CORE
Metadata, citation and similar papers at core.ac.uk
Provided by UWE Bristol Research Repository

Emergency Response in Complex Buildings: Automated Selection of Safest and Balanced Routes

V. Zverovich*, L. Mahdjoubi, P. Boguslawski
University of the West of England, Bristol, United Kingdom
&
F. Fadli, H. Barki
Department of Architecture and Urban Planning, College of Engineering, Qatar University, PO BOX 2713, Doha, Qatar

Abstract: *The extreme importance of emergency response in complex buildings during natural and human-induced disasters has been widely acknowledged. In particular, there is a need for efficient algorithms for finding safest evacuation routes, which would take into account the 3D structure of buildings, their relevant semantics and the nature and shape of hazards. In this paper, we propose algorithms for safest routes and balanced routes in buildings, where an extreme event with many epicentres is occurring. In a balanced route, a trade-off between route length and hazard proximity is made. The algorithms are based on a novel approach that integrates a multi-attribute decision-making technique, Dijkstra's classical algorithm and the introduced hazard proximity numbers, hazard propagation coefficient and proximity index for a route.*

1 INTRODUCTION

Emergency response in the built environment is being widely studied, with a significant surge of interest in this area after 9/11. The focus of such studies is on rescue and evacuation, which are based on route finding and indoor navigation (Choi and Lee, 2009; Kwan and Lee, 2005; Lay, 2007; Lee, 2007; Liu and Zlatanova, 2011, 2012; Vanclooster et al., 2014). Despite many publications in this field in recent times, there is still a lack of appropriate evacuation algorithms and their implementations (Lee, 2007; Lee and Zlatanova, 2008; Meijers et al., 2005; Vanclooster et al., 2010).

Kwan and Lee (2005) investigated possible improvements of navigable networks for the particular purpose of facilitating quick emergency response to terrorist attacks in the integrated system of the ground transportation system and multi-story office buildings. They concluded that extending the standard 2D GIS (Two-Dimensional Geographic Information System) to a real-time 3D GIS has a "considerable potential for improving the speed of emergency response after terrorist attacks on multi-level structures in urban areas". Meijers et al. (2005) developed a semantic model representing interior spaces in a building. Their model can be used for an intelligent computation of evacuation routes.

Lee (2007) reviewed 3D models and building evacuation models, and developed a pedestrian-based indoor navigation model using 3D GIS. The human behaviour was studied by Choi and Lee (2009) using a social force model. Recently, an advanced configurable crowd model for different behaviours and scenarios was developed by Sun and Wu (2014), in particular the simulation of evacuation in a building was implemented. Liu and Zlatanova (2011) proposed a new door-to-door approach for finding routes between rooms and also a detailed route in a single room. Vanclooster et al. (2010) developed a capacity constrained flow algorithm on a 3D geometric network model. In 2014, Vanclooster et al. applied Grum's least risk path algorithm to an indoor space for minimising risks of getting lost, and proposed several improvements to Grum's algorithm in order to make it more compatible with indoor networks. The role of elevators and stairs in efficient evacuation was investigated by Lay (2007).

A spatial model always underlies any evacuation algorithms, including those mentioned above. 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 in order to model precise geometric properties (e.g. distance between nodes and their locations) and provide real navigation routes. This representation can be used for graph algorithms such as Dijkstra's or A* algorithms for finding shortest routes in evacuation planning (Dijkstra, 1959; Hart et al., 1968). Also, there is a considerable potential for using Ant Colony Optimization algorithms, which have been successfully applied for

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

finding evacuation routes during a tsunami (Forcael et al., 2014), combining both safety and distance to determine the best route. An advanced and efficient strategy for the shortest path problem with uncertain travel cost can be found in Shahabi et al. (2015). Their approach might be particularly relevant for evacuation algorithms in the built environment because the distribution of people in a building during an extreme event is typically unknown, so that the link travel time function is uncertain.

As a rule, 2D floor plans are used for reconstruction of horizontal navigable networks, and a 3D building is obtained by linking contiguous floors at some connection points, e.g. staircases. Spatial relationships between the rooms in the vertical direction are not reflected in the model. This solution is sufficient for a simple analysis of indoor human movements, but other important phenomena related to emergency response, e.g. fire spread or heat propagation, cannot be simulated in such models. Also, many 3D navigation models represent buildings with a simple, often regular, structure, which is not sufficient for complex interiors. Liu and Zlatanova (2012) proposed an interesting concept of automatic navigable network generation based on the geometry and semantics of a building. Their method requires a valid spatial model with preserved consistency between the geometry and topology, which is not always readily available.

