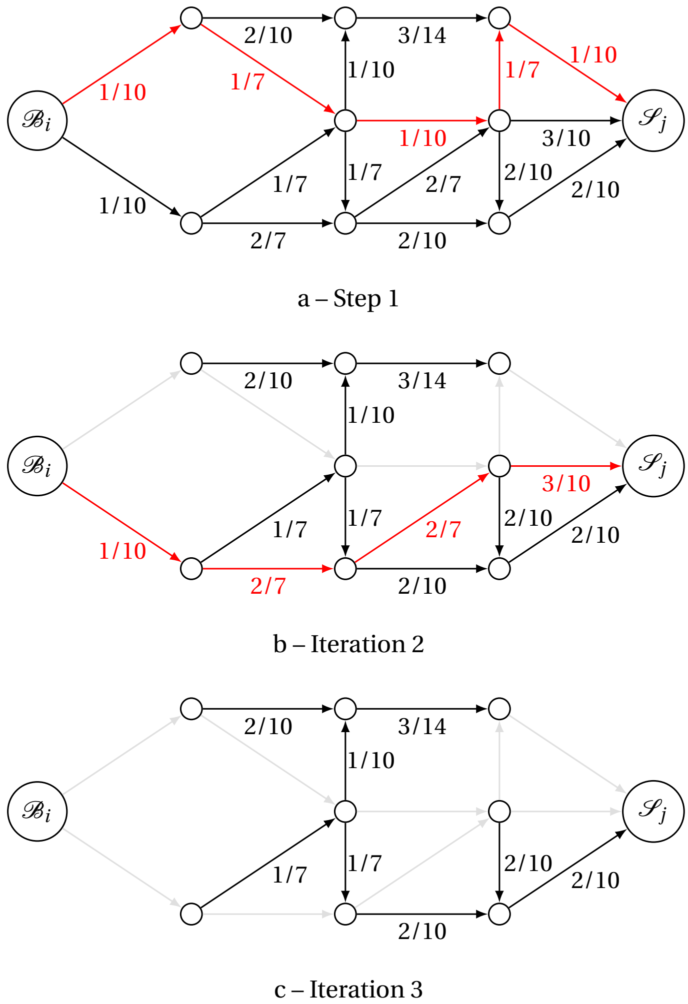


FIGURE 2 – Behavior of the algorithm K− all

strategies can be used to reduce the number of computed paths. The first strategy is based on the setting of a tolerance factor $\phi \geq 1$ [14] which purposes to consider only the paths whose length is smaller than that of the first shortest path multiplied by ϕ. However, difficulty here lies in determining the value of the factor ϕ.

The second strategy is to consider only the non-dominated paths by constructing a Pareto front [6]. As the K-min method calculates at each iteration the shortest path, the path computed at the current iteration has a length greater than or equal to that calculated in a previous iteration. Thus, according to this logic, the path determined in the previous iteration dominates the calculated path if and only if the capacity of the latter is less than or equal (if their lengths are not equal) to that of the previous.

Finally, the third and final strategy is to select only paths that minimize not the common nodes [19] but common arcs because of the existence of multiple arcs between nodes.

The weight "time" used here to compute the shortest path is an additive criterion, while the capacity of arcs can not be considered as a cost function by the two previous algorithms since it is a maximization criterion.

A criterion which can combine the two criteria (time and capacity) is the ratio of free-flow travel time on arc to the number of its lanes. Therefore, determining the shortest path is based on the travel time influenced by the capacity.

The different methods developed above (K-min, K-all, criteria : time and time weighted by capacity), result in the production of K-best paths between an origin-destination pair.

Under the evacuation model STOM [1], we have selected the first 3 shortest paths (K = 3) for each origin-destination pair computed by K−min method based on time divided by number of lanes criterion. A comparison test between K−min and K−all methods according to the criteria described above will be performed in the next and last section.

It should be noted here that it is possible to deactivate some arcs or routes that will be reserved for the evacuation of specific major issues (nuclear infrastructure, hospitals, retirement homes, etc..).

