

# Analytic Prioritization of Indoor Routes for Search and Rescue Operations in Hazardous Environments

V. Zverovich\*, L. Mahdjoubi & P. Boguslawski

University of the West of England, Bristol, UK

&

F. Fadli

Department of Architecture and Urban Planning, Qatar University, Doha, Qatar

**Abstract:** Applications to prioritize indoor routes for emergency situations in a complex built facility have been restricted to building simulations and network approaches. These types of applications often failed to account for the complexity and trade-offs needed to select the optimal indoor path during an emergency situation. In this article, we propose a step change for finding the optimal routes for Search And Rescue (SAR) teams in a building, where a multi-epicenter extreme event is occurring. We have developed an algorithm that is based on a novel approach integrating the Analytic Hierarchy Process (AHP), statistical characteristics, the propagation of hazard, Duckham–Kulik’s adapted algorithm, Dijkstra’s classical algorithm, and the binary search with three criteria: hazard proximity, distance/travel time, and route complexity. The sub-criteria for the route complexity are validated in the context of SAR using a real-life building (Doha World Trade Centre). The important feature of the algorithm is its ability to generate an optimal route depending on user’s needs. The findings revealed that the generated optimal routes are indeed the “best” trade-off among distance/travel time, hazard proximity, and route complexity. The test results also demonstrated the robustness of the algorithm with respect to different parameters, and its insensitivity to different scenarios of uncontrolled evacuation.

\*To whom correspondence should be addressed. E-mail: vadim.zverovich@uwe.ac.uk.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

## 1 INTRODUCTION

Managing efficiently disaster scenes in built facilities is critical for Search And Rescue (SAR) personnel to provide a time critical response, and thus eliminate the potential for the loss of life, injury, and damage. Prevailing approaches have used either heuristic/knowledge rules encapsulated in a prototype computer model or Geography Information Science based emergency management systems. The major limitation of many systems developed so far is the lack of practical applicability to an emergency response that provides decision makers with means to assess several alternatives based on multiple, conflicting criteria (Gomes and Lins, 2002).

To deal with the complexity of variables in an emergency situation, the Analytic Hierarchy Process (AHP) has offered one of the best ways for providing consistent comparisons and selecting the best alternative among several conflicting criteria. As an approach designed to handle decisions with multiple attributes, the AHP has been applied in a wide range of areas. For instance, Wang et al. (2010) recently applied the AHP for developing a comprehensive evaluation index system used to weight railway emergency plans. The AHP approach can be summarized into four main steps and the calculation procedure as follows: (1) structuring a problem into a decision hierarchy consisting of criteria and alternatives; (2) establishing pairwise comparisons between decision elements at each hierarchy level; (3) transforming comparison matrices into sets of weights; (4) aggregating the weights to rank the alternatives.

Despite the potential of the AHP technique, there is currently little research that applied this approach in the context of emergency response. In addition, there has been no attempt to integrate the powerful analytic approach with 3D indoor environments. This is the central issue of our research.

In this article, we propose an algorithm for finding the optimal routes for SAR teams in a building, where a multi-epicenter extreme event is occurring. The important feature of the algorithm is its ability to generate an optimal route depending on user's needs, that is, the user has an option to specify the preferential ranking of three basic criteria. The algorithm is based on a novel approach that integrates a new version of the AHP for multi-attribute decision making, the direct iteration algorithm for eigenvectors/eigenvalues, statistical characteristics, the propagation of hazard, Duckham–Kulik's adapted algorithm for simplest paths, Dijkstra's classical algorithm for shortest paths, and the binary search with three criteria: hazard proximity, distance, and route complexity. We introduce and validate sub-criteria for the route complexity, and further develop hazard proximity numbers and the proximity index for large open spaces in a building. The algorithm is presented in Section 2, and in Section 3 it is validated and tested on a realistic complex building. In Section 4, we illustrate how the distance criterion can be replaced by travel time using Nelson–MacLennan–Pauls' formula for people's indoor speed, and extend this formula for an SAR team moving in a counter-flow. Also, our algorithm is compared with an existing method.

### 1.1 State of the art in SAR and indoor environments

The typical research in emergency response is devoted to evacuation and rescue, with a focus on indoor navigation and route finding. Park et al. (2009) developed a time-dependent optimal routing algorithm based on a 2D network representing the building configuration, which has been enriched by relevant information about the facility. The focus of their research is on computing optimal routes leading SAR personnel to disaster locations, taking into account the location of evacuees and smoke density. The method requires detecting positions of people in a building per time period. For this purpose, an evacuation simulation system for identifying the movement patterns of people in emergency situations was used. Even though the algorithm had considerable potential, the authors concluded that the method needs "further improvements to fully apply to real-time evacuation systems."

The simplest path algorithm for a road network was proposed by Duckham and Kulik (2003). The purpose of their method is to "minimize the complexity of a

route description, based on the amount of information required to negotiate each decision point." It is interesting to note that, unlike shortest paths, simplest paths are neither symmetric nor satisfy the triangle inequality. Although the simplest path algorithm was developed for road networks, one of the advantages of the algorithm is that any weighting function can be used in its input as a complexity function. This property will be used in Section 2, where a new complexity function for indoor navigation is developed.

Another relevant article is by Liu and Zlatanova (2011), who proposed a new door-to-door approach for finding routes between rooms and also a detailed route in a single room. Their algorithm was tested on a 2D floor plan of a building with complex indoor structure. Vanclooster et al. (2014) applied Grum's least risk path algorithm to an indoor space for minimizing risks of getting lost, and proposed several improvements to Grum's algorithm to make it more compatible with indoor networks.

Although this article is devoted to indoor navigation of SAR teams, some important surveys in the area of evacuation are worth mentioning. A review of 16 evacuation models was given by Gwynne et al. (1999), and Kuligowski (2004) reviewed 28 egress models. Further reviews of fire and evacuation models can be found in Friedman (1992), Olenick and Carpenter (2003), and Watts (1987).

Any algorithm for indoor navigation is always based on a spatial model. A good example of such a model was given by Kwan and Lee (2005, Figure 5). The structure of a building is represented as a logical network, where the nodes represent spatial objects such as rooms, corridors, and other navigable areas. The edges represent navigable connections between adjacent objects. The network can be further extended to a geometric network to model precise geometric properties (e.g., distance between nodes and their locations) and provide real navigation routes. Boguslawski et al. (2015) recently developed a 3D building model, which is an integration of Building Information Modeling technology and Geography Information Science analysis.

A number of works in the area of rescue operations during emergency response have focused on stochastic aspects in different contexts. For example, Chang et al. (2007) formulated the flood emergency logistics problem with uncertainty as two stochastic programming models. Aboshosha and Zell (2003) used the stochastic control theory and fuzzy inference systems to provide rescue robots with an adaptive behavior when searching for victims in various disasters. Barbarosoğlu and Arda (2004) developed a two-stage stochastic programming model for planning the transportation of first-aid commodities to disaster areas. Further research

is needed to address uncertain dynamics of extreme events.

Another relevant area is multi-objective shortest path (MOSP) problems. The first part of the algorithm presented in this article consists of MOSP heuristics, which are based on binary searches. There are other interesting approaches, for example, an Ant Colony Optimization (ACO) algorithm has been successfully applied for finding evacuation routes during a tsunami (Forcael et al., 2014). This algorithm is an ACO heuristic for the MOSP problem. Such heuristics are important, because standard MOSP problems are computationally harder than ones with a single objective. Some surveys in this area are worth mentioning. Pangilinan and Janssens (2007) gave an overview of the MOSP problems and a review of essential issues for their solution. Skriver (2000) and Ulungu and Teghem (1991) reviewed the existing literature on MOSP problems.

## 1.2 The AHP approach for ranking alternatives

An important integrated part of the presented algorithms is the AHP, which is one of the best multi-attribute rating techniques developed by Thomas Saaty (1980, 1990, 1999) for the United States government. The AHP is an eigenvalue approach designed to handle decisions with multiple attributes, and it has been applied in a wide range of areas. A survey of the AHP, its applications, and interesting facts were given by Saaty (1999), Zahedi (1986), and Foreman (1993). The most recent application of the AHP was developed by Turskis et al. (2016) for selecting the foundation type for a single-storey dwelling house. Another interesting application of the fuzzy AHP to tunnel health evaluation was proposed by Zhang et al. (2014).

The first stage in the AHP is to set up a decision hierarchy, which is called a hierarchy tree. This means that the decision problem is broken down into hierarchies of its decision elements. At the root of the tree is the most general objective of the problem, then all the relevant attributes are arranged at the first hierarchy level. At the next level of the hierarchy tree, some of the attributes might be broken down into more detail and so forth. Finally, at the lowest level in the hierarchy tree, the available alternatives are set out.

At the second stage, each attribute is compared in turn with every other attribute at the same level of the hierarchy. For each comparison, a decision maker has to determine to what extent one attribute is more (or less) important than the other attribute. When making such a decision, the following ratings are used (see Saaty, 1980): 1 (equally important), 3 (weakly more important), 5 (strongly more important), 7 (very strongly more important), 9 (extremely more important). For

example, if attribute X is weakly more important than attribute Y, then X is 3 times more important than Y. Therefore, the latter attribute is only 1/3 as important as the former attribute, that is, 3 times less important. Ratings between the above numbers are allowed; hence one may use direct numerical inputs on the scale from 1/9 (extremely less important) to 9 (extremely more important). Thus, for a given hierarchy of size  $n$ , one can construct a reciprocal comparison  $n \times n$  matrix  $A$ , where its elements have the property  $A_{i,j} = 1/A_{j,i}$  and the main diagonal consists of 1's. The process of pairwise comparisons should be carried out for each hierarchy. Finally, the available alternatives are pairwise compared with respect to the attributes in the levels above.

The third stage is to transform the comparison matrices into sets of weights representing the relative importance of all the attributes (or alternatives) at the same hierarchy level. For an  $n \times n$  comparison matrix  $A$  of a particular hierarchy, let  $\lambda_{\max}$  denote the largest eigenvalue and  $w$  be the corresponding normalized eigenvector, that is,  $Aw = \lambda_{\max} w$ . The components of the vector  $w = (w_1, w_2, \dots, w_n)$  are the weights of the attributes (or alternatives) of that hierarchy, so that the first attribute (alternative) has weight  $w_1$ , the second  $w_2$ , and so on. Next, the Consistency Index (CI) of  $A$  is calculated as follows:  $CI = \frac{\lambda_{\max} - n}{n - 1}$ . The CI is then normalized by the Random Index (RI), which is the average CI over random entries of reciprocal matrices of the same order. The accurate list of RIs for  $n \leq 100$  was given by Donegan and Dodd (1991). Thus, the Consistency Ratio (CR) is calculated as follows:  $CR = \frac{CI}{RI}$ . The comparison matrix is considered as consistent if  $CR < 0.05$  for  $n = 3$ ,  $CR < 0.09$  for  $n = 4$ , and  $CR < 0.1$  for  $n > 4$ . An inconsistent matrix should be reconsidered.

The final stage of the AHP is to aggregate the weights and compare the alternative options. To find the aggregate score of an alternative, all paths going from the root of the tree to this particular alternative are first identified. For each path, all weights along that path are multiplied together. Then the resulting numbers are summed for all the above paths. Thus, each of the alternatives will be given an aggregate score between 0 and 1, to signify the priority of that alternative with respect to other alternatives.

