Journal of Computer Science

Original Research Paper

Establishing an Evacuation Network: A Path Ranking Approach

[1,2]Houssein Alaeddine, [1]Kamal Serrhini, [1]Mindjid Maizia and [2]Emmanuel Néron

[1]UMR, CItés, TERritoires, Environnement et Sociétés (CITERES), Université François-Rabelais, Tours, France
[2]ERL CNRS, Laboratoire d'Informatique Fondamentale et Appliquée de Tours (LIFAT), Université François-Rabelais, Tours, France

Article history
Received: 27-08-2019
Revised: 31-10-2019
Accepted: 03-12-2019

Corresponding Author:
Houssein Alaeddine
UMR, CItés, TERritoires,
Environnement et Sociétés
(CITERES), Université
François-Rabelais, Tours, France
Email: houssein.alaeddine@hotmail.fr

Abstract: The evacuation of population exposed to flood hazard requires the establishment of a specific transportation network. This is to compute escape routes to be taken by affected population. At evacuation time, inhabitants of affected area must be evacuated through transportation network to safety area. The main objective of this work is to provide assistance to rescue forces in terms of accessibility by providing itinieraries between buildings at risk of flooding and safety points equipped for this purpose. Technically, the problem of *k*-shortest paths between two nodes in network has been extensively studied in the literature offering several efficient algorithms for different applications. Those algorithms so-called ranking methods aim to compute the first shortest path then the second one and so on. It's well known that the computation in ranking methods is based only on one criterion which is generally distance or 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 method aims to compute a Pareto front that represents a set of non-dominated paths. However it is very difficult to take a decision on which *k*-paths to select among this set and especially when selected paths are served as a data input for a sensitive problem such as the evacuation. We contribute in this paper to establish an evacuation network dedicated to the evacuation of population exposed to natural hazards and more particularly to flood hazard. Two ranking methods to compute paths between each origin (building) - destination (shelter) pair are presented. The criteria free-flow travel time, capacity and number of lanes of road are considered in computing paths. Thus we aim in the proposed ranking approach to simulate a multi-criteria aspect by combining travel time and number of lanes as weight function. The establishment of network (determination of *k*-paths between each building and shelter associated) is then performed according to several measures we introduce in this paper.

Keywords: Accessibility, Ranking Path, Mass Evacuation, Optimization, *k*-Paths

Introduction

This paper addresses the problem of the mass evacuation of cities affected by flooding. The evacuation of persons exposed to natural hazards such as floods and tsunamis occupies a priority position on the agenda of governments of multiple countries. As mentioned by The New York Times, the 2018's flash flood that happened in the Aude region in southwestern France caused at least 11 dead. Moreover, Midwestern and Southern United States have experienced major floods in March 2019 where nearly 14 million people have been affected by the flooding. The occurrence of such event requires the development of specific tool dedicated for the evacuation of population. This tool must ensure quick and safe evacuation in order to reduce the number of potential victims.

An evacuation system must consider several constraints into account such as accessibility, traffic network, safety points, material equipment, human resources, etc.

In France, exist two types of evacuation (1) a preventive evacuation based on a flood forecasting system and concern especially cities protected by dykes (2) an evacuation during the disaster based on flooding scenarios and real time information. This type of evacuation concerns cities affected by slow-onset flooding.

Several steps are required to simulate the evacuation of an urban area such as the preparation of spatial database (urban network, buildings to be evacuated and safety points), the establishment of evacuation network, the minimization of total evacuation time and the feasibility of evacuation plan (Alaeddine *et al*., 2015). We are interested in this paper by the establishment of evacuation network in case of preventive evacuation. This network allows to evacuate people from area at risk to safety area. Note that in case of evacuation during disaster, this network or a part of it can be updated according to dynamic flooding scenarios and real time information (accident, flooded road, etc.,).

The computation of *k*-shortest paths in a graph, for a given pair of nodes, was the subject of many research works in different disciplines such as computer science, operations research and engineering. Their applications include network and electrical routing, transportation, evacuation, etc., (Hershberger and Suri, 2001). Sabri *et al.* (2014) used Dijkstra's algorithm to assist evacuation route in higher and close building. This is to find the exit and safe route in the short time to help evacuee to escape from building safely. The method implemented in Matlab involves three steps: (1) Structuring the layout plan of the selected building, (2) creating the graph where nodes represent doors and weight is the distance between nodes, (3) computing the shortest path between the current place of the evacuee in the building and the safe place. This model is interesting for building evacuation, the selfish behavior of evacuees is taken into account where all evacuees are assigned to the first shortest egress route. Indeed evacuees choose the fastest routes to leave the building. However the developed model faces some difficulties such as graph size (declaration of matrix in Matlab is of limited size), high density of shared roads since evacuees are assigned to the first shortest path. Indeed, the evacuation time will be considerable and the evacuation process may not to be safe as evacuees must stop and wait on congested roads. These difficulties are overcomed in our presented paper while keeping the evacuees' behavior in mind.

Stepanov and Smith (2009) have addressed a multi-objective evacuation routing in transportation networks. The measures clearance time, total travelled distance and blocking probabilities are used to evaluate the evacuation plan. Their work is interesting and comprehensive since all evacuation steps, including evacuees' behavior, are included in the developed model. A benchmark for an optimal routing policy is established, where all evacuees are assigned to the first shortest egress routes. According to authors, this policy generates blockage on some raods with small capacity i.e., evacuees will not be able to proceed to some roads (due to their saturation) and have to stop and wait. This questions the safety of the established evacuation plan. The second policy allows to evacuate people through 15 shortest paths in order to minimize road blockage and therefore total clearence time. These paths are calculated using Eppstein's algorithm (represent all possible deviations from the shortest path) (Eppstein, 1999). However the total travelled distance has increased, which questions the feasibility of evacuation plan in terms of evacuees' behavior. Moreover, the applications are based on a theoretic small graph. A huge transportation network might increase the computation time considerably. In addition, the authors considered a simple criterion (time) to build an evacuation network. These limits are also overcomed in our paper.