1. A Spatio-Temporal Optimization Model for the evacuation of population exposed to natural disasters. This model is a part of the ACCELL (évaluation de l'ACCessibilité d'Enjeux Localisés en situation d'inondation sur le bassin de la Loire) project granted by la Région Centre and the ERDF (European Regional Development Fund)

3 Applications and results

The study site on which the model is applied and tested is the valley of Tours [2] [1]. The elaborated evacuation network corresponds to the 3-best paths between buildings and shelters and is characterized by : 13446 nodes, 22676 arcs, 9825 buildings grouped in 2753 buildings [3] and 2 safety points.

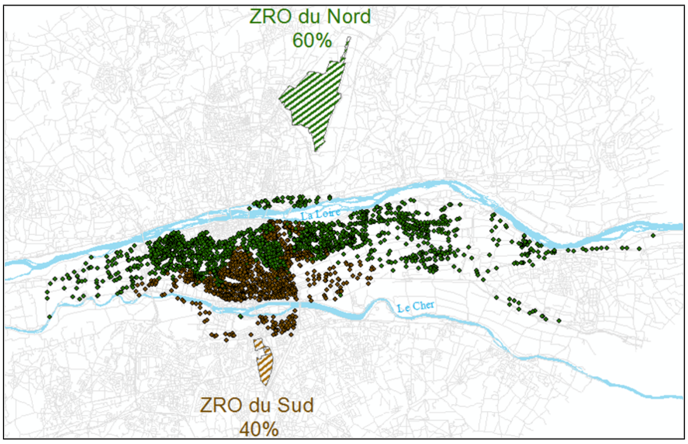
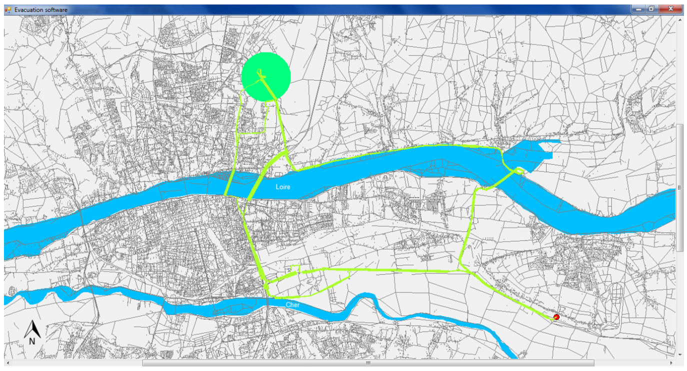


FIGURE 3 – The valley of Tours, FR 37

This section aims to examine the behavior of the two algorithms K−min and K−all on a real graph : valley of Tours (France, 37). The two criteria that come into play in the calculation are the free-flow travel time on arc and this time divided by the number of lanes of arc.

(a) K−Min : time criterion



(b) K−all : time criterion

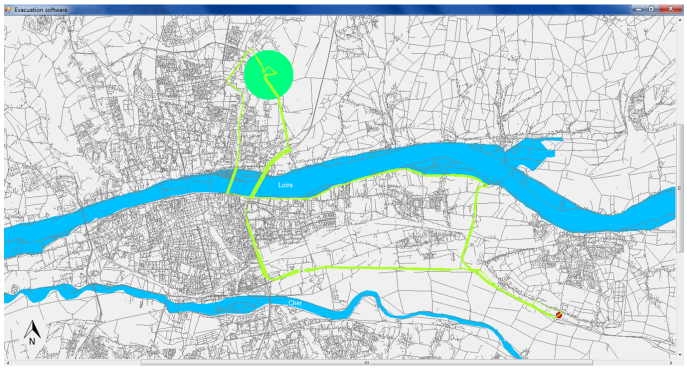


FIGURE 4 – Behavior of K−min and K−all based on time criterion

3.1 Behavior of the two algorithms according to the selected criteria

The figure 4 illustrates the behavior of K−min and K−all according to these two criteria. The thickness of each green line corresponds to the capacity of the arc that the line represents. Part (a) shows seven paths calculated by the algorithm K−min according to the criterion "time" between a building to be evacuated and its destination. While only two paths were calculated by the algorithm K−all according to the same criterion (b). This difference in the number of paths is related to the quick disconnect of network using the algorithm K-all. Parts (c) and (d) show the behavior of these two algorithms but based on the time divided by number of lanes criterion. We notice a difference of computed paths by the two algorithms according to two selected criteria.