## 2 ALGORITHM FOR PRIORITIZATION OF INDOOR ROUTES (PIR-ALGORITHM)

We start this section with problem definition. Suppose that there is a building, where an extreme event with many epicenters is occurring. As defined by Zverovich et al. (2016), the location of a bomb or a terrorist is an "epicenter," whereas for fire the "epicenters" can

be defined as points (nodes) in the building, where the temperature exceeds a certain threshold. Note that a large extreme event would be typically represented by several epicenters, which are denoted by  $z_1, z_2, \dots, z_k$ .

Further, assume that an agent has to go from point  $p$  to point  $q$  in the building. For example, using one of the exits as an entrance point, an SAR team must find the “best” route to trapped people in the building, and then find a route back to one of the exits, which might be different to the first route. Taking into account that there is hazard in the building, such routes must be reasonably safe, simple, and short/fast; they will be called “*optimal routes*.” Thus, three criteria should be considered for a route: hazard proximity, route complexity, and distance. Note that the distance criterion may be replaced by travel time if information about the distribution of people in the building is available. The agent must be given an option to choose a right balance of the aforementioned criteria. For instance, the agent might have personal protective equipment, so that the hazard proximity becomes least important; another user might want to minimize the route complexity, thus making this criterion most important. The algorithm for finding the optimal routes must possess the aforementioned properties. Moreover, it must be robust and efficient, and produce a reliable result.

In what follows, we will exploit the 3D building model recently developed by Boguslawski et al. (2015). An original BIM model was created in Autodesk Revit and exported to a surface representation stored in the gbXML format. Subsequently, a graph representation was reconstructed based on the model geometry, topology, and semantical information. Topological relationships and information about doors between adjacent rooms were used to generate a logical network. In this network, denoted by  $G$ , nodes represent spaces (e.g., rooms), whereas links represent adjacency relationship among those spaces. Some of the links are marked as navigable if two adjacent spaces have a door between them. In addition, links between staircase nodes are considered for navigation, which automatically allow for navigation between the building floors. Afterwards, more detailed navigable networks are generated for specific spaces with a complex geometry and several doors, for example, corridors (Boguslawski et al., 2016). The navigable networks together with navigable links from the logical network  $G$  are used to form the unified navigable network  $G^*$ , where special links representing doors and staircases are introduced. A movement from one space to another can be detected when one of the special links is used. In other words, nodes belonging to an individual space are linked with nodes from another space by special links. An example of the unified network  $G^*$  will be given in Section 3.

The input of PIR-Algorithm consists of the aforementioned graphs  $G$  and  $G^*$ , and the epicenters of an extreme event  $z_1, z_2, \dots, z_k$ . The input also includes the start node  $p$  and the destination node  $q$  of the required  $(p, q)$ -route. Note that  $p$  may be the artificial node outside the building connected to all exits for finding a route going from one of the exits to the specified location  $q$ . Alternatively, the node  $p$  may be one of the locations in the building, in which case the destination node  $q$  is optional. If  $q$  is not specified in this case, then by default it is one of the exits.

The optional maximal propagation coefficient  $\rho_{\max}$  is also a part of the input, by default  $\rho_{\max} = 100$ . The propagation coefficient  $\rho$  represents the degree of hazard spread through a building, and this spread is reflected in hazard proximity numbers defined in Step 3(a) of PIR-algorithm. For instance, if  $\rho$  is high, then hazard proximity numbers for nodes propagate quickly from 100 (in the epicenter) to small positive numbers (far from the epicenter). This puts a strong emphasis on the epicenter and the rooms in its close proximity. In contrast, if  $\rho$  is a small positive number, then the propagation is slow, thus putting less emphasis on the epicenter and the nearby rooms. In the extreme case  $\rho = 0$  there is no propagation of hazard, that is, all hazard proximity numbers for nodes are equal.

Finally, a user can optionally choose the preferential ranking of the three criteria: distance (D), hazard proximity (HP), and route complexity (RC). By default,  $HP > D > RC$ , that is, HP is the most important criterion, D is the second most important, and RC is the least important. For an advanced user, who might have personal protective equipment and/or wish to minimize the route complexity, seven options are available:

$D > HP > RC$ ,  $D > RC > HP$ ,  $HP > D > RC$ ,  $HP > RC > D$ ,  
 $RC > D > HP$ ,  $RC > HP > D$ ,  $D = HP = RC$

In general, PIR-Algorithm has two parts. In the first part (Steps 1–5), a set  $R$  of feasible  $(p, q)$ -routes is generated. This set includes three “extreme” routes: the shortest, the safest and the simplest routes, as well as a number of routes where the aforementioned criteria (D, HP, RC) are taken into account with different degrees of importance. It may be pointed out that the set  $R$  typically consists of all “reasonable”  $(p, q)$ -routes, which is achieved by two binary searches. The second part of PIR-Algorithm (Steps 6–12) constitutes a stochastic version of the AHP. Using statistical and quantitative characteristics of the routes from the set  $R$ , the AHP prioritizes the routes with respect to the specified preferential ranking of the criteria. Thus, the algorithm finds the best  $(p, q)$ -route, the second best  $(p, q)$ -route, etc.

For convenience, we summarize the notation used in PIR-Algorithm:

$AS(P_j)$	Aggregate score for route $P_j$
$C_j$	Complexity of route $P_j$
$C(e)$	Complexity of link $e$
$D(e)$	Length of link $e$ in meters
$D_j$	Length of $P_j$ in meters
$E(G)$	Set of links in $G$
$G$	Graph of connections
$G^*$	Unified network
$H(v)$	Hazard proximity number for node $v$
$H_i(v)$	Hazard proximity number for node $v$ w.r.t. $z_i$
$H(e)$	Pure hazard proximity number for link $e$
$HD(e)$	Hazard proximity number for link $e$
$M_i$	Comparison matrix for $i$ -th hierarchy
$P_\rho$	$(p,q)$ -route for the propagation coefficient $\rho$ based on distance/hazard proximity
$P'_\rho$	$(p,q)$ -route for the propagation coefficient $\rho$ based on complexity/hazard proximity
$PI_j$	Proximity index for $P_j$
$R$	Set of $(p,q)$ -routes
$b(v, z_i)$	Minimum number of obstructions (i.e., walls, floors, ceilings) between $v$ and $z_i$
$d(v, z_i)$	Direct distance from $v$ to $z_i$ in meters
$l(P_j)$	Number of links in $P_j$
$n$	Number of routes in the set $R$
$r(e)$	Proximity ratio for link $e$
$r_i(e)$	Proximity ratio for link $e$ w.r.t. $z_i$
$w_i$	Normalized eigenvector corresponding to $\lambda_i$
$z_i$	$i$ -th epicenter
$\lambda_i$	Largest eigenvalue of $M_i$
$\mu_X$	Mean of parameter $X$
$\rho$	Propagation coefficient
$\sigma_X$	Sample standard deviation of parameter $X$
$\rho_{\max}$	Maximal propagation coefficient

**PIR-Algorithm:** Prioritization of  $(p,q)$ -routes in a building, where an extreme event is occurring

#### MODULE 1

**Input:** The graphs  $G$  and  $G^*$ , which constitute the 3D model of the building.

The epicenters of an extreme event (nodes  $z_i$ ,  $i = 1, 2, \dots, k$ ).

Node  $p$ ; node  $q$  (optional;  $q$  is one of the exits by default).

The maximal propagation coefficient  $\rho_{\max}$  (optional, by default  $\rho_{\max} = 100$ ).

Preferential ranking of distance (D), hazard proximity (HP), and route complexity (RC) (optional, by default  $HP > D > RC$ , that is, HP is the most important criterion and D is the second most important one).

**Output:** Optimal  $(p,q)$ -route and information about its parameters.

- (1) Calculate  $d(v, z_i)$  and  $b(v, z_i)$  for all  $v \in V(G^*)$  and  $i = 1, 2, \dots, k$  by repeating the following steps for each node  $z_i$ :
  - (a) For each node  $v \in V(G^*)$ , calculate  $d(v, z_i)$ , the direct distance from  $v$  to  $z_i$  in meters.
  - (b) Run the Breadth First Search Algorithm in the graph  $G$  from node  $z_i$ . It returns the number of links in shortest  $(v, z_i)$ -paths in  $G$  for all nodes  $v$ , that is, the minimum number of obstructions  $b(v, z_i)$  between  $v$  and  $z_i$  for all nodes  $v \in V(G)$ .
  - (c) For each node  $v \in V(G^*) - V(G)$ , put  $b(v, z_i) = b(w, z_i)$ , where  $w \in V(G)$  represents the cell whose tessellation contains  $v$ .
- (2) Put  $R = \emptyset$ , where  $R$  is a set of  $(p,q)$ -routes.
- (3) Carry out the binary search with respect to  $\rho$ ,  $0 \leq \rho \leq \rho_{\max}$ , starting with  $\rho = 0, \rho_{\max}, 0.5\rho_{\max}$ , etc.

For each value of  $\rho$ , implement the following:

- (a) Compute the hazard proximity numbers for nodes:

$$H_i(v) = \frac{100}{\left(1 + \frac{\rho}{100}\right) \sqrt{d(v, z_i) \times [1 + b(v, z_i)]}}$$

for each node  $v \in V(G^*)$  and each  $i = 1, 2, \dots, k$ . Then calculate  $H(v) = \max_{1 \leq i \leq k} H_i(v)$  for each node  $v \in V(G^*)$ .

- (b) For each link  $e = uv \in E(G^*)$ , compute the hazard proximity numbers for links:

$$HD(e) = 0.5 [H(u) + H(v)] \times D(e)$$

where  $D(e)$  is the length of  $e$  in meters.

- (c) Run Dijkstra's algorithm in the graph  $G^*$  from node  $p$  with link weights  $HD(e)$ . It produces the  $(p,q)$ -route  $P_\rho$  corresponding to the propagation coefficient  $\rho$ .
- (d) Put  $R = R \cup \{P_\rho\}$  if  $P_\rho \notin R$ .  
Go to Step 4 if at least one of stopping criteria is satisfied (a specified number of generated routes, length of the widest interval, and a running time).
- (4) Carry out the binary search with respect to  $\rho$ ,  $0 \leq \rho \leq \rho_{\max}$ , starting with  $\rho = 0, \rho_{\max}, 0.5\rho_{\max}$ , etc.  
For each value of  $\rho$ , implement the following:
  - (a) Compute the hazard proximity numbers for nodes:

$$H_i(v) = \frac{100}{\left(1 + \frac{\rho}{100}\right) \sqrt{d(v, z_i) \times [1 + b(v, z_i)]}}$$

for each node  $v \in V(G^*)$  and each  $i = 1, 2, \dots, k$ . Then calculate  $H(v) = \max_{1 \leq i \leq k} H_i(v)$  for each node  $v \in V(G^*)$ .



- (b) For each link  $e = uv \in E(G^*)$ , compute the pure hazard proximity numbers for links:

$$H(e) = 0.5[H(u) + H(v)]$$

- (c) Run Duckham–Kulik’s adapted algorithm in the graph  $G^*$  from node  $p$  with link weights  $H(e) \times C(e)$ . Here  $C(e)$  is the complexity of link  $e$ , that is,

$C(e) = 0.3922 + 0.0049D(e)$  if  $e$  is a staircase link going up,

$C(e) = 0.3137 + 0.0049D(e)$  if  $e$  is a staircase link going down,

$C(e) = 0.1961 + 0.0049D(e)$  if  $e$  goes through a door, otherwise

$C(e) = 0.0490\alpha + 0.0049D(e)$  where  $\alpha$  is the angle (in radians) between  $e$  and the “previous” link  $c$  if  $e$  and  $c$  are considered as vectors.

The algorithm produces the  $(p,q)$ -route  $P'_\rho$  corresponding to the propagation coefficient  $\rho$ .