k-shortest Paths Problem

The ranking *k*-shortest path is a more generalized problem of the basic problem of finding the shortest path between an origin-destination pair (Hoffmana and Pavley, 1959). It aims to compute at each iteration the *k*-th shortest path that has axiomatically a cost (time or distance) higher or equal to that of the $(k\text{-}1)_{th}$ path. For different reasons the computation of the second (third, etc.,) shortest path is important. For example, in emergency situation the first shortest path between an origin-destination cannot be the only escape route for 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 (Mohammadi and Hunter, 2012)

Chen *et al.* (2001) 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 in addition to the waiting time on nodes are the criteria used. Hershberger *et al.* (2007) 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 (2001). Let $p = \{a_1,...,a_r\}$ be a shortest path from an origin s to a destination *t*. The replacement paths problem aims to find at each iteration $1 \leq i \leq r$ the shortest path from s to t that does not pass through the arc a_i. Eppstein (1999) addressed the problem of *k* shortest paths with minimum total length using heap-ordered tree where cycles of repeated nodes are allowed. The method is to compute the shortest path tree then to build a graph to represent all possible deviations from the shortest path. According to Jimenez and Marzal (2003), 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 selecting the *k*-shortest paths. Hamed (2010) developed a genetic algorithm to compute the *k*-shortest paths based on a real world system (links bandwidth of the network) where repeated nodes are allowed. A comparative study of *k*-shortest paths algorithms was studied by Brander and Sinclair (1995), where four methods (Yen, 1971; Lawler, 1972; Katoh *et al*., 1982; Hoffmana and Pavley. 1959) were selected from over seventy papers focusing on this problem. Moreover, in order to reduce street congestion in modern cities, Wu and Hartley (2004) presented *k*-shortest paths algorithms to accommodate user preference in the optimization of public transport travel. This would encourage private-vehicle drivers to use public transportation. Jimenez and Marzal (1999) addressed the computation of *k*-shortest paths between a source and a sink by developing a recursive enumeration algorithm to solve a set of equations by generalizing the equations of Bellman and Adomian (1985) for the single shortest path problem.

On the other side, several authors have addressed the shortest path problem on a multi-criteria basis. Lee and Wu (1999) have developed an algorithm to calculate the k-best paths with a minimum total cost and a minimum number of common nodes. Pangilinan and Janssens (2007) have addressed the problem of multi-objectives path aiming to find all effective ways (non-dominated or Pareto-optimal) from a source node to a destination node with multiple objectives. Sauvanet *et al*. (2011) addressed as well the multi-objective path problem for the benefit of cyclists, where several criteria such as distance, insecurity and stress are considered (Néron and Sauvanet, 2010).

Contribution

The contribution of this paper is to establish a specific transportation network dedicated to the evacuation of population exposed to flood hazard. We present two raking algorithms based on travel time, capacity and number of lanes of road criteria to compute shortest paths between each building-shelter pair. The ratio $\frac{travel\,time}{number\,of\,lanes}$ is used as weight function in the computation of shortest path. This ratio tends to simulate a multi-criteria aspect by combining both time and capacity. Finally, the construction of evacuation network, namely the determination of *k*–shortest paths between each building and shelter associated, is performed using several evaluation measures introduced in this paper. More concretely, the main contribution of this paper is to assist rescue forces and decision makers involved in flood risk management to apply the evacuation plan in a short time and therefore reduce the number of victims.

The algorithms are implemented in C++ language. The incidence list (linked list) is used to represent huge transportation network. The Dijkstra's algorithm (Dijkstra, 1959) is implemented here with Fibonacci heap (Dredman and Tarjan, 1987) to speed up calculation. It should be noted that Dijkstra's algorithm can be replaced by any other algorithm to compute the shortest path between source-destination nodes.

Ranking Methods

Let us consider $B = \left\{\mathscr{B}_1, ..., \mathscr{B}_n\right\}$ a set of *n* buildings to be evacuated towards a set of *m* shelters $S = \left\{\mathscr{S}_1, ..., \mathscr{S}_m\right\}$. We assume that one or more shelters are assigned to each building. The aim here is to build an evacuation network $\mathscr{G} = (\mathscr{N}, \mathscr{A})$ where $\mathscr{N}$ is the set of nodes and $\mathscr{A}$ is the set of arcs i.e., compute *k* evacuation paths between each building and each safety point associated.

k-Min Method

This method aims to compute the *k*-shortest paths between a given origin-destination. It gets its name from the fact that arc(*s*) with minimum capacity are removed (Fig. 1). At each iteration, the *k*-min algorithm computes the shortest path, then determines the new (residual) capacity of each arc belonging to the computed path. The new capacity is equal to the current capacity minus the capacity of minimum arc. Finally, the iteration is terminated by removing from graph the arc (*s*) with zero capacity.

The pseudo-algorithm 1 describes the *k*–min method. Moreover the Fig. 2 shows the behavior of this algorithm on a small theoretical graph.