The efficiency of the two algorithms and the two criteria will be compared below.

3.2 Comparison of the two algorithms based on the two criteria selected

A comparison between K−min and K−all, according to time and time divided by number of lanes criteria, based on the capacities of the K− shortest paths computed among buildings and shelters is given by the figure 5.

Each point shown in this figure corresponds to the capacity of K− best paths from a given building. Note that the capacity of K− best paths is equal to the sum of capacities of all paths if these latter don't share a common arcs (the case of paths computed by K− all). Otherwise this capacity is computed iteratively taken into account the common arcs among paths. The number of paths computed by K−min is limited to 2 because K−all has provided only two paths at the maximum per building.

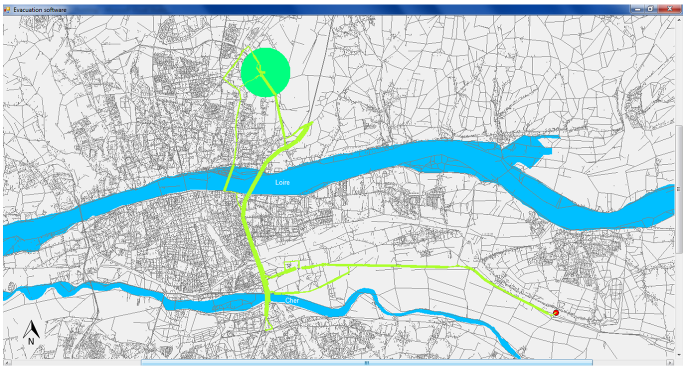
The difference illustrated by the capacities of K−best paths computed by $K-min$ and $K-all$ according to time and time divided by number of lanes criteria, will be visible at the level of the total evacuation time in the figure 6. The first curve (in blue) shows the relation between the total evacuation time and the number of paths computed by the algorithm K−min according to time criterion. While the second one (in red) shows this relation by the same algorithm but according to the other criterion ($\frac{time}{number\ of\ lanes}$).

The algorithm K−all quickly disconnecting the network (2 paths at maximum were calculated for each building) results in a total evacuation time smaller compared with the algorithm K−min for the same number of paths. This disparity is the result of the different strategies followed by K−min and K−all. Beyond 4 paths computed by K-min, the total evacuation time sees a remarkable decline compared to that provided by 2−all.

The exclusion of all arcs belonging to shortest paths computed in previous iterations, the strategy adopted by K-all, results in, as indicated above, the quick disconnect of network between each origin-destination pair and therefore a part of capacity of network may be lost.

2. France 37
3. By assigning each building to the nearest network node

(c) K−Min : time divided by number of lanes criterion



(d) K−all : time divided by number of lanes criterion

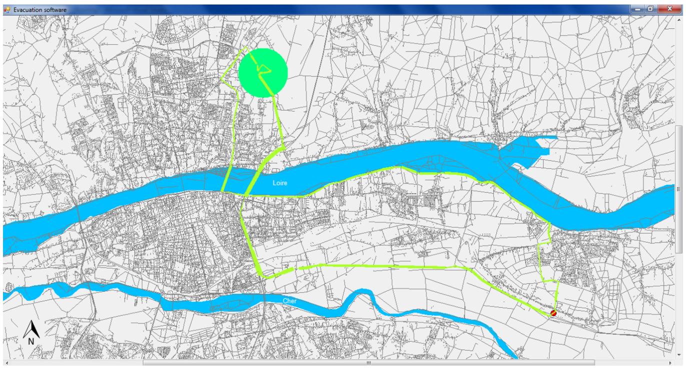


FIGURE 4 – Behavior of K−min and K−all based on time divided by number of lanes criterion