- (d) Put  $R = R \cup \{P'_\rho\}$  if  $P'_\rho \notin R$ .

Go to Step 5 (Module 2) if at least one of stopping criteria is satisfied (a specified number of generated routes, length of the widest interval, and a running time).

## 2.1 The first part of PIR-Algorithm — binary searches

In the first step of PIR-Algorithm, the direct distances  $d(v, z_i)$  and the number of obstructions  $b(v, z_i)$  are calculated for all nodes  $v$  in  $G^*$  and all epicenters  $z_i$ . The set  $R$  of feasible  $(p,q)$ -routes is initialized in Step 2. Then, in Step 3, the binary search is carried out with respect to the propagation coefficient  $\rho$ , taking into account distance and hazard proximity, which are reflected in the hazard proximity numbers  $HD(e)$ . The first run is for  $\rho = 0$ , producing the shortest  $(p,q)$ -route  $P_0$  because all hazard proximity numbers for links are  $100D(e)$ . The route  $P_0$  is included in the set  $R$ . The next run is for  $\rho = \rho_{\max}$ . If the resulting route coincides with  $P_0$ , then there is no interval for the binary search and it is terminated. Otherwise, the route is different from  $P_0$  and it is included in  $R$ . The next run is for  $\rho = 0.5\rho_{\max}$ . There are three possibilities here. If the resulting route is a new one, then it is included in  $R$  and the binary search continues for two intervals  $(0; 0.5\rho_{\max})$  and  $(0.5\rho_{\max}; \rho_{\max})$ . If the resulting route coincides with one of the routes in the set  $R$ , then one of the intervals is removed from the search and the other interval is used in the binary search. For example, if the route coincides with  $P_0$ , then the binary search continues for the interval  $(0.5\rho_{\max}; \rho_{\max})$ , whereas the interval  $(0; 0.5\rho_{\max})$  is removed. This procedure is terminated if at least one of stopping criteria is satisfied: a specified

number of generated routes included in  $R$ , a specified length of the widest interval, and a running time.

The hazard proximity numbers for nodes and links in Steps 3(a) and 3(b) were introduced by Zverovich et al. (2016), where they are explained and validated. However, a slightly different formula for  $H_i(v)$  was used in their article:

$$H_i(v) = \frac{100}{(1 + \rho/100)\sqrt{d(v, z_i) \times b(v, z_i)}}$$

which is not applicable for large open spaces. Indeed, if the epicenter  $z_i$  of hazard is located in a large open space  $L$ , then the number of obstructions  $b(v, z_i)$  for all nodes  $v$  in  $L$  is equal to zero, which means  $H_i(v) = 100$  for all nodes in  $L$  and for any value of  $\rho$ . Thus, in this particular case, we have a non-discrimination problem, that is, the above formula does not distinguish between nodes in close proximity to the hazard and nodes which are further away. Actually, in a large open space with an epicenter, the distance should be used as a criterion of hazard proximity within this space. This can be achieved if the term  $b(v, z_i)$  is replaced by  $[1+b(v, z_i)]$  in the aforementioned formula. Now, if  $b(v, z_i) = 0$ , then the hazard proximity number  $H_i(v)$  does depend on the distance, and  $H_i(v) = 100$  only if the node  $v$  coincides with the epicenter  $z_i$  or  $\rho = 0$ . Thus, the updated formula for  $H_i(v)$  in Step 3(a) extends the previous one by improving hazard propagation in large open spaces. The formula for the proximity ratios w.r.t.  $z_i$  in Step 6(b) was updated in a similar way.

The second binary search (Step 4) of the algorithm is carried out with respect to  $\rho$ , taking into account link complexity and hazard proximity. Basically, this step is similar to the previous one, however, instead of distances, link complexities are used. The main difference is that some link complexities cannot be calculated in advance, for example the angle between two links is not a property of a given link but rather a pair of adjacent links. Hence, Dijkstra’s algorithm cannot be directly used here. Instead, we apply the simplest path algorithm by Duckham and Kulik (2003). In their algorithm, one can use any weighting function

$f : \mathcal{E} \rightarrow \mathbb{R}^+$ , where

$$\mathcal{E} = \{(ab, bc) \in (E(G^*) \cup \text{nil}) \times E(G^*)\}$$

and  $E(G^*)$  is the set of links in  $G^*$ . This function is defined below, and it is based on specific weights for the complexity attributes. Let us label the five complexity attributes as follows:

- A1: Staircase link going up;
- A2: Staircase link going down;
- A3: Door;

A4: Turn of 1 radian;  
 A5: Distance (10 meters).

Although the first four attributes are obviously elements of complexity, the inclusion of the distance attribute as a complexity criterion should be explained. Let us consider two straight routes inside a corridor without doors, say 5 meters long and 50 meters long. The first four attributes add no complexity to the routes, so without the distance attribute it is impossible to distinguish between the two routes from the viewpoint of their complexities. This would contradict to common sense: the former route is obviously “simpler” than the latter, which is only reflected in their lengths. For assigning appropriate weights to the attributes, we apply one particular step from the AHP (see Saaty, 1980): first, construct a comparison matrix, where the rows and columns correspond to the attributes A1–A5:

$$\begin{matrix}
 & \begin{matrix} A1 & A2 & A3 & A4 & A5 \end{matrix} \\
 \begin{matrix} A1 \\ A2 \\ A3 \\ A4 \\ A5 \end{matrix} & \begin{pmatrix} 1 & 1.25 & 2 & 8 & 8 \\ 1/1.25 & 1 & 1.6 & 6.4 & 6.4 \\ 1/2 & 1/1.6 & 1 & 4 & 4 \\ 1/8 & 1/6.4 & 1/4 & 1 & 1 \\ 1/8 & 1/6.4 & 1/4 & 1 & 1 \end{pmatrix}
 \end{matrix}$$

The entries in this matrix reflect the relative importance of the attributes according to the rules of the AHP. For instance, the entry “8” means that A1 is nearly extremely more important than A4. Further, the largest eigenvalue is 5 and the corresponding normalized eigenvector is (0.3922; 0.3137; 0.1961; 0.0490; 0.0490). This eigenvector provides the required weights for the attributes, whereas the Consistency Ratio of 0 indicates that the comparison matrix is perfectly consistent. Thus, the complexity attributes are assigned the following weights:

- A1: 0.3922 (staircase link going up);
- A2: 0.3137 (staircase link going down);
- A3: 0.1961(door);
- A4: 0.0490 (turn of 1 radian);
- A5: 0.0049 (distance of 1 meter).

Now, the aforementioned function  $f$  is defined as follows:

$$f(ab, bc) = [0.3922 + 0.0049D(bc)]H(bc) \text{ if } bc \text{ is a staircase link going up (} ab \text{ may be “nil”),}$$

$$f(ab, bc) = [0.3137 + 0.0049D(bc)]H(bc) \text{ if } bc \text{ is a staircase link going down (} ab \text{ may be “nil”),}$$

$$f(ab, bc) = [0.1961 + 0.0049D(bc)]H(bc) \text{ if } bc \text{ is a link going through a door (} ab \text{ may be “nil”),}$$

$$f(ab, bc) = [0.0490\alpha + 0.0049D(bc)]H(bc) \text{ where } \alpha \text{ is the angle (in radians) between the non-staircase links } ab \text{ and } bc, ab \neq \text{nil, and } bc \text{ is not a door link.}$$

Note that in the first three cases it is allowed to have  $ab = \text{nil}$ . Hence, if the first link  $e$  in a route is a door or a staircase link, then the value of the function  $f(\text{nil}, e)$  is calculated for such a link. It may be pointed out that the first run of the binary search for  $\rho = 0$  will produce the simplest  $(p, q)$ -route, because in this case  $H(e) = 100$  for any link  $e$ . For other values of  $\rho$ , both link complexities and hazard proximities are taken into account when generating the corresponding  $(p, q)$ -routes, which is reflected in the definition of the function  $f$ . For instance, the value of the function  $f(ab, bc)$  for a door  $bc$  increases if it is closer to the epicenter of hazard because  $H(bc)$  will be larger. This binary search is terminated if at least one of stopping criteria is satisfied: a specified number of generated routes included in  $R$ , a specified length of the widest interval, and a running time.

## 2.2 The second part of PIR-Algorithm — the AHP

In Step 5, all the routes in the set  $R$  are denoted by  $P_1, P_2, \dots, P_n$  for simplicity of presentation. These routes form the input for the AHP. Next, in Step 6, the following parameters are calculated for each route in  $R$ : the route length, the proximity ratios, the proximity index, and the route complexity. The calculation of complexity is consistent with that of Step 4(c). The proximity ratios and indices are similar to those used in Zverovich et al. (2016) with the only difference in the term  $[1+b(v, z_i)]$ , which is needed to avoid the aforementioned problem of non-discrimination in large open spaces. The necessary statistical characteristics are computed in Step 7.

**PIR-Algorithm:** Prioritization of  $(p, q)$ -routes in a building, where an extreme event is occurring

### MODULE 2

- (5) Denote all the routes in  $R$  by  $P_1, P_2, \dots, P_n$ .
- (6) For each route  $P_j \in R$ , calculate the following:
  - (a) The length of  $P_j$ :  $D_j = \sum_{e \in P_j} D(e)$
  - (b) The proximity ratios w.r.t.  $z_i$ :

$$r_i(e) = \frac{\sqrt{d(u, z_i) \times [1 + b(u, z_i)]} + \sqrt{d(v, z_i) \times [1 + b(v, z_i)]}}{2D(e)}$$

for each  $i = 1, 2, \dots, k$  and each link  $e = uv$  in  $P_j$ .

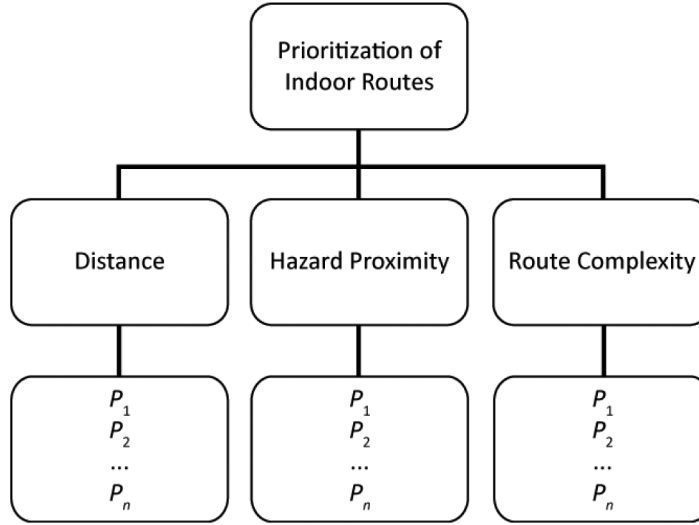


Fig. 1. Hierarchy tree for the AHP.