In their pioneering work Kisko and Francis (1985) proposed the EVACNET+ software application for evacuation scenarios planning. By formulating the task at hand as an optimization problem, they developed a user-friendly interface allowing a user interaction, for the computation of egress times and the determination of problematic building locations. For the sake of efficient emergency evacuation, workload relaxation techniques from querying theory were used by Deng et al. (2008) in order to deal with the curse of complexity in large buildings. For this purpose, the authors developed tools for modelling and optimizing the occupancy evolution during evacuation scenarios. They derived complexity lower bounds on the evacuation time and consolidated their findings by providing realistic building simulations. Pursals and Garzon (2009) presented a new formulation for the problem of building evacuation, through the development of a model for occupants' movement, allowing a good choice of evacuation paths during emergency scenarios. In contrast to the approach for the proper selection of evacuation routes leading trapped occupants to main building exit points, Park et al. (2009) focused on computing optimal routes leading search and rescue personal to disaster locations. For this purpose, the authors developed a time-dependent optimal routing solution based on a network representing the building configuration, which has been enriched by relevant information about the facility.

Cellular automata have been widely used for evacuation simulations. For example, a crowd simulation model for large facilities was given by Abdelghany et al. (2010) for replicating the selection of exit gates, based on a trade-off between travel distance and the level of congestion. In the work of Kirchner and Schadschneider (2002) such models have been applied to analyse evacuation scenarios and reduce egress time for situations with a small number of doors. In the same spirit, Daoliang et al. (2006) developed a 2D cellular automata prototype. It was used in evacuation scenarios for the simulation of exit dynamics of the occupants of a building, i.e. for better choices of egress doors. They also provided some hints on the choice of the dimensions of the building exit points; this can be helpful for good building designs. Inspired by this work, Varas et al. (2007) also applied cellular automata but focused instead on studying the effect of fixed obstacles. The problem of the presence of obstacles in evacuation simulations has also been studied by Huang and Guo (2008), who proposed the model of a floor field and used a rectangular lattice site for the computation of navigation routes within it. One of their main conclusions was that familiarizing the occupants with rooms' inner configurations and exit door locations plays an important role in reducing the evacuation time. Based on available video records of student evacuation from a classroom, typical evacuation characteristics were investigated by Zhang et al. (2008) using an improved multi-grid model. The simulation process was modelled and its comparison with real evacuation experiments showed its closeness to reality, thus opening the way for understanding evacuation behavioural aspects. The discussed papers are summarised in the following table:

Table 1
Summary of the literature

3D modelling	Barki2015, Boguslawski2011&2015, Choi2009, Kwan2005, Lee2007&2008, Lienhardt1991, Liu2012, Meijers2005, Weiler1988
Route/evacuation optimization	Dijkstra1959, Hart1968, Kisko1985, Liu2011, Park2009, Pursals2009, Shahabi2015, Vanclooster2010&2014
Cellular automata and similar	Abdelghany2010, Daoliang2006, Kirchner2002, Varas 2007; Huang2008 (floor field model), Sun2014 (generic crowd model), Zhang2008 (multi-grid model)
Alternative evacuation models	Deng2008 (queueing model & fluid model), Forcael2014 (ant colony model), Lay2007 (role of elevators & stairs)

In this paper, we present an algorithm for finding the safest route in a building, where an extreme event with many epicentres is occurring. Another algorithm produces a balanced route, which is achieved by a trade-off between route length and hazard proximity. The algorithms are based on a novel approach that integrates the recently developed 3D building model described in the next section, a multi-attribute decision-making technique, Dijkstra's classical algorithm and the introduced hazard proximity numbers, hazard propagation coefficient and proximity index for a route. This study proposes the enhanced concept of hazard proximity with two criteria: distance and the number of obstructions (i.e. walls and floors) between the epicentre of an extreme event and points in the building. The algorithms are validated by testing them on different buildings and discussing the results in Section 5. Note that the underlying 3D model is constructed automatically and it includes all the semantics (3D or others) necessary for the aforementioned algorithms.