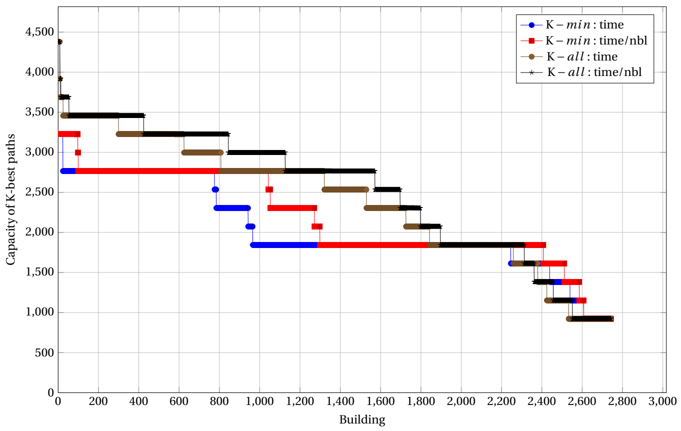


FIGURE 5 – Comparison of the capacity of K−best paths from each building between K−min and K−all (K = 2) according to the two criteria : time and time divided by number of lanes

About the criteria selected, the time divided by the number of lanes is more effective to minimize the total evacuation time than the travel time criterion. Moreover, since it is not possible to evacuate people from a building by a large number of paths, the algorithm K−all seems more appropriate. However, we must pay attention not only to the total evacuation time but also to the length of each path computed in order to make acceptable the evacuation routes by the population concerned.

The figure 7 shows the inverse relationship between the number of computed paths and the number of buildings. In other words, all buildings have at least one escape route and beyond that number (K > 1) the number of buildings with K-best paths gradually decreases.

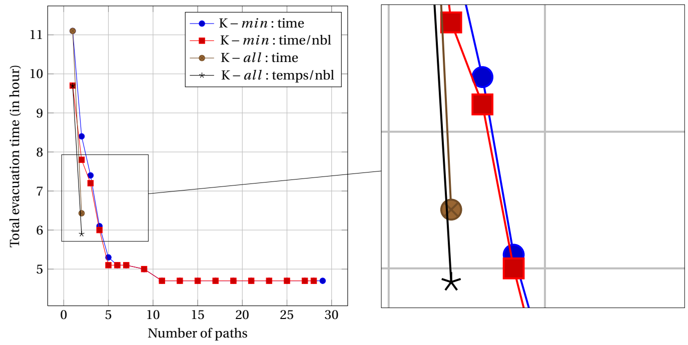


FIGURE 6 – Comparison of the total evacuation time in terms of number of paths (K) between K−min and K−all (K = 2) according to the two criteria : time and time divided by number of lanes

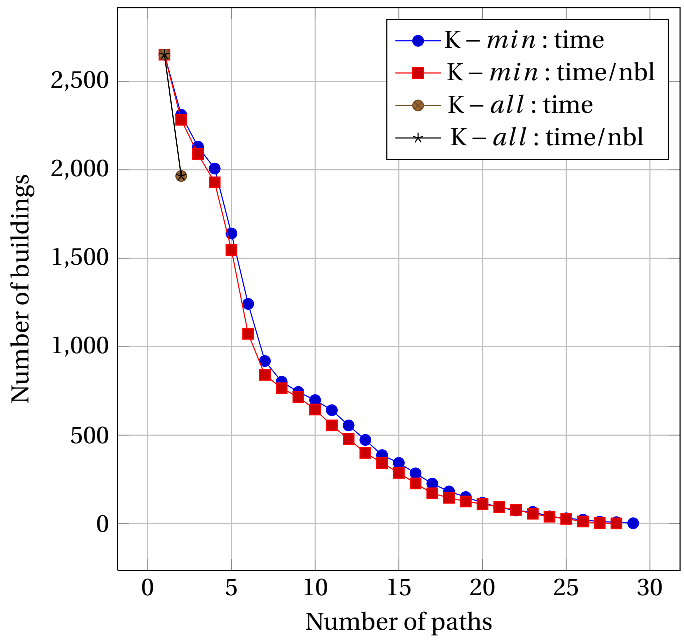


FIGURE 7 – Comparison of number of buildings in terms of number of paths between K−min and K−all (K = 2) according to the two criteria : time and time divided by number of lanes