- (c) The proximity ratios  $r(e) = \min_{1 \leq i \leq k} r_i(e)$  for each link  $e = uv$  in  $P_j$ .
- (d) The proximity index  $PI_j = \frac{l(P_j)}{\sum_{e \in P_j} \frac{1}{r(e)}}$   
 $PI$  is the harmonic mean of  $r(e)$ 's, and  $l(P_j)$  is the number of links in  $P_j$ .
- (e) The complexity of  $P_j$ :  $C_j = 0.3922\Omega_j + 0.3137\omega_j + 0.1961\delta_j + 0.0490\chi_j + 0.0049D_j$ , where  $\Omega_j$  is the number of staircase links going up in  $P_j$ ,  $\omega_j$  is the number of staircase links going down,  $\delta_j$  is the number of doors,  $\chi_j$  is the total turning angle (in radians), and  $D_j$  is the length of  $P_j$  (in meters).
- (7) For the routes in  $R$ , compute the means and the sample standard deviations of distances, proximity indices and complexities:  $\mu_D, \sigma_D, \mu_{PI}, \sigma_{PI}, \mu_C, \sigma_C$ .
- (8) Create the  $3 \times 3$  matrix  $M_1$  of relative importance of distance (D), hazard proximity (HP), and route complexity (RC), based on the specified preferential ranking. The row and columns of  $M_1$  correspond to D, HP, and RC, respectively. By default,  $HP > D > RC$ , that is,

$$M_1 = \begin{pmatrix} 1 & 1/2 & 2 \\ 2 & 1 & 4 \\ 1/2 & 1/4 & 1 \end{pmatrix}$$

- (9) Create three  $n \times n$  comparison matrices  $M_2, M_3, M_4$  for the relative importance of the routes in the set  $R$  w.r.t. distance, proximity index and complexity as follows:

For each entry  $M_2[i, j]$ , compute  $\beta = (D_j - D_i)/\sigma_D$ . If  $\sigma_D = 0$ , then put  $\beta = 0$ . Calculate  $M_2[i, j] = 9^{0.2\beta}$ . Put  $M_2[i, j] = 1/9$  if  $\beta < -5$ ; put  $M_2[i, j] = 9$  if  $\beta > 5$ .

For each entry  $M_3[i, j]$ , compute  $\beta = (PI_i - PI_j)/\sigma_{PI}$ . If  $\sigma_{PI} = 0$ , then put  $\beta = 0$ . Calculate  $M_3[i, j] = 9^{0.2\beta}$ . Put  $M_3[i, j] = 1/9$  if  $\beta < -5$ ; put  $M_3[i, j] = 9$  if  $\beta > 5$ .

For each entry  $M_4[i, j]$ , compute  $\beta = (C_j - C_i)/\sigma_C$ . If  $\sigma_C = 0$ , then put  $\beta = 0$ . Calculate  $M_4[i, j] = 9^{0.2\beta}$ . Put  $M_4[i, j] = 1/9$  if  $\beta < -5$ ; put  $M_4[i, j] = 9$  if  $\beta > 5$ .

- (10) For each matrix  $M_i$ , run the Direct Iteration Algorithm to calculate its largest eigenvalue  $\lambda_i$  and the corresponding normalized eigenvector  $w_i$  with precision 0.0001.
- (11) For each route  $P_j \in R$ , calculate the aggregate score:

$$AS(P_j) = w_1[1]w_2[j] + w_1[2]w_3[j] + w_1[3]w_4[j]$$

- (12) Determine the route  $P_j$  with the highest aggregate score. Report  $P_j$  and information about its parameters. Algorithm stops.

The initial stage of the AHP (see Saaty, 1980) is to set up a hierarchy tree, which is shown in Figure 1. The first level of hierarchy in the tree consists of three criteria: distance (D), hazard proximity (HP), and route complexity (RC), whereas the lowest three levels comprise the routes in the set  $R$ . In Step 8 of PIR-Algorithm, the comparison matrix  $M_1$  is created, which is based on the specified preferential ranking of the criteria. The row and columns of  $M_1$  correspond to D, HP, and RC, respectively. By default,  $HP > D > RC$ , that is,

$$M_1 = \begin{pmatrix} 1 & 1/2 & 2 \\ 2 & 1 & 4 \\ 1/2 & 1/4 & 1 \end{pmatrix}$$



In general, there are 7 possibilities for the preferential ranking:  $D>HP>RC$ ,  $D>RC>HP$ ,  $HP>D>RC$ ,  $HP>RC>D$ ,  $RC>D>HP$ ,  $RC>HP>D$ , and  $D=HP=RC$ . For example, if  $D>HP>RC$ , then

$$M_1 = \begin{pmatrix} 1 & 2 & 4 \\ 1/2 & 1 & 2 \\ 1/4 & 1/2 & 1 \end{pmatrix};$$

if  $D=HP=RC$ , then  $M_1$  consists of 1's. Notice that the matrix  $M_1$  is always perfectly consistent.

Three  $n \times n$  comparison matrices  $M_2$ ,  $M_3$ , and  $M_4$  are constructed in Step 9. They represent the relative importance of the routes in the set  $R$  w.r.t. distance, proximity index and complexity, respectively, and the construction of the matrices is based on the corresponding parameters and standard deviations. For instance, the  $(i, j)$ -element of the matrix  $M_2$  is calculated as follows: we first compute  $\beta = (D_j - D_i)/\sigma_D$ , where  $D_j$  and  $D_i$  are the lengths of the routes  $P_j$  and  $P_i$  in meters; if  $\sigma_D = 0$ , then we put  $\beta = 0$ . Next,  $M_2[i, j] = 1/9$  if  $\beta < -5$ ,  $M_2[i, j] = 9$  if  $\beta > 5$ , and  $M_2[i, j] = 9^{0.2\beta}$  otherwise. Thus, the  $(i, j)$ -element of the matrix  $M_2$  represents the relative importance of the routes  $P_i$  and  $P_j$  in terms of standard deviations between their lengths. For example, if the difference in lengths is 2.5 standard deviations, then one route is weakly more important than the other, which will be reflected by the entry "3" in the comparison matrix. It may be pointed out that different functions for constructing the above matrices have been tested, and it turned out that the best one is the exponential function  $9^{0.2\beta}$ . Also, because of the way the matrices  $M_2$ ,  $M_3$ , and  $M_4$  are constructed, the corresponding Consistency Indices and Consistency Ratios are always very small, or equal to zero. Hence, the matrices are very consistent and there is no need to adjust them.

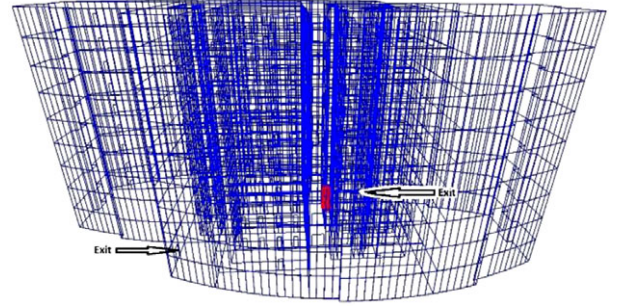
In Step 10, the standard Direct Iteration Algorithm is used to calculate the largest eigenvalue  $\lambda_i$  for each matrix  $M_i$  and the corresponding normalized eigenvector  $w_i$  with precision 0.0001. For each route  $P_j \in R$ , the aggregate score is computed in Step 11 as follows:

$$AS(P_j) = w_1[1] w_2[j] + w_1[2] w_3[j] + w_1[3] w_4[j]$$

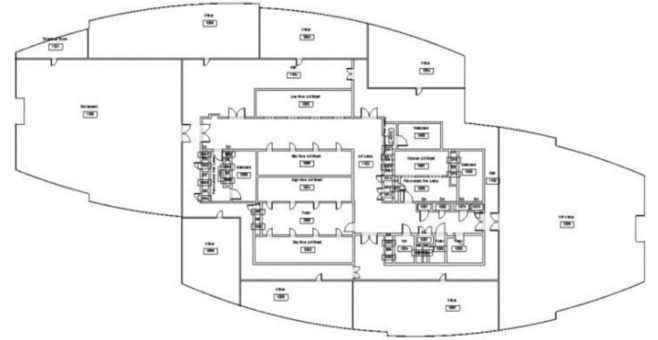
Finally, in Step 12, the route  $P_j$  with the highest aggregate score is determined and reported, together with the information about the parameters of this route.

### 3 TESTING PIR-ALGORITHM

We start testing PIR-Algorithm with a rather complex building shown in Figure 2. This 9-floor building has 5 stairwells and two exits (indicated by arrows in Figures 2 and 4), and it is based on the typical floor of the Doha



**Fig. 2.** The 9-floor building model based on the typical floor of the DWTC building.



**Fig. 3.** First floor of the DWTC building.

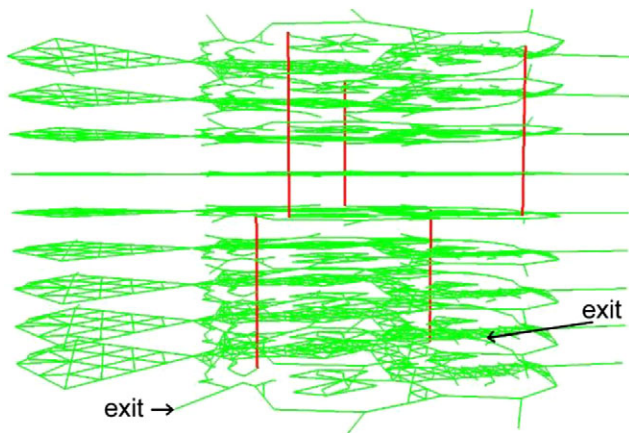
World Trade Centre (DWTC) depicted in Figure 3. All the stairwells are highlighted in red in Figure 4, which represents the navigable unified network  $G^*$  of the building.

The epicenter of an extreme event is on the ground floor, and it is labeled by a red rectangle in Figures 2 and 5. The start node  $p$  is the artificial node outside the building (not shown in the unified network), and the destination node  $q$  represents a large room located on the last floor in the right part of the building. Thus, we are looking for the optimal route going from one of the exits to the room  $q$ . In what follows, we put  $\rho_{\max} = 100$  for one epicenter, because our numerous tests showed that "unreasonable" routes are often produced for  $\rho > 100$ , which do not belong to the efficient frontier. Also, the default preferential ranking of the criteria is used, that is,  $HP>D>RC$ .

The first binary search in Step 3 of PIR-Algorithm produces six routes  $P_1-P_6$ , whereas the second binary search in Step 4 also generates six routes  $P_7-P_{12}$ . These routes are depicted in Figure 5, and their parameters are summarized in Table 1. Note that in both binary searches one stopping criterion was used: the length of the widest interval is 0.01. It may be pointed out that different exits are used as an entrance point in the generated routes.

**Table 1**  
Parameters of routes  $P_1$ – $P_{12}$

Route	Propagation coefficient $\rho$	Length $D_j$	Proximity Index $PI_j$	Complexity $C_j$	Aggregate Score $AS(P_j)$
1 <sup>st</sup> binary search					
$P_1$	0	154.61	1.89	6.67	0.085
$P_2$	42.481	222.79	4.29	9.28	0.090
$P_3$	15.478	194.88	3.76	7.47	0.089
$P_4$	7.190	183.30	3.79	7.98	0.092
$P_5$	29.576	222.58	4.24	9.00	0.088
$P_6$	26.485	196.45	3.50	7.83	0.080
2 <sup>nd</sup> binary search					
$P_7$	0	168.65	1.69	6.37	0.074
$P_8$	14.451	212.40	3.56	7.73	0.076
$P_9$	8.632	186.42	3.96	7.31	0.099
$P_{10}$	5.660	199.07	3.12	6.99	0.075
$P_{11}$	3.284	159.43	1.82	6.38	0.082
$P_{12}$	5.602	208.84	3.00	7.02	0.069



**Fig. 4.** The unified network  $G^s$  of the model.

As can be seen in Table 1, the shortest route  $P_1$  corresponds to the propagation coefficient  $\rho = 0$  in the first binary search. The safest route  $P_2$  has the largest proximity index 4.29 (2dp) and the corresponding propagation coefficient is 42.481, which is given to 3dp and rounded up to keep the correspondence to  $P_2$ . More precisely, the route  $P_2$  would be generated for all values of  $\rho$  between 42.481 and 100, but the lowest value is only reported in the table. Thus, formally speaking, the route  $P_2$  corresponds to the interval [42.481, 100] of propagation coefficients, whereas the route  $P_1$  corresponds to the interval [0, 7.190), where the number 7.190 is excluded from the interval. Note that the safest route is generated for the largest propagation coefficient in the first binary search, because maximum emphasis is put

on the epicenter of hazard in this case. Also, it may be pointed out that the safest route  $P_2$  is the longest and most complex one, which is a typical picture unless the safest route coincides with the shortest. The simplest route  $P_7$  corresponds to the propagation coefficient  $\rho = 0$  in the second binary search, which is always the case because there is no propagation of hazard for  $\rho = 0$  and only complexity of the route is minimized.