Algorithm 1 *k*–min

```
procedure find the k− paths
  Input: Network, Buildings, Shelters
  for each building $\mathscr{B}_i$ do
    for each safety point associated $\mathscr{S}_j$ do
      repeat
        Compute the shortest path p between $\mathscr{B}_j$ and $\mathscr{S}_j$
        Reduce the capacity of arcs a belonging to p
            ▹ capacity of a ← current capacity of $a - \min_{e \in p} c(e)$
        Deactivate all arcs with zero capacity
      until there is no path
      Reactivate arcs and reset their capacities
    end for
  end for
end procedure
```

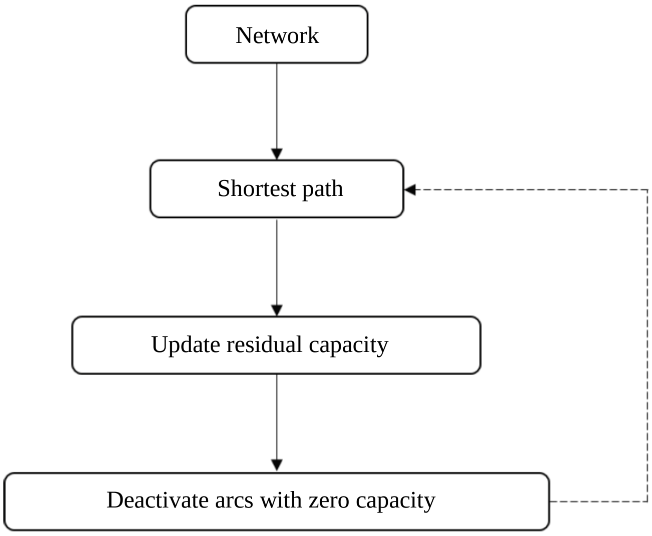
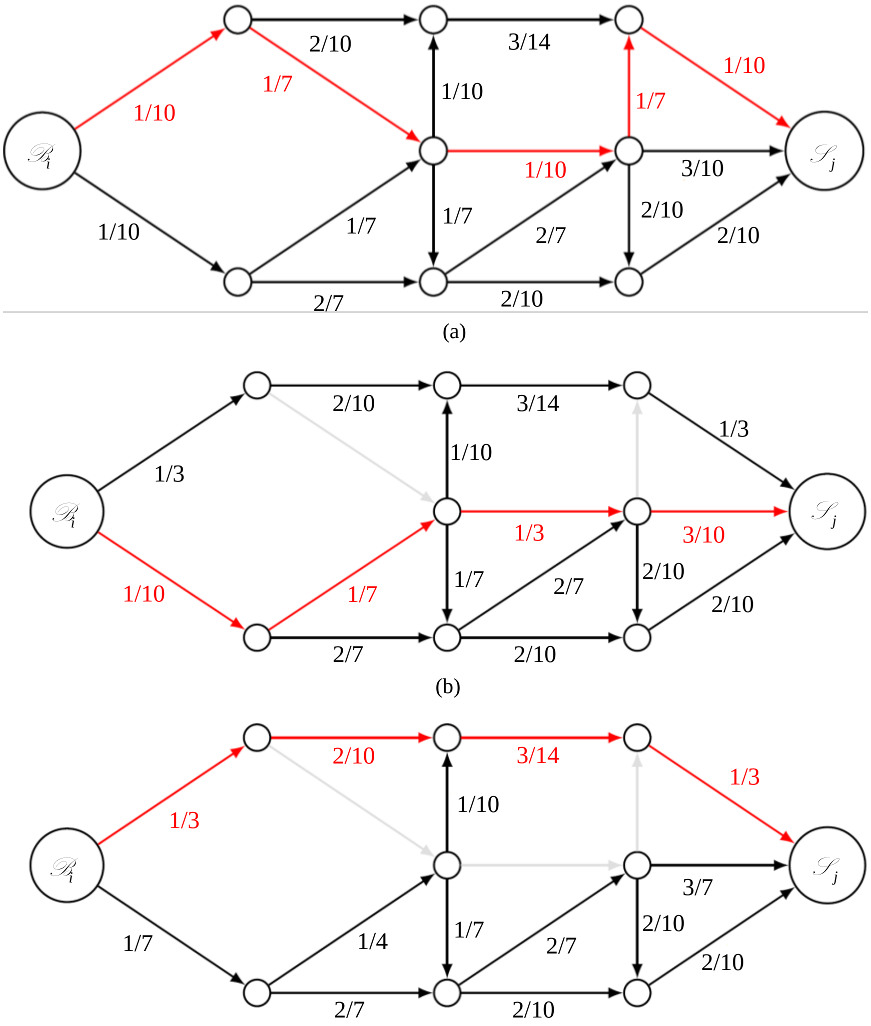


Fig. 1: k-min method



(a)

(b)

(c)

Fig. 2: Behavior of *k*-min algorithm. Weight on arc: Travel time/capacity(a) Iteration 1; (b) Iteration 2; (c) Iteration

Thus, this method enables to calculate only useful paths for each building. Indeed paths sharing minimum arc in terms of capacity are of no interest. This strategy allows to reduce road blockage as residents of a given building can be assigned to different paths. However, the number of paths must be chosen carefully as evacuees from same building might not accept to exit through several different paths. In fact, rescue forces can not obligate evacuees from same building to evacuate through 10 different paths for example. Moreover, the travelled distance(time) must be evaluated because residents are unfamiliar with longest distances. Indeed, evacuees choose the fastest (familiar) routes to leave the building. This evaluation related to the acceptability of evacuation plan will be performed on a huge real graph in the experimentation section.

Method k-all

The 2nd method *k*-all was proposed by Martins (1984) under the name of "removing path method". It also aims to compute the *k*− shortest paths between a given building-shelter. This method gets its name from the fact that all arcs are removed (Fig. 3). At each iteration, the *k*-all algorithm computes the shortest path, then deactivate all arcs belonging to the path calculated. Thus, paths calculated are arc-disjoint i.e., they have no common arc.

The pseudo-algorithm 2 describes the method *k*−all while the Fig. 4 shows the behavior of this algorithm on a small graph.

Algorithm 2 *k*−all

```
procedure FIND THE k− PATHS
    Input: Network, Buildings, Shelters
    for each building $\mathscr{R}_i$ do
        for each safety point associated $\mathscr{S}_j$ do
            repeat
                Compute the shortest path p between $\mathscr{R}_i$ and $\mathscr{S}_j$
                Deactivate all arcs belonging to p
            until there is no path
            Reactivate arcs
        end for
    end for
end procedure
```

We remark that only 2 paths have been found between $\mathscr{R}_i$ and $\mathscr{S}_j$. Therefore, a part of capacity of network between $\mathscr{R}_i$ and $\mathscr{S}_j$ is lost because all arcs are removed. The sum of capacity of paths calculated is 14 units while this network can support more flow from $\mathscr{R}_i$ to $\mathscr{S}_j$.