2 THE BIM-GIS MODEL

The research in this paper is based on the 3D building model recently developed by Boguslawski et al. (2015) and Barki et al. (2015). This model is an integration of Building Information Modelling (BIM) technology and Geography Information Science (GIS) analysis. Ideally, the Industry Foundation Classes (IFC) format could be used to exchange model between the design environment and GIS analysis tool. However, the information content in IFC is very high for GIS applications such as emergency management. The original model needs to be simplified in the process of generalization in order to reduce the storage cost and extract building elements and geometry essential for reconstruction of indoor spatial relations. This can be achieved through development of generalization procedures or using simplified models, which can be automatically generated in commercial software packages, e.g. Green Building XML (gbXML). We consider the gbXML format as a simplified BIM model, which is the main data format used as an input in our research: a detailed geometry is simplified to a level sufficient to preserve the adjacency relationship between rooms, which is essential. The information about room volumes and wall surfaces, including openings (i.e. doors and windows), is accompanied by attributes such as room names etc. The openings are used for conventional navigation and egress routes computation. However, other alternative routes can also be considered in case of direct hazard, e.g. walls or partitions can be drilled in order to get access to adjacent rooms. This requires additional information about construction materials, which can be obtained from the original model. The external links in gbXML allow to look up the required parameters if the original model is available, e.g. the IFC model. For example, suppose that there are two adjacent rooms (without a door between them) modelled by nodes *u* and *v*, and we know that the wall between them is thin, which is an attribute of the link *uv*. Then the link *uv* is made available for navigation in the network and the user is informed if this link happens to be in the egress route.

Our models are Boundary Representations (B-Rep) of the concerned buildings, where volumes (cells) are enclosed by faces, faces by edges and edges by points. A B-Rep may be modelled and implemented by various data structures, e.g. the radial-edge (Weiler, 1988), G-Maps (Lienhardt, 1991), or recently developed dual half-edges (DHE) (Boguslawski, 2011). The DHE structure has been adopted to encode the model geometry as the underlying navigable network can be simultaneously and automatically constructed together with the 3D model.

A simple model of a building stored in the Autodesk Revit format was exported to the gbXML model and reconstructed using the primal/dual DHE data structure (see Figure 1). Surfaces stored in the gbXML model represent boundaries of spaces, i.e. rooms. Information about adjacent spaces and incorporated openings, e.g. doors and windows, is attached to each surface. Original surfaces are represented with DHE as double-sided faces. Adjacency information is used to group surfaces into sets representing boundaries of separate spaces. They are merged together (see Figure 2) along adjacent edges into closed cells representing rooms using the Cardboard & Tape method (Boguslawski, 2011). The resulting model is a cell complex, where rooms are represented as cells in the primal structure with an associated dual node unambiguously representing this cell. Adjacent cells are connected by dual edges bounded by dual nodes (see Figure 3).

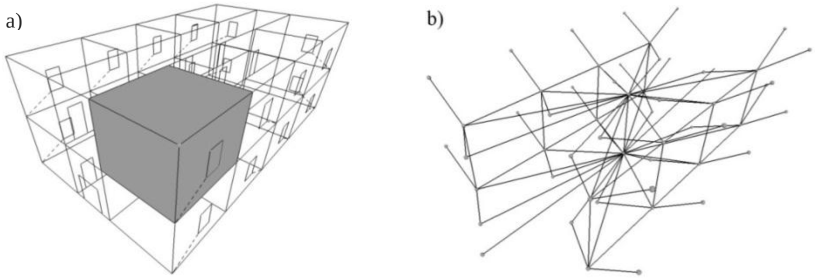


Figure 1 Simple model reconstructed using the DHE data structure: a) structure of a building includes one selected room (grey cell); windows are connected to wall boundaries by bridge edges (dotted lines); b) graph of connections between rooms

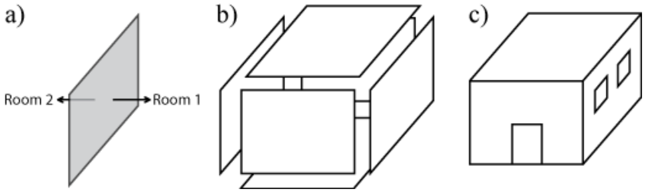


Figure 2 DHE model reconstruction from gbXML: a) gbXML surfaces include information about adjacent rooms; b) surfaces bounding a room are merged and form a cell; c) room with incorporated openings

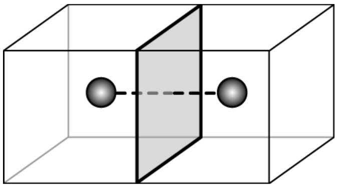


Figure 3 Two rooms represented by dual nodes are connected by a dual link penetrating a shared face