Note that the route  $P_{10}$  is better than  $P_{12}$  for all attributes, and also the routes  $P_3$ ,  $P_6$ ,  $P_8$  are “dominated” by  $P_9$ . Therefore, the efficient frontier consists of eight routes:  $P_1$ ,  $P_2$ ,  $P_4$ ,  $P_5$ ,  $P_7$ ,  $P_9$ ,  $P_{10}$ ,  $P_{11}$ . The efficient frontier is illustrated in Figure 6, where the above routes are labeled by numbers 1,2,4,5,7,9,10,11. Its lower part comprises three routes:  $P_1$ ,  $P_7$ , and  $P_{11}$ , which have very good lengths and complexities, but the proximity indices are rather low. In the upper part there are two routes:  $P_2$  and  $P_5$ , which are very long and complex, but safer than other routes. Notice that the routes  $P_2$  and  $P_5$  look similar in Figure 5, but there is some difference on the top floor. The middle part of the efficient frontier consists of three routes:  $P_4$ ,  $P_9$ , and  $P_{10}$ . The first two routes form two local maxima and have high proximity indices and very reasonable lengths and complexities; hence, based on common sense, they are good candidates for being the optimal routes for the default preferential ranking.

The optimal route for the preferential ranking  $HP > D > RC$  is  $P_9$  because it has the largest aggregate score of 0.099 (3dp). This route has the third highest proximity index (3.96), which is close to the best value (4.29), and also reasonable length (186.42 m) and complexity (7.31), which are slightly better than the

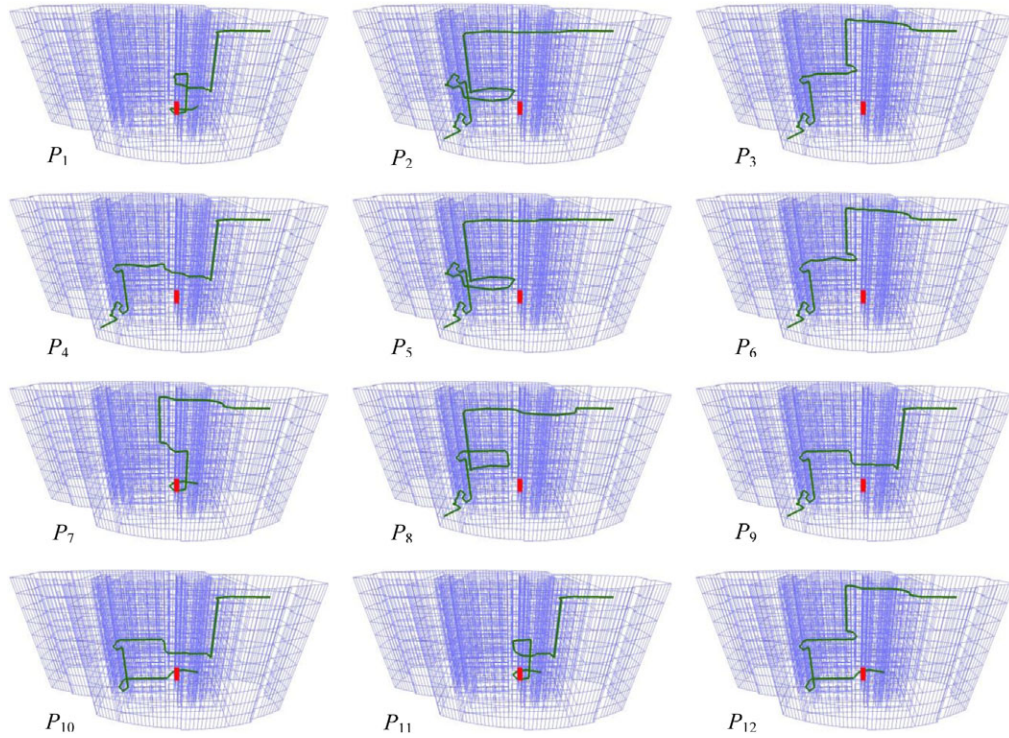


Fig. 5. The routes  $P_1$ – $P_{12}$  generated by the binary searches.

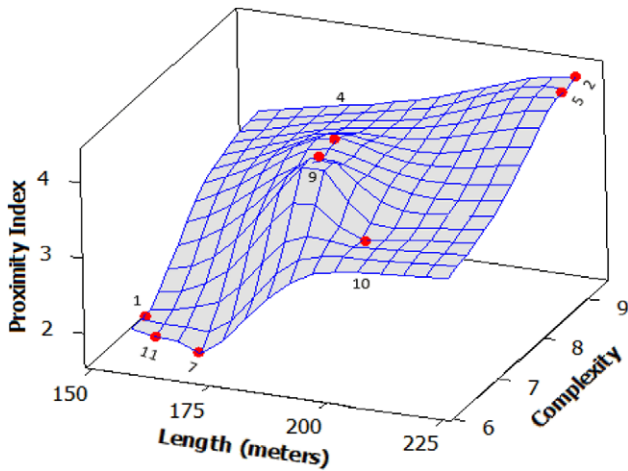


Fig. 6. The efficient frontier.

corresponding average values shown below. Here are the statistical parameters (means and sample standard deviations):

$$\mu_D = 192.45, \sigma_D = 22.90,$$

$$\mu_{PI} = 3.22, \sigma_{PI} = 0.94,$$

$$\mu_C = 7.50, \sigma_C = 0.93$$

Note that the normality test (Anderson-Darling) showed that the data sets in Table 1 displayed in the

columns “Length,” “Proximity Index,” and “Complexity” are normally distributed. For completeness, we also give the normalized eigenvectors:

$$w_1 = (0.286, 0.571, 0.143),$$

$$w_2 = (0.157, 0.043, 0.073, 0.091, 0.043, 0.071, 0.120, 0.052, 0.086, 0.067, 0.143, 0.056),$$

$$w_3 = (0.041, 0.127, 0.099, 0.101, 0.124, 0.088, 0.038, 0.090, 0.109, 0.073, 0.040, 0.070),$$

$$w_4 = (0.114, 0.033, 0.078, 0.061, 0.038, 0.066, 0.131, 0.069, 0.084, 0.098, 0.131, 0.097).$$

The eigenvectors for length ( $w_2$ ), proximity index ( $w_3$ ), and complexity ( $w_4$ ) are visualized in the bar chart of Figure 7, together with the aggregate score, which is the weighted average of the elements of  $w_2$ ,  $w_3$ , and  $w_4$  using the weights of the eigenvector  $w_1$ . Also, the largest eigenvalues of the  $12 \times 12$  comparison matrices  $M_2$ ,  $M_3$ , and  $M_4$  are 12, and hence the corresponding Consistency Indices and Consistency Ratios are equal to zero, so that the matrices are perfectly consistent.

Because the input of PIR-Algorithm includes the preferential ranking of the main criteria, we provide the optimal routes for different preferential rankings: the route  $P_1$  for  $D > HP > RC$  and  $D > RC > HP$ ; the route



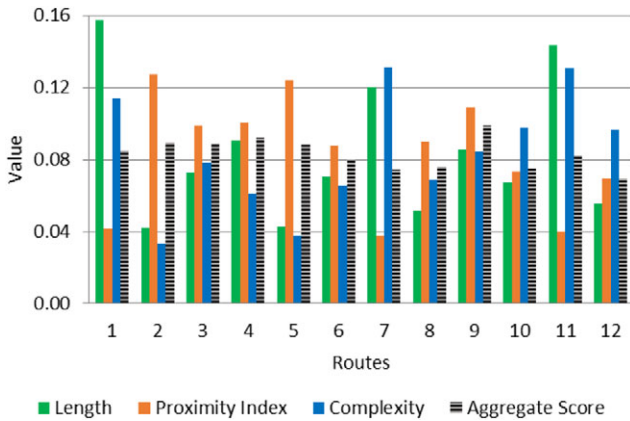


Fig. 7. The bar chart for eigenvectors and aggregate score.

$P_9$  for HP>D>RC and HP>RC>D; the route  $P_{11}$  for RC>D>HP, RC>HP>D, and D=HP=RC. Thus, if hazard proximity is not the most important attribute, then the optimal route is either  $P_1$  or  $P_{11}$ . This can be explained by the fact that  $P_1$  is the shortest route with the third best complexity (6.67), which is very close to the best complexity (6.37), so  $P_1$  is the optimal route if distance is the most important criterion. The route  $P_{11}$  exhibits the second best complexity (6.38), which is extremely close to the best value (6.37), and it is the second shortest (159.43 m), which is also very close to best length (154.61 m). Hence  $P_{11}$  is the optimal route if complexity is the most important attribute or all the criteria are of the same importance.

It may be pointed out in conclusion of this section that the proposed method is quite robust with respect to different parameters used in the algorithm, that is, the method is not sensitive to small changes in the parameters. For example, the sensitivity analysis was applied to the default matrix  $M_1$ , where two elements “2” are adjusted by at most  $\pm 0.5$  in such a way that the matrix remains perfectly consistent. More precisely, the elements  $M_1[2,1] = 2$  and  $M_1[1,3] = 2$ , representing the relative importance of HP to D and D to RC, may be any numbers in the interval [1.5, 2.5]. Next,  $M_1[1,2] = 1/M_1[2,1]$  by definition and  $M_1[1,2] \times M_1[2,3] = M_1[1,3]$  to keep the matrix perfectly consistent, that is,

$$M_1[2,3] = M_1[1,3]/M_1[1,2] = M_1[1,3] \times M_1[2,1]$$

Also,  $M_1[3,1]$  and  $M_1[3,2]$  are the reciprocals of  $M_1[1,3]$  and  $M_1[2,3]$  by definition. Thus, the entire matrix is determined by its two elements  $M_1[2,1]$  and  $M_1[1,3]$ . Now, if these elements are any numbers in the interval [1.5, 2.5], then the corresponding optimal route is  $P_9$ , that is, it remains unchanged. Similar sensitivity analysis was carried out for the weights used in the complexity function in Step 4(c) of the algorithm. It showed that small

changes in the weights do not change the set of generated routes in Step 4, and therefore the optimal route remains unchanged. The same comment can be done for the exponential function  $9^{0.2\beta}$  used in Step 9 of PIR-Algorithm.