The number of paths generated by *k*−all depends strongly on the connectivity of network. Indeed, a transportation network with weak connectivity might disconnect quickly (starting from the first shortest path). This disconnectivity will manifest itself on a real graph as we will show in the result section. Moreover, the travel time on the second (third,...) path, if found, must be evaluated according to the behavior of evacuees because the paths calculated by this method seem to be long as they are arc-disjoint. The result section evaluates the output of this method on a real network according to several measures such as average travelled time and total evacuation time.

Selection of k-Paths

The *k*−min algorithm provides several paths between each origin-destination pair. Two complementary strategies can be used to determine *k*. The first strategy is based on setting a tolerance factor $\phi \geq 1$ (Jahn *et al*., 2005) where only paths whose lengths are smaller than the first shortest path multiplied by ϕ are considered. The second strategy consists in considering only paths that verify $\lambda t + (1-\lambda)c < \delta$ where t is the path length, c its capacity and $\lambda \in [0, 1]$. However, it is difficult to determine the values of ϕ, λ and δ.

The travel time used in computation is an additive criterion, while the capacity is a maximization criterion i.e., it cannot be considered as a cost function by shortest path algorithms. In order to combine both criteria (time, capacity), we propose the cost function: Time divided by capacity or time divided by number of lanes.

In our evacuation model STOM[1], we consider only the first three shortest paths ($k = 3$) provided by *k*−min between each origin-destination. Indeed, it is not easy for residents to recongnize more than 3 different paths from their building to the shelter associated (Alaeddine *et al*., 2015).

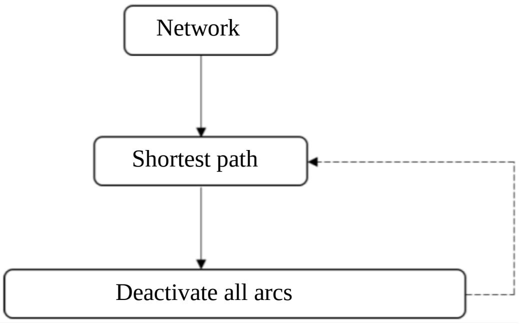
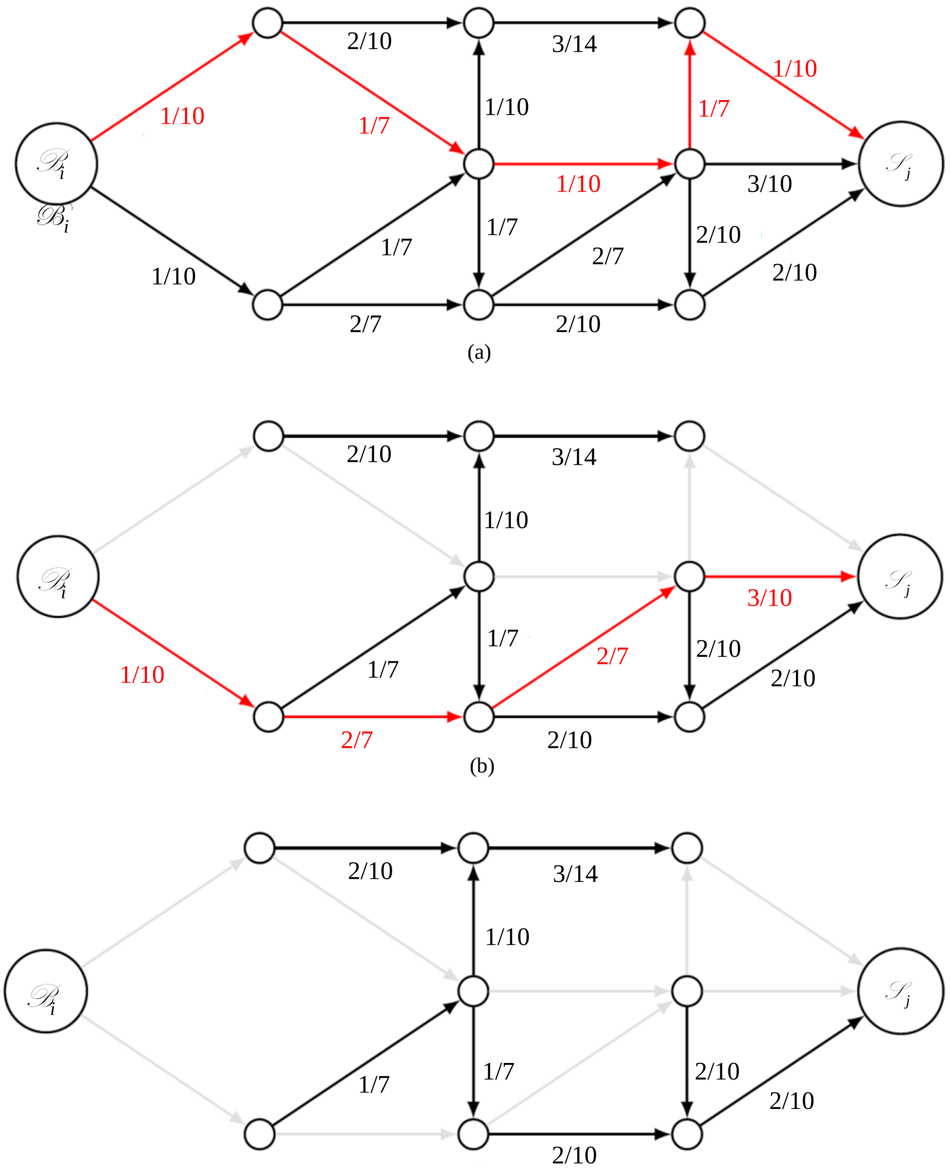


Fig. 3: *k*-all method

[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)



(a)

(b)

(c)

Fig. 4: Behavior of *k*- all algorithm. Weight on arc: travel time/capacity(a) Iteration 1; (b) Iteration 2; (c) Iteration 3

The next section justify this choice. A comparison between *k*−min and *k*−all according to different criteria and measures is performed.