Moreover the figure 8 illustrates the relationship between the average travel time from buildings to safe points and the index of the computed path (1^{st} shortest path, 2^{nd} shortest path, etc.). The average journey time on the k^{th} path is the sum of travel times on this path divided by the number of buildings that have it. We remark in this figure that the average travel time on the 2^{nd} shortest path computed by K−all is almost equal to that on the 5^{nd} shortest path computed by $K - min$. Therefore, this second path may be faced with a problem of non-compliance by the evacuees.

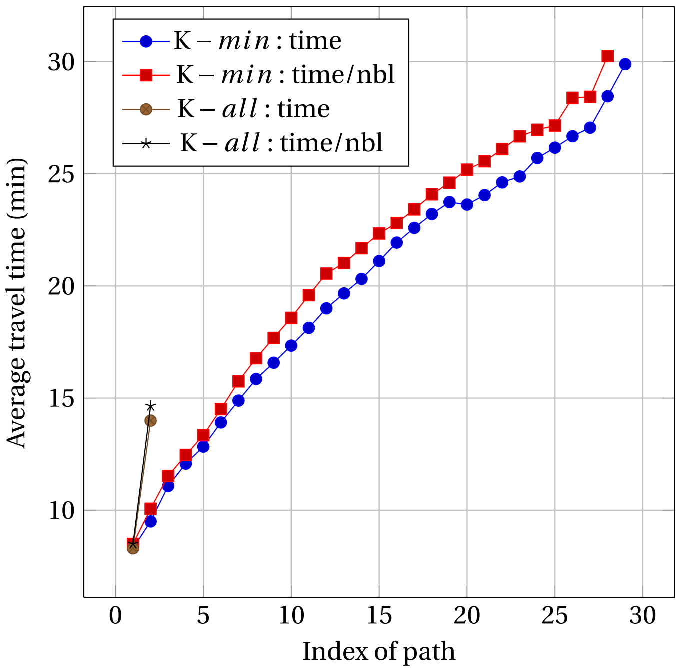


FIGURE 8 – Comparison of average travel times according to index of paths between K−min and K−all (K = 2) (criteria : time and time divided by number of lanes)

Given the foregoing, we adopted in STOM a compromise approach by applying K-min method according to time divided by number of lanes criterion, with K = 3.

Conclusion

In this paper we presented a ranking methods to compute the K-best paths in network. The free-flow travel time, the capacity and the number of lanes of roads are the criteria on which these algorithms were based. The optimization approach followed in this paper allowed to combine between time and traffic. An application and results were performed for the case of a real evacuation.

References

[1] H. Alaeddine, K. Serrhini, M. Maïzia, and E. Néron. A spatio-temporel optimization model for the evacuation of the population exposed to natural disasters. *portail.scd.univ-tours.fr/iii/encore/?lang=eng*, 2014.

[2] K. AlNasr, L. Chen, D. Ranjan, M. Zubair, D. Si, , and J. He. A constrained k-shortest path algorithm to rank the topologies of the protein secondary structure elements detected in cryoem volume maps. *proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, pages 181–186, 2013.

[3] R. Bellman and G. Adomian. Dynamic programming and partial differential equations. *Mathematics and Its Applications*, 15:28–35, 1985.

[4] A. Brander and M. Sinclair. A comparative study of k-shortest path algorithms. *Proc 11th UK Performance Engineering Worksh. for computer and Telecommunications Systems*, 1995.

[5] Y. Chen, D. Rinks, and K. Tang. The first k minimum cost paths in a time-schedule network. *Journal of the Operational Research Society*, 52:102–108, 2001.