#### 4 INCORPORATING “TIME” INTO PIR-ALGORITHM

To take into consideration the time required to travel from the start node to the destination node in a building, it is necessary to know the length and speed for all links in the route. Indeed, the travel time  $T(e)$  along link  $e$  is calculated as follows:

$$T(e) = D(e)/S(e)$$

where  $D(e)$  is the length of  $e$  in meters, and  $S(e)$  is the speed along the link  $e$  (meters per second). The most reliable formula for people’s indoor speed is Nelson–MacLennan–Pauls’ relationship between speed and density (see Nelson and MacLennan, 1995; Pauls, 1995):

$$S(e) = \begin{cases} 0.856K & \text{if } 0 \leq \Delta < 0.54 \\ (1 - 0.266 \Delta)K & \text{if } 0.54 \leq \Delta \leq 3.75 \end{cases} \quad (1)$$

where  $\Delta$  is the population density (persons per square meter) in the area corresponding to the link  $e$ , and the constant  $K$  is defined as follows:

$$\begin{aligned} K &= 1.40 \text{ for horizontal movement,} \\ K &= 1.08 \text{ for moving downstairs,} \\ K &= 0.81 \text{ for moving upstairs.} \end{aligned}$$

Notice that the value of the constant  $K$  for moving upstairs was derived from the results of Fruin (1987), whereas its value for moving downstairs depends on the characteristics of stairs such as the length of riser and tread (see Nelson and MacLennan, 1995). The fundamental formula (1) provides the linear relationship between people’s speed and density. More precisely, if the population density  $\Delta$  is less than 0.54 persons/m<sup>2</sup>, then people’s movement would be dependent on their personal characteristics, and so the average constant speed of 0.856 m/s is used. If  $\Delta > 3.75$  persons/m<sup>2</sup>, then no movement is possible until the density is reduced. Between the density values of 0.54 and 3.75 persons/m<sup>2</sup>, the speed is given by the linear function  $(1 - 0.266\Delta)K$ .

Thus, for calculation of the travel time along a given route it is necessary to know the distribution of people in the building, that is, the population densities in its areas. In some cases, such a distribution is known, and for emergency situations this can be achieved by the simulation of evacuation flows of people. The latter is

out of scope of this article; however, there are software packages for modeling the distribution of people in a building. For example, Kisko and Francis (1985) proposed the EVACNET+ software application for evacuation scenarios planning. Other alternative solutions are available as well: Sun and Wu (2014) recently developed an advanced configurable crowd model for different behaviors and scenarios, and implemented it for the simulation of evacuation in a building. Also, Shahabi et al. (2015) proposed an efficient strategy for the shortest path problem with uncertain travel cost, which might be useful when dealing with uncertain travel time function.

Therefore, in the modified version of PIR-Algorithm presented below, we assume that the population densities are given for all cells in the building. Because each cell is represented by a node in the unified network  $G^*$ , the densities are associated with nodes in the set  $V(G^*)$ . In the new Step 0, we first calculate the density for the link  $uv$ , that is, the average density for two areas associated with the nodes  $u$  and  $v$ . Then the speed  $S(e)$  along the link  $e$  is computed using formula (1). Here we assume that the agent is moving together with the flow of people. In emergency situations, this typically happens when moving downstairs. If the agent is moving in a counter-flow, for example, upstairs, then one should use formula (2), which will be discussed later. Finally, the travel time  $T(e)$  along link  $e$  is calculated in Step 0(c).

**PIR-Algorithm2** (with Time): Prioritization of  $(p,q)$ -routes in a building with an extreme event

**Input:** ...

Preferential ranking of travel time (TT), hazard proximity (HP), and route complexity (RC) (optional, by default  $HP > TT > RC$ ).

The population density  $\Delta(v)$  (persons/m<sup>2</sup>) for the area corresponding to node  $v \in V(G^*)$ .

- (0) For each link  $e = uv \in E(G^*)$ , compute the following:
- The population density for the area corresponding to  $e$ :  $\Delta(e) = 0.5[\Delta(u) + \Delta(v)]$ .
  - The speed  $S(e)$  along the link  $e$  using formula (1), or (2) for counter-flow movement.
  - The travel time  $T(e)$  along link  $e$ :  $T(e) = D(e)/S(e)$ .
- ...

- (3)(b) For each link  $e = uv \in E(G^*)$ , compute the hazard proximity numbers for links:

$$HD(e) = 0.5[H(u) + H(v)] \times T(e)$$

where  $T(e)$  is the travel time along  $e$  in seconds.

...

- (6) For each route  $P_j \in R$ , calculate the following:
- The length  $D_j = \sum_{e \in P_j} D(e)$  and the travel time

$$T_j = \sum_{e \in P_j} T(e).$$

- The proximity ratios w.r.t.  $z_i$ :

$$r_i(e) = \frac{\sqrt{d(u, z_i) \times [1 + b(u, z_i)]} + \sqrt{d(v, z_i) \times [1 + b(v, z_i)]}}{2T(e)}$$

for each  $i = 1, 2, \dots, k$  and each link  $e = uv$  in  $P_j$ .

...

- (7) For the routes in  $R$ , compute the means and the sample standard deviations of travel times, proximity indices, and complexities:  $\mu_T, \sigma_T, \mu_{PI}, \sigma_{PI}, \mu_C, \sigma_C$ .
- ...

- (9) Create three  $n \times n$  comparison matrices  $M_2, M_3, M_4$  for the relative importance of the routes in the set  $R$  w.r.t. travel time, proximity index, and complexity as follows:

For each entry  $M_2[i, j]$ , compute  $\beta = (T_j - T_i)/\sigma_T$ . If  $\sigma_T = 0$ , then put  $\beta = 0$ . Calculate  $M_2[i, j] = 9^{0.2\beta}$ . Put  $M_2[i, j] = 1/9$  if  $\beta < -5$ ; put  $M_2[i, j] = 9$  if  $\beta > 5$ .

...

Further, in Step 3(b), the formula for the hazard proximity numbers is updated by using travel time instead of link length. Thus, if there is no hazard propagation, that is,  $\rho = 0$ , then the fastest route will be generated in the first binary search instead of the shortest one. In the modified Step 6, in addition to the route length, we calculate the travel time for each route in the set  $R$ . Also, the formula for proximity ratios is updated with travel time  $T(e)$ , because longer travel time for a link should decrease the proximity ratio for this link and the proximity index for the route, thus making it worse. Next, in Step 7, the statistical parameters are now calculated for travel times, and the comparison matrix  $M_2$  in Step 9 represents the relative importance of routes in the set  $R$  with respect to travel time.

#### 4.1 Testing PIR-Algorithm2

Before we start testing PIR-Algorithm2, let us apply the original PIR-Algorithm to the scenario described in Section 3 with the start and destination nodes swapped, that is, the start node is the large room in the right part of the last floor and the destination node is the artificial node outside the building. Note that there is no



**Table 2**  
Parameters of routes  $X_1$ – $X_{11}$  for moving downstairs

Route	Propagation coefficient $\rho$	Length $D_j$	Proximity Index $PI_j$	Complexity $C_j$	Aggregate Score $AS(X_j)$
1 <sup>st</sup> binary search					
$X_1$	0	154.61	1.89	6.03	0.0908
$X_2$	42.481	222.79	4.29	8.56	0.0903
$X_3$	15.478	194.88	3.76	6.91	0.0900
$X_4$	7.190	183.30	3.79	7.35	0.0942
$X_5$	29.576	222.58	4.24	8.28	0.0893
$X_6$	26.485	196.45	3.50	7.32	0.0803
2 <sup>nd</sup> binary search					
$X_7$	0	157.66	1.79	5.68	0.0904
$X_8$	30.018	207.35	4.02	7.25	0.0909
$X_9$	10.625	185.73	3.98	6.71	0.1022
$X_{10}$	5.878	198.38	3.12	6.36	0.0776
$X_{11}$	29.236	184.44	4.01	6.69	0.1040

symmetry in moving downstairs and upstairs because the complexity of the former is lower. The result is summarized in Table 2. The first binary search produces the same routes as in Table 1, because it is independent of complexity, but the complexity figures are lower than the corresponding numbers in Table 1. The second binary search generates five routes, which are basically similar to the routes in the second half of Table 1. The optimal route is  $X_{11}$ , which is similar to the optimal route  $P_9$  for moving upstairs, depicted in Figure 5.

Now, let us consider the following four typical scenarios:

**Scenario 1:** There are no people in the building, that is, the population densities for all nodes are equal to 0.

**Scenario 2:** The initial stage of uncontrolled evacuation, when people have just left their offices. Thus,  $\Delta = 0$  for any office,  $\Delta = 1$  for all staircases and landings, and  $\Delta = 0.5$  for all other spaces (halls, corridors, etc.).

**Scenario 3:** The middle stage of uncontrolled evacuation:  $\Delta = 3$  for all staircases and landings,  $\Delta = 2$  in the areas on the ground floor between staircases and entrances to the building,  $\Delta = 2$  in the areas on Floor 4 where people switch staircases,  $\Delta = 0$  for all other spaces.

**Scenario 4:** The final stage of uncontrolled evacuation:  $\Delta = 3$  for staircases and landings between the ground floor and Floor 4,  $\Delta = 2$  in the areas on the ground floor between staircases and entrances to the building,  $\Delta = 0$  for all other spaces.

In the first scenario, we assume that there are no people inside the building, for example, during weekend or night. Notice that this scenario also covers the

sparse distribution of people ( $\Delta < 0.54$ ) when an agent is moving together with people's flow, because in this case the speed given by formula (1) is a constant and hence the travel time for a given link or route is fixed. Scenario 2 describes an initial stage of evacuation, when all offices are empty, the crowd condition in the staircases is moderate, and other spaces have the minimum crowd condition. Scenario 3 is for a middle stage of evacuation, when all staircases experience the crush crowd condition, and the medium densities are on the ground floor in the specified areas and on Floor 4 in the areas between staircases. This scenario also covers the situation when there are some queues near the entrances to the staircases on some floors, because the generated routes will remain unchanged. The final stage of evacuation is given in Scenario 4, where the staircases between the ground floor and Floor 4 have the crush crowd condition and the medium densities are on the ground floor between staircases and entrances to the building.

The results of PIR-Algorithm2 applied to Scenarios 1–4 are summarized in Table 3. Compared to the set of routes in Table 2, there is a new route of length 206.52 m for Scenario 1, two new routes (206.52 m, 206.38 m) for Scenarios 2 and 3, and two new routes (206.52 m, 183.28 m) for Scenario 4. The optimal route for all the scenarios is the same: it is the route  $X_{11}$  of length 184.44 m (it is similar to the route  $P_9$  depicted in Figure 5). This demonstrates some robustness of the algorithm and its insensitivity to different scenarios of uncontrolled evacuation, even though there is a dramatic change in travel times for Scenarios 1 and 3.