Applications and Results

The evacuation model is applied and tested on the valley of Tours[2]. This site is composed of 2 safety points and 9825 buildings grouped into 2753 buildings[3]. According to shleter capacity and proximity, buildings in green (60%) are to be evacuated to the North, while buildings in brown (40%) are to be evacuated to the South (Fig. 5) (Alaeddine *et al*., 2015).

This section aims to compare the algorithms *k*−min and *k*−all on this real graph according to, travel time and travel time divided by number of lanes criteria and average travelled time and total clearence time measures.

Comparison of Algorithms

The Fig. 6 illustrates the results of *k*−min and *k*−all on a real graph. For a clarity purpose, we present only the paths between one origin-destination. The line thickness corresponds to arc capacity.

[2] France 37

[3] By assigning each building to the nearest network node

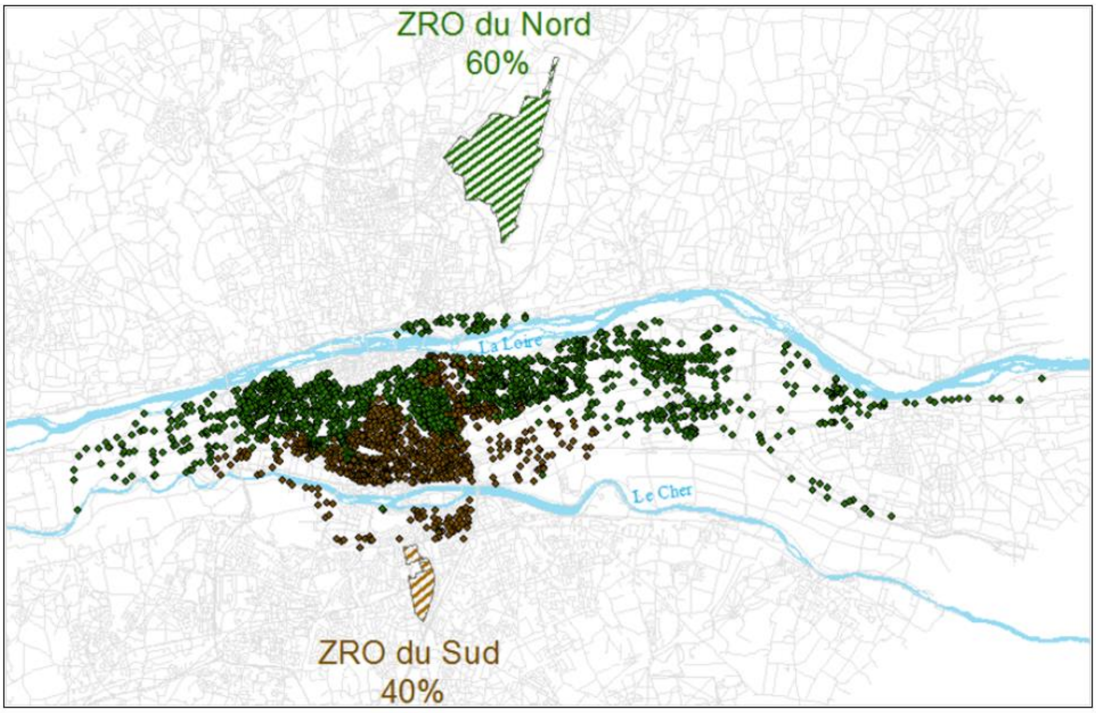
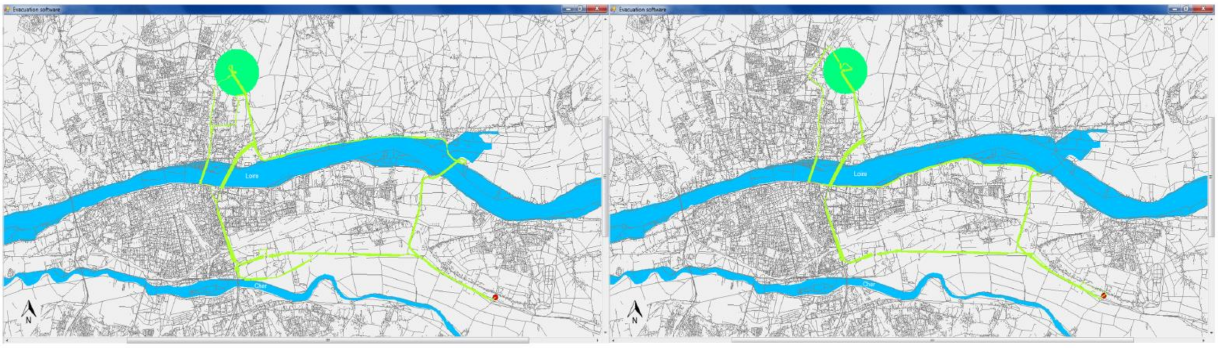


Fig. 5: The valley of Tours, FR 37



(a)

(b)

Fig. 6: Paths calculated by *k*-min and *k*-all between one origin-destination; (a) *k*-Min: Time criterion; (b) *k*-all: Time criterion

The part (a) shows seven paths calculated by the algorithm k−min, while only two paths are calculated by *k*−all according to the same criterion "time". The difference in number of paths is related to the quick disconnect of network using *k*-all.

The Fig. 7 shows the relationship between the number of buildings and the number of paths calculated. We remark that all buildings have at least one evacuation path, then the number of buidings gradually decreases with $k > 1$.