A door between two rooms is represented as a zero-volume cell with an associated unique node. Thus, there are two dual edges connecting the first room node to the door node, and the door node to the second room node. The same idea is applied in case of a door between an internal room and an external space. The latter is represented as a cell or a set of connected cells if it was partitioned. The complete graph of indoor connections is shown in Figure 1b. Some openings, which are not directly connected to the boundary of the enclosing surface, e.g. windows, are connected to the surface boundary by bridge edges (dotted lines in Figure 1a). A bridge edge is not a part of the original model, but it is introduced in order to preserve a valid topology of the B-Rep model.

The structure of the model presented in Figure 1 is simple and was reconstructed without additional improvement or validation. However, models with more complex structure exported to the gbXML format must be processed first in order to reconstruct a valid navigable network. Some common issues are unclosed and overlapping cells. Such models are valid for most of engineering analyses, but not for GIS, which requires a complete topology with a proper representation of spatial relations among objects. For validated models, the graph of connections reflecting spatial relationships among cells in the complex is created automatically using the DHE construction operators. For further details, see the paper by Boguslawski et al. (2015).

Summarising the BIM-GIS model, a building in general consists of several connected rooms that have volumes (corridors, offices, storage spaces etc. are considered as rooms too), so they are represented by primal cells. The geometry of a room is modelled by the links and nodes of a cell, and relations between adjacent rooms are represented with dual links connecting the corresponding cells. Those relations are described in terms of access level from one room to the adjacent one: access to the next room is by a door; the next room is not accessible because of a wall, but if the wall is thin a hole can be made. It may not be possible to get directly to the next room if the wall is made of concrete. This is an example of a basic set of attributes that are assigned to connections between rooms and then used as weights in graph traversal algorithms, e.g. Dijkstra’s algorithm. Rooms are not the only objects in a building that are important. Walls, doors, windows are essential and included in the model. They are represented as cells with geometry, and some attributes are assigned to them.

3 TEST BUILDING AND TESSELLATION OF CORRIDORS

Figure 4a shows a hypothetical building created in Revit. The building is then exported to gbXML and reconstructed with the DHE structure (Figure 4b). The latter is the primal model, which includes the geometry of the building. The graph of connections, i.e. the dual model, is then automatically created (Figure 4c). The building has three exits and three stairwells, and all floors have a similar structure, except for the exits at the ground floor. The plan of the ground floor is shown in Figure 5.

For navigation purposes, all corridors and large open spaces in a building should be partitioned automatically in order to generate a navigable network reflecting real navigation routes. Without a proper partition, an incorrect distance between nodes (cells) might be calculated and hence wrong connections (links) might be selected for evacuation. This is illustrated in Figure 5, where a person has to go from Room 11 to the nearest exit. According to the current model with the corridor modelled by just one node, the person would follow a dashed line, while the actual walking pattern should be different, one possible path is shown in Figure 6 (the bold line). To achieve an appropriate tessellation, we use an approach based on the Voronoi Diagrams (VD). VD partition a space into a set of adjacent cells represented as a graph. The dual graph to VD consists of links connecting adjacent dual nodes, which represent primal cells. These links form a network we use for navigation. It should be noted that for some concave shapes a Voronoi tessellation may produce cells, which are split into several unconnected parts, e.g. when a boundary of a corridor overlaps with the cell and some parts of the divided cell are not connected to the cell enclosing the dual node. However, in our implementation this situation does not exist. All dual nodes are enclosed by exactly one Voronoi cell.

For the corridors of the building in Figure 4a, this is illustrated in Figure 6, where one possible tessellation of the corridor is shown. Note that some cells in this tessellation have a node, e.g. the node *u* in Figure 6, which is either a door or a concave corner. Hence, such a cell is already associated with one primal node in the 3D model. However, for other cells, new tessellation nodes are introduced with the corresponding links, an example is the node *v* in Figure 6. Note that different tessellations are possible and the density of tessellation may be higher for larger areas. The choice of the tessellation density depends on the precision, which is required in the navigation route.