It may be pointed out that, for all the scenarios, the travel time, the proximity index and the complexity of

**Table 3**  
Results of PIR-Algorithm2 for Scenarios 1–4 (moving downstairs)

Length	Scenario 1				Scenario 2				Scenario 3				Scenario 4						
	Time	PI	C	AS	Length	Time	PI	C	AS	Length	Time	PI	C	AS	Length	Time	PI	C	AS
154.61	153	1.98	6.03	0.0908	154.61	171	1.80	6.03	0.0915	154.61	525	0.62	6.03	0.0927	154.61	338	0.74	6.03	0.0849
222.79	210	4.55	8.56	0.0909	222.79	227	4.19	8.56	0.0919	222.79	617	1.52	8.56	0.0932	222.79	400	2.09	8.56	0.0847
206.52	196	3.95	7.69	0.0813	206.52	214	3.63	7.69	0.0820	206.52	592	1.29	7.69	0.0825	206.52	386	1.77	7.69	0.0739
183.30	177	3.96	7.35	0.0932	183.30	195	3.61	7.35	0.0933	206.38	592	1.29	7.65	0.0827	183.30	367	1.73	7.35	0.0820
194.88	186	3.93	6.91	0.0892	206.38	214	3.63	7.65	0.0822	183.30	569	1.25	7.35	0.0886	183.28	367	1.78	7.43	0.0843
222.58	210	4.49	8.28	0.0899	222.58	227	4.14	8.28	0.0909	222.58	617	1.50	8.28	0.0921	194.88	377	1.74	6.91	0.0800
157.66	155	1.87	5.68	0.0904	157.66	173	1.70	5.68	0.0911	157.66	530	0.58	5.68	0.0919	222.58	400	2.06	8.28	0.0838
207.35	197	4.24	7.25	0.0909	207.35	215	3.89	7.25	0.0916	207.35	594	1.39	7.25	0.0921	157.66	341	0.70	5.68	0.0846
185.73	179	4.16	6.71	0.1012	185.73	197	3.80	6.71	0.1013	185.73	572	1.32	6.71	0.0963	207.35	387	1.91	7.25	0.0825
198.38	189	3.38	6.36	0.0795	198.38	207	3.16	6.36	0.0814	198.38	581	1.24	6.36	0.0903	185.73	369	1.83	6.71	0.0895
184.44	178	4.19	6.69	0.1028	184.44	196	3.82	6.69	0.1028	184.44	570	1.32	6.69	0.0976	198.38	378	1.65	6.36	0.0793
															184.44	368	1.83	6.69	0.0905

the optimal route  $X_{11}$  are better than the corresponding mean values. The only other route possessing this property is the route  $X_9$  of length 185.73 m, which is very similar to  $X_{11}$ . As can be seen in Table 3, the route  $X_9$  is the second best route in all the scenarios. The third best route is of length 183.30 m for Scenarios 1 and 2; it is the route  $P_4$  shown in Figure 5. Also, the third best routes for Scenarios 3 and 4 are of length 222.79 m and 154.61 m, respectively. The former is the safest available route  $P_2$  and the latter is the shortest/quickest route  $P_1$ , both are depicted in Figure 5.

In the previous test, we assumed that an agent is moving together with the flow of people. However, an SAR team or a single rescuer can overtake people’s flow and move with a higher speed than one given by formula (1). Let us now consider the situation when an agent is moving with speed increased by 25%, that is, the new speed is  $1.25 S(e)$ , where  $S(e)$  is calculated according to (1). The results of this test are very similar to the previous one. For all the scenarios, exactly the same routes are generated as in Table 3. The travel times for the optimal route (184.44 m) are, of course, different: 142 s, 156 s, 456 s, 294 s for Scenarios 1–4, respectively.

#### 4.2 Moving upstairs in a counter-flow

Let us consider the situation when an SAR team or a single rescuer is moving upstairs in Scenarios 1–4. In this case, the movement is in a counter-flow, so the formula (1) is not applicable. The rescuers’ speed is dependent on their personal characteristics, whether they carry heavy equipment, etc. Let us assume that, for horizontal movement, the speed is 1.5 m/s for density  $\Delta=0 \text{ p/m}^2$ , which is consistent with the previous example. Also, suppose that the speed is 0.9, 0.5, and 0.3 m/s for densities 1, 2, and 3  $\text{p/m}^2$ , respectively. This data set can be described by the following function:

$$S(e) = \hat{K} 0.6^\Delta \tag{2}$$

where  $S(e)$  is the speed along link  $e$ ,  $\Delta$  is the population density in the area corresponding to  $e$ , and the constant  $\hat{K}$  is defined as follows:

- $\hat{K} = 1.50$  for horizontal movement,
- $\hat{K} = 1.16$  for moving downstairs,
- $\hat{K} = 0.87$  for moving upstairs.

Here we assume that the ratios (1.50:1.16:0.87) of the coefficient  $\hat{K}$  for different types of movement are the same as the ratios of the similar constant  $K$  in formula (1). The formula (2) should be considered as a first approximation of a rescuer’s speed in a counter-flow. Additional research is needed to develop this formula further. In contrast to formula (1) with a linear relationship, formula (2) represents an exponential relationship



**Table 5**  
Comparison of PIR-Algorithm2 with TDOR-Algorithm

	<i>PIR-Algorithm2</i>	<i>TDOR-Algorithm</i>
Dimension of model	3D (represented by two networks)	2D (tested for one floor only)
Applicability	Any extreme event with epicenters	Fire/smoke only if sensor information is available
Output	A set of routes (the optimal route depends on user's needs)	A single route (not dependent on user's needs)
Route proximity to hazard	It is measured by proximity index	Not measured
Hazard propagation	Yes (based on distances and the number of obstructions)	No (sensor information is needed)
Link proximity to hazard	Yes	Yes
Travel time	Yes	Yes, but along a flow only
Route complexity	Yes (based on 5 attributes)	No
Speed along a flow	Yes	Yes
Speed in a counter-flow	Yes	No
Dynamic aspects	Partial	Partial
Reliability/robustness	It is implied by the AHP, which possesses these properties	It has to be proved yet
Sensitivity	Insensitive to different parameters and scenarios of evacuation	It has to be proved yet

optimal route is produced, but  $P_9$  has a very high aggregate score, hence it can be considered as near-optimal.

Finally, let us apply sensitivity analysis to the speed of rescuers, because it is rather variable, depending on visibility, their personal characteristics, whether heavy equipment is needed, etc. More precisely, we consider two situations when the speed given by formula (2) is increased by 25% and decreased by 25% for the above Scenarios 1–4. Both tests generate the same sets of routes and the same optimal routes. Thus, even though formula (2) represents a first approximation of the real speed in a counter-flow, the method demonstrates some robustness with respect to variability in the speed.

### 4.3 Comparison with TDOR-Algorithm

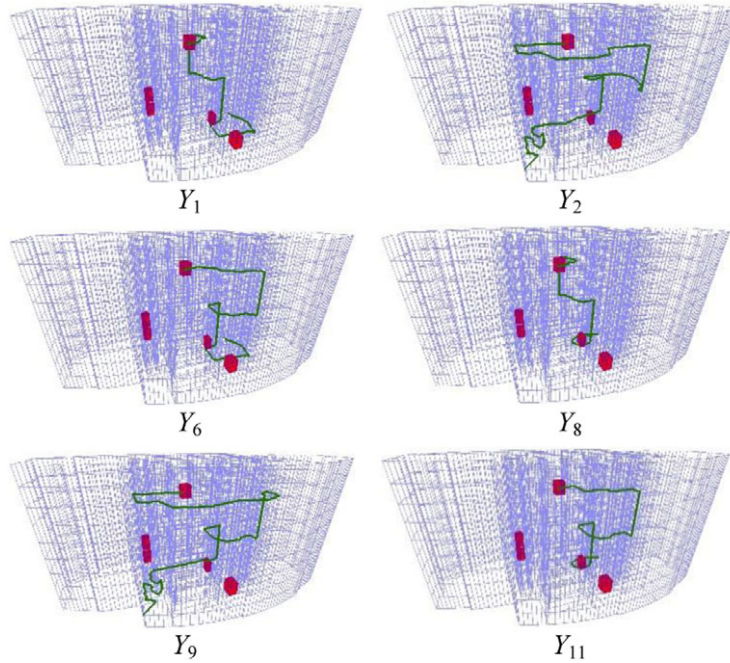
In this final section, PIR-Algorithm2 will be compared to the time-dependent optimal routing algorithm (TDOR-Algorithm) developed by Park et al. (2009). The main differences are summarized in Table 5. As explained in Section 2, PIR-Algorithm2 is based on the 3D model of a building, which is represented by the logical and unified networks. In addition, it is applicable to any extreme event with epicenters. In contrast, TDOR-Algorithm was only tested for a 2D model, and its applicability is limited to fire/smoke assuming that sensor information is available.

Another important feature of PIR-Algorithm2 is its ability to generate an optimal route depending on user's needs, thus producing a set of optimal routes. The hazard proximity of a route is measured by the

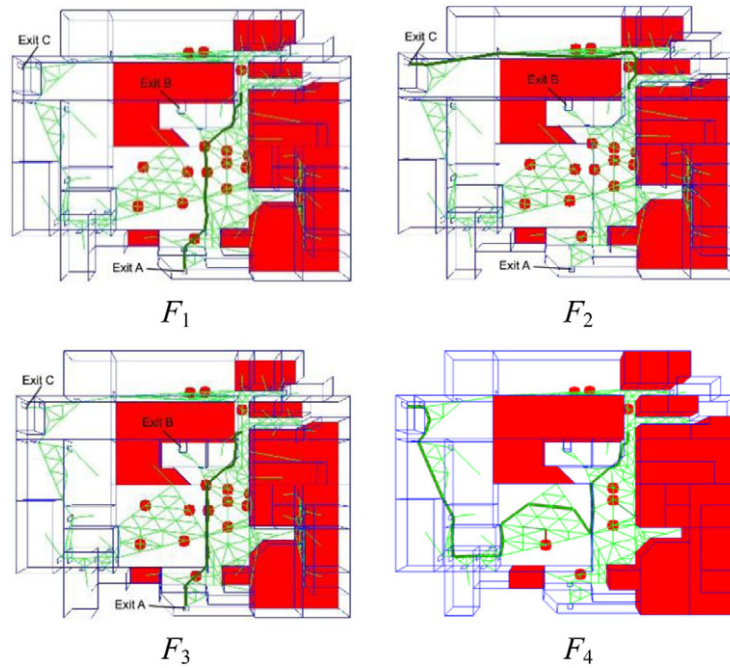
proximity index. TDOR-Algorithm generates a single route, and no measure of its hazard proximity is given. The algorithms determine link proximity to the hazard differently, the former is based on hazard propagation, whereas the latter relies on sensor information. Further advantages of PIR-Algorithm2 are that the route complexity is taken into account as well as the formula for agent's speed in a counter-flow. Both algorithms are applicable to various scenarios of evacuation, and hazard epicenters can be easily updated if sensor information is available. However, the dynamics of hazardous events should be developed further for situations without sensors. Finally, PIR-Algorithm2 is reliable and insensitive to different parameters and evacuation scenarios as shown in previous sections, whereas the same properties for TDOR-Algorithm have not been demonstrated yet.

In the following example, we illustrate the feature of PIR-Algorithm2 to generate an optimal route depending on user's needs. Let us consider Scenario 4 from the previous section when moving upstairs. We introduce five epicenters as shown in Figure 9, where the left red cylinder represents two epicenters on different floors. The destination node  $q$  is on the seventh floor in the middle part of the building. Because there are many epicenters, we put  $\rho_{\max} = 200$ . Depending on the user's preferential ranking, PIR-Algorithm2 produces the following routes shown in Figure 9: route  $Y_1$  for TT>HP>RC, route  $Y_8$  for TT>RC>HP, route  $Y_6$  for HP>TT>RC, route  $Y_9$  for HP>RC>TT, route  $Y_8$  for RC>TT>HP, route  $Y_{11}$  for RC>HP>TT, and route  $Y_8$





**Fig. 9.** Moving upstairs in Scenario 4 with five epicenters.



**Fig. 10.** Test building for TDOR-Algorithm.

for  $TT=HP=RC$ . Note that route  $Y_8$  appears several times because it is the simplest route with travel time close to the fastest route  $Y_1$ .

To apply TDOR-Algorithm to this scenario, we have to make some assumptions. Firstly, we assume that a correct formula for the movement in a counter-flow

is used in this algorithm. Secondly, the algorithm was only tested in a 2D model, so we assume that it is also applicable to a 3D model with a proper 3D hazard propagation. Under these assumptions, our calculations show that TDOR-Algorithm would produce the route  $Y_2$  (Time = 442 s; PI = 1.42; C = 10.27), which is similar



to the route  $Y_9$  (Time = 443 s; PI = 1.43; C = 9.28), generated by PIR-Algorithm2 with input HP>RC>TT. Note that travel times and proximity indices of these routes are very similar, however the complexity of the latter route is better. Thus, PIR-Algorithm2 produces a set of optimal routes depending on user's needs, whereas TDOR-Algorithm only generates a single route. In addition, PIR-Algorithm2 takes into account route complexity and calculates the proximity index, which is important in deciding how dangerous the route is.