Algorithms and Criteria Comparison

A comparison between 2−min and 2−all, according to both criteria, is performed in the Fig. 8. The number of paths is limited to 2 since the algorithm *k*−all has provided two paths at max per building (Fig. 7). This figure shows the sum of capacities of paths[4] for each building according to both algorithms and both criteria. Each point represented in the figure corresponds to (building id[5], sum of capacities of paths from that building). Note that the sum of capacities of paths is computed iteratively taking into account the common arcs between paths. We remark that *k*−all dominates *k*−min since paths generated by *k*- al l are arc-disjoint.

[4] the sum of capacities of paths that connect each buiding to its shelter associated

[5] buildings are numbered 0, 1, 2,...

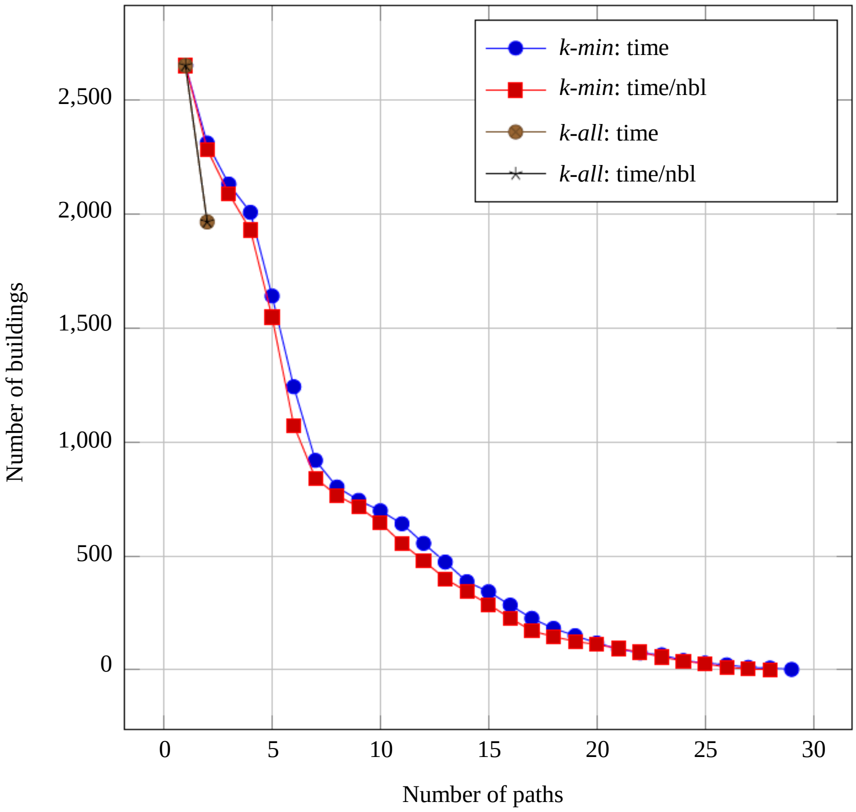


Fig. 7: Relationship between number of paths calculated and number of buildings. Note: nbl (number of lanes)

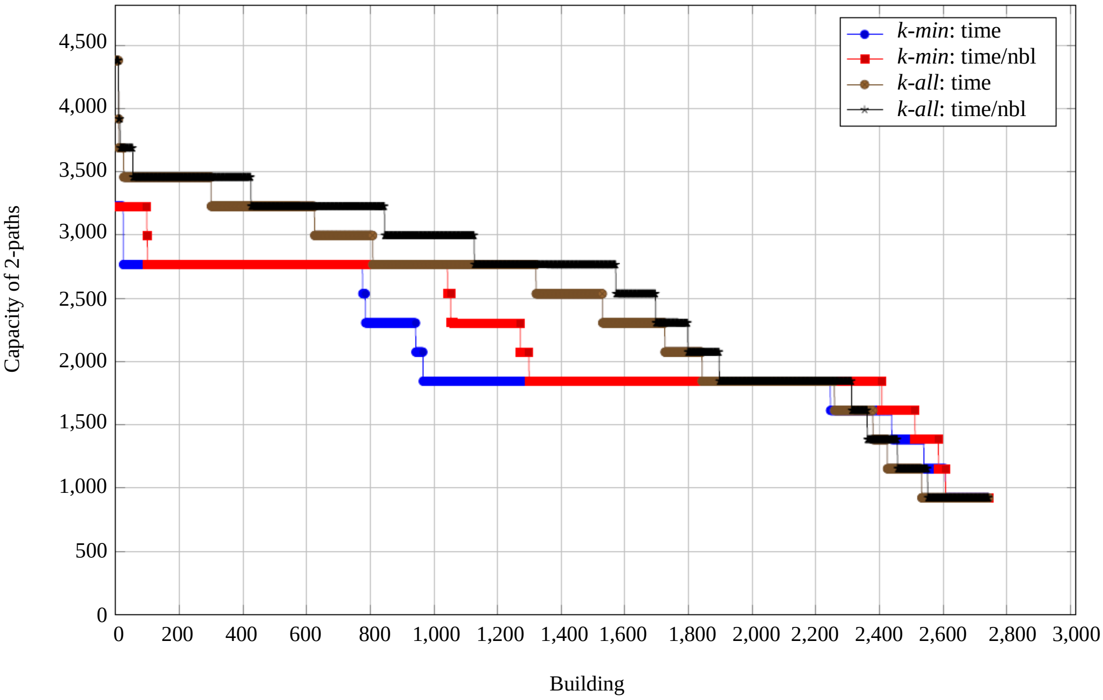


Fig. 8: Relationship between buildings and capacities of 2-paths

The difference in results in both algorithms and criteria, in terms of sum of capacities, will be illustrated according to total evacuation time (Fig. 9). The first two curves (in blue and red respectively) show the relationship between the total evacuation time and the number of paths computed by the algorithm k−min according to time criterion and $\frac{time}{number\ of\ lanes}$ criterion respectively. The other two curves concern the algorithm k-all. The algorithm k−all quickly disconnects the network, two paths at max have been provided per building. We remark that the total evacuation time is smaller in k−all than in k−min for the same number of paths (2, for example). Beyond 4 paths calculated by k-min, the total evacuation time decreases remarkably and gets smaller than that for 2−all. Indeed, the quick disconnect of network caused by k−all results in loosing a part of network capacity.