Let us denote by G the original 3D dual model of the building without tessellation, i.e. the graph of connections between cells, and by G^+ the 3D model of the building with all the necessary tessellations. Thus, G^+ has some additional nodes such as corner points and new nodes added during the tessellation process of corridors and large open spaces. Examples of the graphs G and G^+ are given in Figure 7, where only navigable links are shown. Note that non-navigable links, which are not shown in Figure 7, represent physical obstructions, e.g. the link between the nodes in the left and right top corners. Now, the shortest route from a selected room to the nearest exit can be easily calculated using Dijkstra's algorithm.

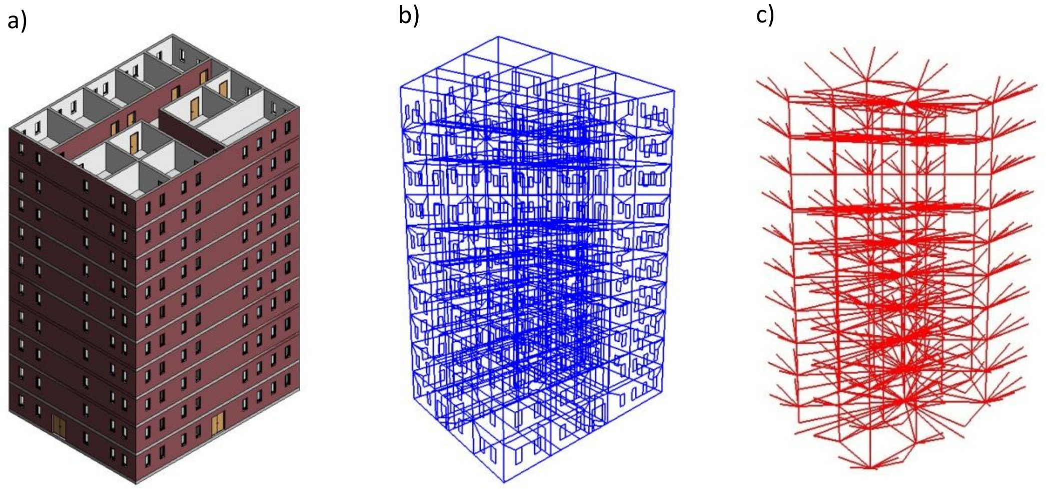


Figure 4 a) Test building; b) DHE reconstruction; c) graph of connections

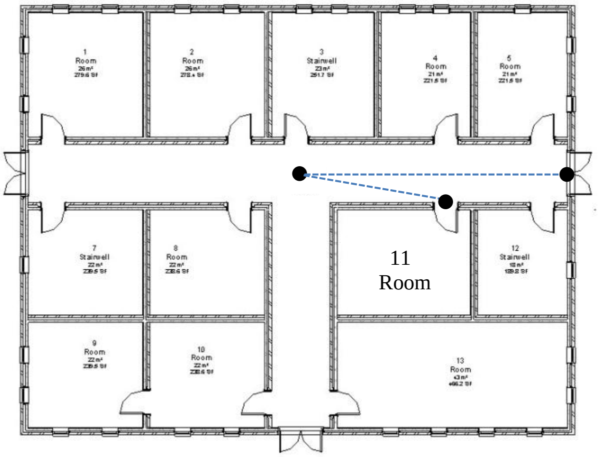


Figure 5 Plan of the ground floor of the test building

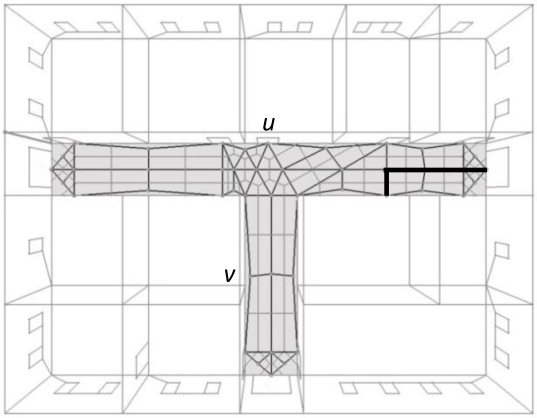


Figure 6 Tessellation of the corridor

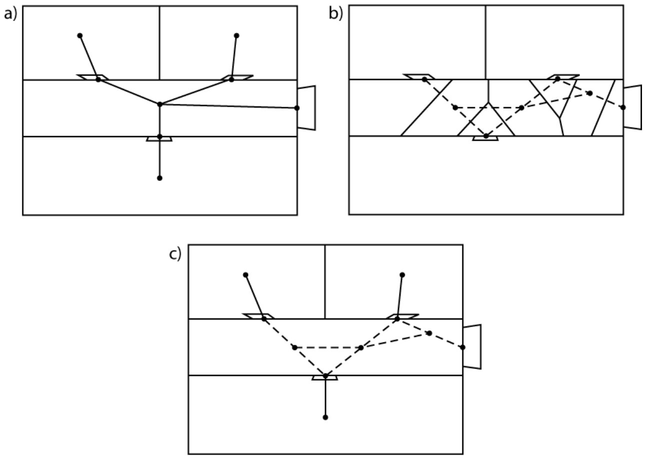


Figure 7 a) Graph G: logical network of the building (only navigable links via doors are shown); b) navigable network generated for a corridor: solid lines represent a Voronoi tessellation, dotted lines represent the navigable network; c) graph G^+: combined networks

4 ALGORITHMS FOR SAFEST AND BALANCED ROUTES

In this section, two algorithms for finding safest and balanced routes in buildings are presented. The algorithms are designed for complex buildings, e.g. high-rise buildings, shopping malls, where an extreme event is occurring. Typically, for people it is safer to stay further away from the epicentres of an extreme event in terms of the distance and the number of obstructions (e.g. walls). In this paper, we consider extreme events having this natural property. Examples of such events are fire, and terrorist activities such as bomb attacks or hostage taking situations. For the latter, the location of a bomb or a terrorist is an *epicentre*, whereas for fire the *epicentres* can be defined as points (nodes) in the building, where the temperature exceeds a certain threshold. Walls, floors, ceilings are called *obstructions*. For a particular point p in the building and an extreme event with the epicentre z, the hazard for that point is a function of the direct distance from p to the epicentre z and the minimum number of obstructions between p and z. Note that an extreme event may have many epicentres. For convenience, we summarize the notation, which will be used in this section:

$AS(P_\rho)$	Aggregate score for P_ρ
AS^*	Maximal aggregate score for all routes in R
$D(e)$	Length of link e in meters
$D(P_\rho)$	Length of P_ρ in meters
G	Graph of connections
G^+	Graph G extended by tessellation nodes
GM	Geometric mean
$H(v)$	Hazard proximity number for node v
$H_i(v)$	Hazard proximity number for node v w.r.t. z_i
$HD(e)$	Hazard proximity number for link e
P_ρ	(p,q)-route for the propagation coefficient ρ
$PI(P_\rho)$	Proximity index for P_ρ
R	Set of (p,q)-routes
$S_D(P_\rho)$	Distance score for P_ρ
$S_{PI}(P_\rho)$	Proximity index score for P_ρ
WGM	Weighted geometric mean
$b(v,z_i)$	Minimum number of obstructions between v and z_i
$d(v,z_i)$	Direct distance from v to z_i in meters
$d^+ (d^-)$	Maximal (minimal) lengths of routes in R
$l(P_\rho)$	Number of links in P_ρ
$p^+ (p^-)$	Maximal (minimal) prox. indices of routes in R
$r(e)$	Proximity ratio for link e
$r_i(e)$	Proximity ratio for link e w.r.t. z_i
t	Hazard tolerance, t = 0, 0.5 or 1
z_i	i-th epicentre
ρ	Propagation coefficient
$\rho_{\max}$	Maximal propagation coefficient
τ	Adapted propagation coefficient, $\tau=1+\rho/100$

4.1 Safest routes

The first objective is to find the safest route in a building, i.e. the total hazard proximity from a route to all the epicentres of the extreme event should be minimised. As explained above, the hazard has two criteria: distance and the number of obstructions. These criteria are very natural and should be considered together. For example, the distance of 40 meters between a person and an epicentre of one of the aforementioned extreme events in a direct visibility might be considered not as safe as the distance of 30 meters with two concrete walls between the person and the epicentre. It may be pointed out that we consider a generic extreme event. For a particular event, e.g. fire, our algorithms can be further developed by taking into account more precise models for heat propagation, smoke spread and structural collapse.

Thus, we consider an extreme event with k epicentres and two criteria for safety: distance and the number of obstructions. Based on these criteria, Algorithm 1 finds the safest available (p,q)-route in a building. Note that the application of this algorithm is threefold: it can be used by a rescue team to get from one of the entrances to a particular place in the building; as an evacuation algorithm from room p to one of the exits; or for navigation from point p to point q in the building.