Let us now consider the test building used for TDOR-Algorithm, which is shown in Figure 10. This is a 2D model of the Student Hall at the University of Seoul with three exits A, B, and C. In this scenario, the rescuers should find the optimal route from one of the exits to the place located in the corridor on the right side of exit B. However, the rescuers can only use exits A and C. The hazard is represented by smoke, detected by sensors, and its epicenters are marked by red circles. If all nodes in a room are epicenters, then the entire room is colored in red. Under this scenario, TDOR-Algorithm generates the single route  $F_2$  shown in Figure 10. If the travel time is the most important criterion, for example, there is protective equipment, then the optimal route produced by PIR-Algorithm2 is the route  $F_1$  depicted in Figure 10. This situation corresponds to the user's input TT>RC>HP or TT>HP>RC. However, if the hazard proximity is the most important attribute (HP>RC>TT or HP>TT>RC), then the optimal route is  $F_2$ , which agrees with TDOR-Algorithm. In the situation when the route complexity is most important, or the three criteria are equally important, the optimal route is  $F_3$ . It may be pointed out that with fewer epicenters and when the hazard proximity is most important, another rather unexpected optimal route  $F_4$  is generated; it is shown in Figure 10. This can be explained by the fact that all other routes go through epicenters or are extremely close to epicenters. The limitations of this scenario are that it is a simple 2D model, where there are relatively few possible routes from the exits to the destination. Nevertheless, it is still possible to see that even in this simple scenario PIR-Algorithm2 is able to produce the optimal route depending on user's needs.

## 5 CONCLUSIONS

In this article, we presented a step change for finding the optimal routes for SAR teams in a building, where an extreme event is occurring. We proposed an algorithm that is based on the new stochastic version of the AHP with three criteria: distance, hazard proximity, and route complexity. Another algorithm incorporates

the travel time instead of the distance criteria, thus taking into account more realistic information about the distribution of people in the building. For calculation of travel times, we used Nelson–MacLennan–Pauls' formula for people's indoor speed, and extended this formula for an SAR team moving in a counter-flow. Five attributes for route complexity were introduced and validated: going upstairs, going downstairs, doors, turning angles, and distance. Also, hazard proximity numbers and the proximity index were further developed for large open indoor spaces and for travel times.

The algorithms were thoroughly tested on the realistic complex building, which is based on the typical floor of the Doha World Trade Centre. The generated optimal routes are indeed the "best" trade-off among distance/travel time, hazard proximity, and route complexity, as discussed in Sections 3 and 4. The test results also demonstrated the robustness of the algorithms with respect to different parameters, and their insensitivity to different scenarios of uncontrolled evacuation.

The dynamic character of hazardous events is partially taken into consideration in the presented algorithms, mainly because beside the optimal route they produce the safest available route assuming that the hazard propagates evenly. In the future work, we will consider the dynamics of the situation more carefully, which is particularly important if the hazard develops rapidly and unevenly. Although the algorithm takes into account the agent's speed along people's flow and in a counter-flow, it should be further developed by including human factors and behavioral issues. In particular, the evacuation scenarios for uncontrolled evacuation can be simulated with behavioral aspects.

The information about construction materials will also be taken into account for a more accurate propagation of hazard and for finding alternative routes in case of direct hazard or non-availability of a safe route, that is, when walls or partitions may be drilled to obtain access to adjacent rooms/corridors. In this context, it would be interesting to develop an Ant Colony Optimization algorithm, which has been successfully applied for finding evacuation routes during a tsunami (Forcael et al., 2014).

## ACKNOWLEDGMENTS

This research/publication was made possible by a National Priority Research Program NPRP award [NPRP-06-1208-2-492] from the Qatar National Research Fund, a member of The Qatar Foundation. The statements made herein are solely the responsibility of the authors.

## REFERENCES

- Aboshosha, A. & Zell, A. (2003), Adaptation of rescue robot behaviour in unknown terrains based on stochastic and fuzzy logic approaches, in *Intelligent Robots and Systems, Proceedings of the 2003 IEEE/RSJ International Conference*, Vol. 3, 2859–64.
- Barbarosoğlu, G. & Arda, Y. (2004), A two-stage stochastic programming framework for transportation planning in disaster response, *Journal of the Operational Research Society*, **55**(1), 43–53.
- Boguslawski, P., Mahdjoubi, L., Zverovich, V., Barki, H. & Fadli, F. (2015), BIM-GIS modelling in support of emergency response applications, in L. Mahdjoubi, C. Brebbia, and R. Laing (eds.), *Building Information Modelling (BIM) in Design, Construction and Operations*, Vol. 149, WIT Press, Southampton, UK, 381–92.
- Boguslawski, P., Mahdjoubi, L., Zverovich, V. & Fadli, F. (2016), Automated construction of variable density navigable networks in a 3D indoor environment for emergency response, *Automation in Construction*, **72**(2), 115–28.
- Chang, M. S., Tseng, Y. L. & Chen, J. W. (2007), A scenario planning approach for the flood emergency logistics preparation problem under uncertainty, *Transportation Research Part E: Logistics and Transportation Review*, **43**(6), 737–54.
- Donegan, H. A. & Dodd, F. J. (1991), A note on Saaty's random indexes, *Mathematical and Computer Modelling*, **15**(10), 135–37.
- Duckham, M. & Kulik, L. (2003), “Simplest” paths: automated route selection for navigation, in W. Kuhn, M. F. Worboys, and S. Timpf (eds.), *Spatial Information Theory: Foundations of Geographic Information Science*, Springer, Berlin, pp. 169–85.
- Forcael, E., González, V., Orozco, F., Vargas, S., Pantoja, A. & Moscoso, P. (2014), Ant colony optimization model for tsunamis evacuation routes, *Computer-Aided Civil and Infrastructure Engineering*, **29**(10), 723–37.
- Foreman, E. H. (1993), Facts and fictions about the analytic hierarchy process, *Mathematical and Computer Modelling*, **17**(4-5), 19–26.
- Friedman, R. (1992), An international survey of computer models for fire and smoke, *Journal of Fire Protection Engineering*, **4**, 81–92.
- Fruin, J. J. (1987), Pedestrian planning and design, *Elevator World Inc.* (Educational Services Division), Alabama, LIB.CON 70-159312.
- Gomes, E. G. & Lins, M. P. E. (2002), Integrating geographical information systems and multi-criteria methods: a case study, *Annals of Operations Research*, **116**, 243–69.
- Gwynne, S., Galea, E. R., Lawrence, P. J., Owen, M. & Filippidis, L. (1999), A review of the methodologies used in the computer simulation of evacuation from the built environment, *Building and Environment*, **34**, 741–49.
- Kisko, T. & Francis, R. (1985), EVACNET+: a computer program to determine optimal building evacuation plans, *Fire Safety Journal*, **9**(2), 211–20.
- Kuligowski, E. D. (2004), Review of 28 egress models, in R. D. Peacock and E. D. Kuligowski (eds.), *Proceedings of the Workshop on Building Occupant Movement during Fire Emergencies*, National Institute of Standards and Technology, Gaithersburg, MD, June 10–11, 2004, 68–90.
- Kwan, M.-P. & Lee, J. (2005), Emergency response after 9/11: the potential of real-time 3D GIS for quick emergency response in micro-spatial environments, *Computers, Environment & Urban Systems*, **29**, 93–113.
- Liu, L. & Zlatanova, S. (2011), A ‘door-to-door’ path-finding approach for indoor navigation, in *Gi4DM 2011: GeoInformation for Disaster Management*, Antalya, Turkey: International Society for Photogrammetry and Remote Sensing (ISPRS).
- Nelson, H. E. & MacLennan, H. A. (1995), Emergency movement, in P. J. DiNenno (ed.), *The SFPE Handbook of Fire Protection Engineering*, National Fire Protection Association, Quincy, MA, 3.286–3.295.
- Olenick, S. M. & Carpenter, D. J. (2003), Updated international survey of computer models for fire and smoke, *Journal of Fire Protection Engineering*, **13**, 87–110.
- Pangilinan, J. M. A. & Janssens, G. (2007), Evolutionary algorithms for the multi-objective shortest path problem, *International Journal of Applied Science, Engineering and Technology*, **4**(1), 205–10.
- Park, I., Jang, G., Park, S. & Lee, J. (2009), Time-dependent optimal routing in micro-scale emergency situation, in *The Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, Taipei, Taiwan, 714–19.
- Pauls, J. (1995), Movement of people, in P. J. DiNenno (ed.), *The SFPE Handbook of Fire Protection Engineering*, National Fire Protection Association, Quincy, MA, 3.263–3.285.
- Saaty T. L. (1980), *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*, McGraw-Hill, New York.
- Saaty T. L. (1990), *Multicriteria Decision Making: The Analytic Hierarchy Process*, RWS Publications, Pittsburgh, PA.
- Saaty T. L. (1999), *Decision Making for Leaders: The Analytic Hierarchy Process for Decision in a Complex World*, RWS Publications, Pittsburgh, PA.
- Shahabi, M., Unnikrishnan, A. & Boyles, S. D. (2015), Robust optimization strategy for the shortest path problem under uncertain link travel cost distribution, *Computer-Aided Civil and Infrastructure Engineering*, **30**(6), 433–48.
- Skriver, A. J. (2000), A classification of bicriterion shortest path (BSP) algorithms, *Asia Pacific Journal of Operational Research*, **17**(2), 199–212.
- Sun, Q. & Wu, S. (2014), A configurable agent-based crowd model with generic behaviour effect representation mechanism, *Computer-Aided Civil and Infrastructure Engineering*, **29**(7), 531–45.
- Turskis, Z., Daniūnas, A., Zavadskas, E. K. & Medzvieckas, J. (2016), Multicriteria evaluation of building foundation alternatives, *Computer-Aided Civil and Infrastructure Engineering*, **31**, 717–29.
- Ulungu, E. L. & Teghem, J. (1991), Multi-objective shortest path problem: a survey, in *Proceedings of the International Workshop on Multicriteria Decision Making: Methods–Algorithms–Applications*, Liblice, Czechoslovakia, 176–88.
- Vanclouster, A., Maeyer, P., Fack, V. & Weghe, N. V. (2014), Calculating least risk paths in 3D indoor space, in U. Isikdag (ed.), *Innovations in 3D Geo-Information Sciences*, Springer International Publishing, Switzerland, pp. 13–31.
- Wang, Y., Zhang, C., Zhu, L. & Sun, Q. (2010), Application of AHP in the evaluation of railway emergency plans, *Networking, Sensing and Control (ICNSC), International Conference*, April 10–12, 2010, 564–69.

- Watts, J. M. (1987), Computer models for evacuation analysis, *Fire Safety Journal*, **12**, 237–45.
- Zahedi F. (1986), The analytic hierarchy process - a survey of the method and its applications, *Interfaces*, **16**, 96–108.
- Zhang, W., Sun, K., Lei, C., Zhang, Y., Li, H. & Spencer Jr., B. F. (2014), Fuzzy analytic hierarchy process synthetic evaluation models for the health monitoring of shield tunnels, *Computer-Aided Civil and Infrastructure Engineering*, **29**(9), 676–88.
- Zverovich, V., Mahdjoubi, L., Boguslawski, P., Fadli, F. & Barki, H. (2016), Emergency response in complex buildings: automated selection of safest and balanced routes, *Computer-Aided Civil and Infrastructure Engineering*, **31**(8), 617–32.