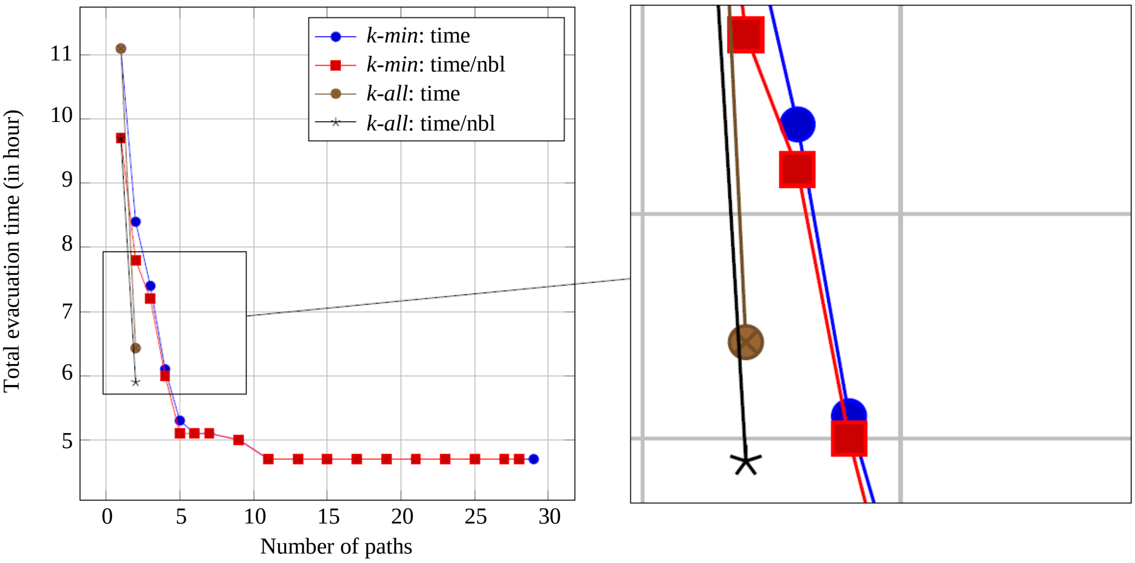


Fig. 9: Relationship between number of paths and total evacuation time

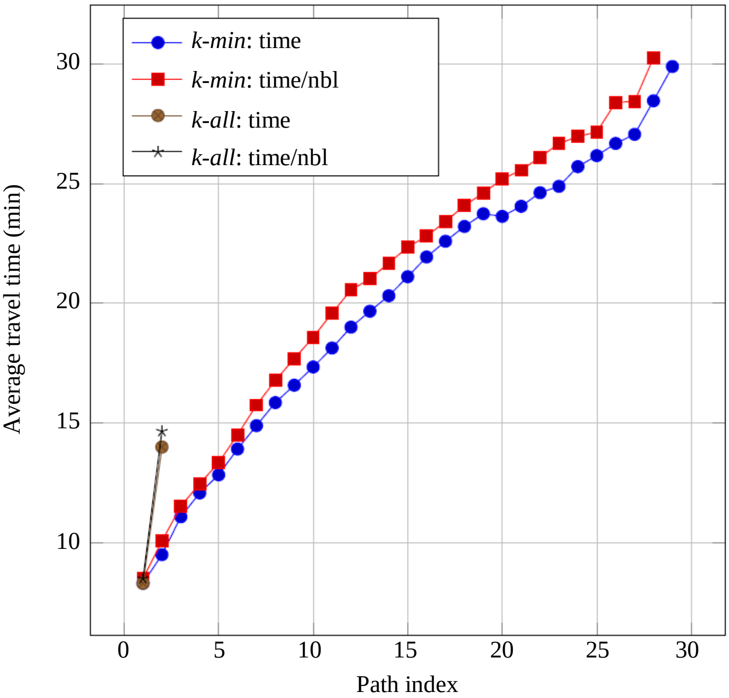


Fig. 10: Relationship between path index and average travel time

Concerning the criteria, we notice that time divided by number of lanes is more effective than time criterion to minimize evacuation time. As it is not easy to evacuate people of same building via several paths, the algorithm k−all seems more appropriate (2−all dominates k−min for $k < 5$ in terms of evacuation time). However, the length of second path provided by 2−all must be evaluated because it risks to not be respected by evacuated people.

The Fig. 10 illustrates the relationship between average travel time and path index (1^{st} shortest path, 2^{nd} shortest path, etc.,). The average travel time on k^{th} path is equal to the sum of travel times on all k_{th} paths divided by their number (number of buildings that have k^{th} path). We remark that the average travel time on the 2^{nd} shortest path computed by k−all is almost equal to that on the 6th shortest path computed by k−min. Therefore, the second path of k−all risks to be rejected by evacuees.

Given the foregoing, the method k−min with $k = 3$ is more appropriate to establish an evacuation network. The two measures namely total clearence time and average travel time are both optimized in this method in comparison to k−all while taking into account the behavior of evacuees (Fig. 9 and 10). Therefore, we adopt in STOM a compromise approach by applying 3-min method using $\frac{time}{number\,of\,lanes}$ criterion.