Let $d(v,z_i)$ denote the direct distance from node v to the epicentre z_i in meters, and $b(v,z_i)$ the minimum number of obstructions between v and z_i. At the first stage of Algorithm 1, the direct distances and the number of obstructions are calculated in the DB_Procedure. More precisely, using the standard geometric technique, the direct distance $d(v,z_i)$ from node v to the epicentre z_i in meters is calculated for all nodes $v \epsilon V(G^+)$ and all epicentres. Then, the Breadth First Search Algorithm is run in the graph G from the node z_i. It finds the number of links in shortest paths from z_i to all other nodes in G. Since each link in those paths represents a physical obstruction, the number of links in such (z_i,v)-path is the minimum number of obstructions (i.e. walls and floors) between z_i and v. This number is denoted by $b(v,z_i)$, and it should be further adjusted for doors and exit doors. For example, if v represents a door with two adjacent rooms r_1 and r_2, then we put

$$b(v,z_i) = \min\{b(r_1,z_i), b(r_2,z_i)\},$$

because the links vr_1 and vr_2 do not represent a physical obstruction. Note that for some nodes, such as corner nodes and the 'new' nodes in the tessellation, the numbers $b(v,z_i)$ have not been calculated because those nodes are not in the dual graph G. Hence, for each such node $v\epsilon V(G^+)–V(G)$ we put $b(v,z_i) = b(w,z_i)$, where the node $w\epsilon V(G)$ represents the cell whose tessellation contains v.

It may be pointed out that the calculation of the parameters $b(v,z_i)$ is only possible because the dual graph incorporates all the necessary 3D information about the building.

```
Algorithm 1: Safest $(p,q)$-route in a building, where an
             extreme event is occurring

Input:  The graphs $G$ and $G^+$, which constitute the 3D
        model of the building. The epicentres of an
        extreme event (nodes $z_i$, $i$=1,2,…,$k$). Node $p$; node
        $q$ (optional; $q$ is one of the exits by default).
        The maximal propagation coefficient $\rho_{\max}$.
Output: Safest $(p,q)$-route $P$.

1: Run DB_Procedure to produce $d(v,z_i)$ and $b(v,z_i)$ for all
   $v\epsilon V(G^+)$ and $i$=1,2,…,$k$.
2: Run HP_Procedure with $\rho = \rho_{\max}$ to produce $HD(e)$ for
   all links in $G^+$.
3: Run Dijkstra's algorithm in the graph $G^+$ from node $p$
   with link weights $HD(e)$. It produces the safest available
   $(p,q)$-route $P$ in the building with two criteria: distance
   and the number of obstructions.
4: Report $P$. Algorithm stops.
```

```
DB_Procedure: Calculation of direct distances and the
              minimum number of obstructions

Input:  The graphs $G$ and $G^+$; nodes $z_i$, $i$=1,2,…,$k$.
Output: $d(v,z_i)$ and $b(v,z_i)$ for all $v\epsilon V(G^+)$ and $i$=1,2,…,$k$.

1: Repeat Steps 2 to 5 for each node $z_i$.
2: For each node $v\epsilon V(G^+)$, calculate $d(v,z_i)$, the direct
   distance from $v$ to $z_i$ in meters.
3: 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 $v$, i.e. the minimum
   number of obstructions $b(v,z_i)$ between $v$ and $z_i$ for all
   nodes $v\epsilon V(G)$.
4: For each node $v\epsilon V(G)$ representing a door, update $b(v,z_i)$
   as follows:
   a. If $v$ is an exit door, then put $b(v,z_i) = b(r,z_i)$, where
      the node $r$ represents a room adjacent to $v$.
   b. If $v$ is a door, but not an exit door, then
      $b(v,z_i) = \min\{b(r_1,z_i), b(r_2,z_i)\}$, where $r_1$ and $r_2$
      represent two rooms adjacent to $v$.
5: For each node $v\epsilon V(G^+)–V(G)$, put $b(v,z_i) = b(w,z_i)$, where
   $w\epsilon V(G)$ represents the cell whose tessellation contains $v$.
6: Report $d(v,z_i)$ and $b(v,z_i)$ for all $v\epsilon V(G^+)$ and $i$=1,2,…,$k$.
   Algorithm stops.
```

```
HP_Procedure: Calculation of hazard proximity numbers

Input:  The graphs $G$ and $G^+$; nodes $z_i$, $i$=1,2,…,$k$;
        the propagation coefficient $\rho\geq 0$;
        $d(v,z_i)$ and $b(v,z_i)$ for all $v\epsilon V(G^+)$ and $i$=1,2,…,$k$.
Output: Hazard proximity numbers $HD(e)$ for all links
        in $G^+$.

1: Calculate $\tau = 1 + \frac{\rho}{100}$.
2: Compute the hazard proximity numbers:
      $H_i(v) = \frac{100}{\tau^{\sqrt{d(v,z_i)\times b(v,z_i)}}}$
   for each node $v\epsilon V(G^+)$ and each $i$=1,2,…,$k$.
3: Calculate $H(v) = \max_{1\leq i\leq k} H_i(v)$ for each node $v\epsilon V(G^+)$.
4: For each link $e=uv\in E(G^+)$, compute the hazard
    proximity numbers for links: $HD(e)$= 0.5[$H(u)$ + $H(v)$]
    × $D(e)$, where $D(e)$ is the length of $e$ in meters.
5: Report $HD(e)$ for all links in $G^+$. Algorithm stops.
```