[6] I. Das and J. Dennis. Normal-boundary intersection: An alternate approach for generating pareto-optimal points in multicriteria optimization problems. *SIAM J. Optim*, 8:631–657, 1998.

[7] E. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik 1*, page 83–89, 1959.

[8] M. Dredman and R. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3):596–615, 1987.

[9] D. Eppstein. Finding the k shortest paths. *Univ. of California, Irvine, Dept. of Information and Computer Science, USA*, pages 94–26, 1994.

[10] A. Hamed. A genetic algorithm for finding the k shortest paths in a network. *Egyptian Informatics Journal*, 11:75–79, 2010.

[11] J. Hershberger, M. Maxel, and S. Suri. Finding the k shortest simple paths: A new algorithm and its implementation. *ACM Transactions on Algorithms (TALG)*, 3, 2007.

[12] J. Hershberger and S. Suri. Vickrey prices and shortest paths: What in an edge worth. *In Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, page 252–259, 2001.

[13] W. Hoffmana and R. Pavley. A method for the solution of the nth best path problem. *Journal of the Association for Computing Machinery (ACM)*, 6:506–514, 1959.

[14] O. Jahn, R. Mohring, A. Schulus, and N. Stier-Moses. System-optimal routing of traffic flows with user constraints in networks with congestion. *Operations Research*, 53:600–616, 2005.

[15] V. Jimenez and A. Marzal. Computing the k shortest paths: A new algorithm and an experimental comparison. *J.S. Vitter and C.D. Zaroliagis (Eds.): WAE'99, LNCS 1668*, pages 15–29, 1999.

[16] V. Jimenez and A. Marzal. A lazy version of eppstein's k shortest paths algorithm. *In WEA. Springer*, pages 179–190, 2003.

[17] N. Katoh, T. Ibaraki, and H. Mine. An efficient algorithm for k shortest simple paths. *Networks*, 12:411–427, 1982.

[18] E. Lawler. Management science. *Theory Series*, 18:401–405, 1972.

[19] S. Lee and C. Wu. A k-best paths algorithm for highly reliable communication networks. *IEICE TRANS. COMMUN*, E82-B, 1999.

[20] E. Martins. An algorithm for ranking paths that may contain cycles. *Journal of Operational Research*, 18:123–130, 1984.

[21] E. Mohammadi and A. Hunter. Multi-criteria path finding. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 157–159, 2012.

[22] E. Néron and G. Sauvanet. Search for the best compromise solution on multiobjective shortest path problem. *Electronic Notes in Discrete Mathematics*, 2010.

[23] J. Pangilinan and G. Janssens. Evolutionary algorithms for the multiobjective shortest path problem. *Mathematics and Its Applications*, 1, 2007.

[24] G. Sauvanet, E. Néron, and H. Baptiste. *Recherche de chemins multiobjectif pour la conception et la réalisation d'une centrale de mobilité destinée aux cyclistes.* PhD thesis, Université François-Rabelais de Tours, 2011.

[25] A. Stepanov and J. Smith. Multi-objective evacuation routing in transportation networks. *European Journal of Operational Research*, 2008.

[26] Q. Wu and J. Hartley. Using k-shortest paths algorithms to accommodate user preferences in the optimization of public transport travel. *Applications of Advanced Technologies in Transportation Engineering*, 2004.

[27] J. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17:712–716, 1971.

Houssein Alaeddine, Laboratoires Citères et Informatique de l'Université François-Rabelais de Tours

E-mail address: houssein.alaeddine@hotmail.fr

Kamal Serhini, Laboratoire Citères de l'Université François-Rabelais de Tours

E-mail address: kamal.serrhini@univ-tours.fr

Mindjid Maïzia, Laboratoire Citères de l'Université François-Rabelais de Tours

E-mail address: mindjid.maizia@univ-tours.fr

Emmanuel Néron, Laboratoire Informatique de l'Université François-Rabelais de Tours

E-mail address emmanuel.neron@univ-tours.fr