Conclusion

This paper addressed the establishment of a specific transportation network dedicated for the evacuation of population exposed to flood hazard[6]. Two ranking methods so-called k−min and k−all were presented and compared on a real graph according to several measures such as average travelled time and total clearence time. We showed that k−min with $k = 3$ optimizes better these measures while observing the evacuees' behavior. Morevover, we adopted the criterion $\frac{travel\,time}{number\,of\,lanes}$ as weight function in the computation of shortest path. Indeed, this ratio as we mentioned previously aims to simulate a multi-criteria aspect by combining both time and capacity. Finally, we present this work for the assistance of rescue forces to realize the evacuation process in a short time and therefore reduce the number of victims.

[6] This approach can be used in case of land use optimization where the goal is to compute the k-paths between each home-workplace. The projects TRUC and BINS-COEUR, funded by LA REGION CENTRE VAL DE LOIRE have benefited from these IT developments on the sites of Montargis and the two metropolises of Orléans and Tours.

Acknowledgment

We would like to thank the editors Dr. Monir Abdullah Abduh Kaid and Prof. Abdalah Rabab and the three anonymous referees for assessing this paper.

Funding information

We would like to thank LA REGION CENTRE VAL DE LOIRE (France), and FEDER (Europe) for funding this research.

Author's Contributions

All authors have equally contributed in this work.

Ethics

This paper is original and contains unpublished material. The corresponding author confirms that the co-authors have read and approved the manuscript and no ethical issues involved.

References

Alaeddine, H., K. Serrhini, M. Maïzia and E. Néron, 2015. A spatiotemporal optimization model for the evacuation of the population exposed to flood hazard. Natural Hazards Earth Syst. Sci., 15: 687-701.

Bellman, R. and G. Adomian, 1985. Dynamic programming and partial differential equations. Mathe. Applic., 15: 28-35.

Brander, A.W. and M.C. Sinclair, 1995. A comparative study of *k*-shortest path algorithms. Proceedings of the 11th UK Performance Engineering Workshop of Computer and Telecommunications Systems, (CTS' 95), Springer, London, pp: 370-379 DOI: 10.1007/978-1-4471-1007-1_25

Chen, Y.L., D. Rinks and K. Tang, 2001. The first *k* minimum cost paths in a time-schedule network. J. Operat. Res. Society, 52: 102-108.

Dijkstra, E.W., 1959. A note on two problems in connection with graphs. Numerische Mathematik, 1: 83-89.

Dredman, M.L. and R.E. Tarjan, 1987. Fibonacci heaps and their uses in improved network optimization algorithms. J. ACM, 34: 596-615.

Eppstein, D., 1999. Finding the *k* shortest paths. Inform. Comput. Sci., 28: 652-673. DOI: 10.1137/S0097539795290477

Hamed, A.Y., 2010. A genetic algorithm for finding the *k*-shortest paths in a network. Egyptian Inform. J., 11: 75-79.

Hershberger, J. and S. Suri, 2001. Vickrey prices and shortest paths: What in an edge worth. Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science, Oct. 8-11, IEEE Xplore Press, Newport Beach, CA, USA, pp: 252-259. DOI: 10.1109/SFCS.2001.959899

Hershberger, J., M. Maxel and S. Suri. 2007 Finding the *k* shortest simple paths: A new algorithm and its implementation. ACM Tran. Algorithms.

Hoffmana, W. and R. Pavley. 1959. A method for the solution of the nth best path problem. J. Assoc. Comput. Machinery, 6: 506-514.

Jahn, O., R. Mohring, A. Schulus and N. Stier-Moses, 2005. System-optimal routing of traffic flows with user constraints in networks with congestion. Operat. Res., 53: 600-616.

Jimenez, V.M. and A. Marzal, 1999. Computing the *k* shortest paths: A new algorithm and an experimental comparison. Proceedings of the 3rd International Workshop on Algorithm Engineering, Jul. 19-21, Springer, UK, pp: 15-29. DOI: 10.1007/3-540-48318-7_4

Jimenez, V.M. and A. Marzal. 2003. A lazy version of eppstein's *k*-shortest paths algorithm. Proceedings of the 2nd International Conference on Experimental and efficient algorithms, May 26-28, Springer, Ascona, Switzerland, pp: 179-190. DOI: 10.1007/3-540-44867-5_14

Katoh, N., T. Ibaraki and H. Mine, 1982. An efficient algorithm for k shortest simple paths. Networks, 12: 411-427.

Lawler, E.L., 1972. Management science. Theory Series, 18: 401-405.

Lee, S.W. and C.S. Wu, 1999. A *k*-best paths algorithm for highly reliable communication networks. IEICE Trans. Commun., 82: 586-590.

Martins, E.Q., 1984. An algorithm for ranking paths that may contain cycles. J. Operat. Res., 18: 123-130.

Mohammadi, E. and A. Hunter, 2012. Multi-criteria path finding. Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Aug. 25-Sept. 01, Melbourne, Australia, pp: 157-159.

Néron, E. and G. Sauvanet, 2010. Search for the best compromise solution on multiobjective shortest path problem. Electron. Notes Dis. Math., 36: 615-622. DOI: 10.1016/j.endm.2010.05.078

Pangilinan, J.M.A. and G.K. Janssens, 2007. Evolutionary algorithms for the multiobjective shortest path problem. Math. Applic.

Sabri, N.A.M., A.S.H. Basari and K.A.F.A. Samah. 2014. The utilisation of dijkstra's algorithm to assist evacuation route in higher and close building. J. Comput. Sci., 11: 330-336. DOI: 10.3844/jcssp.2015.330.336

Sauvanet, G., E. Néron and H. Baptiste, 2011. 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.

Stepanov, A. and J. Smith, 2009. Multi-objective evacuation routing in transportation networks. Eur. J. Operat. Res., 198: 435-446. DOI: 10.1016/j.ejor.2008.08.025

Wu, Q. and J. Hartley, 2004. Using *k*-shortest paths algorithms to accommodate user preferences in the optimization of public transport travel. Applic. Adv. Technol. Transportat. Eng.

Yen, J.Y., 1971. Finding the *k* shortest loopless paths in a network. Manage. Sci., 17: 712-716.