The second step of Algorithm 1 is to compute the hazard proximity numbers for all links in G^+ for the given propagation coefficient $\rho=\rho_{\max}$, where ρ represents the degree of hazard propagation, it is explained in more detail below. This is done in the HP_Procedure. Initially, for each epicentre z_i, the hazard proximity numbers $H_i(v)$ are calculated for all nodes in G^+. These numbers go from small positive values, which mean 'very far from the epicentre', to 100, which stands for 'inside the epicentre z_i'. The hazard proximity numbers are based on the parameters $d(v,z_i)$ and $b(v,z_i)$, which should be first replaced by one variable representing their average. For two variables with different numerical ranges it is appropriate to use a (weighted) geometric mean. Since $d(v,z_i)$ and $b(v,z_i)$ have different ranges, the weighted geometric mean is applied:

$$WGM = (d(v,z_i)^{w_1} b(v,z_i)^{w_2})^{\frac{1}{w_1+w_2}},$$

where w_1 and w_2 are the relative weights for the direct distance and the number of obstructions. Because the direct distance is as important as the number of obstructions, we can assume that the corresponding weights for the two variables are in proportion 50:50, i.e. w_1=0.5 and w_2=0.5. However, these weights can be adjusted if necessary, e.g. for buildings with many large open spaces and few obstructions. Thus, the above formula is simplified to the standard geometric mean:

$$GM = \sqrt{d(v,z_i) \times b(v,z_i)}.$$

Next, the values of geometric means should be transformed to the scale going from 100 to 0 taking into account the propagation coefficient $\rho \geq 0$. This is achieved by using the following formula:

$$\frac{100}{\left(1+\frac{\rho}{100}\right)^{\sqrt{d(v,z_i)\times b(v,z_i)}}}.$$

Now, if we denote $\tau = 1 + \frac{\rho}{100}$, then a well-justified formula for the hazard proximity numbers is obtained:

$$H_i(v) = \frac{100}{\tau^{\sqrt{d(v,z_i)\times b(v,z_i)}}}.$$

For instance, if ρ=100, then hazard proximity numbers for nodes propagate quickly from 100 (in the epicentre) to small positive numbers (far from the epicentre). This puts a strong emphasis on the epicentre and the rooms in its close proximity. In contrast, if ρ is a small positive number, then the propagation is slow, thus putting less emphasis on the epicentre and the nearby rooms. In the extreme case ρ=0 there is no propagation, i.e. all hazard proximity numbers for nodes are equal to 100.

Having calculated $H_i(v)$ for all the nodes in G^+ and all values of i=1,2,…,k, the following formula is used to compute the final hazard values for nodes:

$$H(v) = \max_{1\leq i\leq k} H_i\,(v).$$

Here we assume that the hazard at a particular node is equal to the maximal hazard proximity number at this node for all the epicentres, this approach is justified for many cases. A different formula can be easily incorporated in the algorithm if it is necessary, for example, to take into account the cumulative effect of all the hazard proximity numbers $H_i\,(v)$. Further, for each link e=uv in the graph G^+, the hazard proximity number $HD(e)$ for e is determined by calculating the arithmetic average of the hazard proximity numbers of its end-nodes, and then by multiplying the resulting number by the length of e in meters:

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

The last operation is important because G^+ is not a homogeneous network. For example, let us suppose that one link is 2 meters long and another is 10 meters long, and they both have the same hazard proximity number, say 10. If they both are used in a navigation route, then it is natural to assume that a travel time for a longer link would be approximately five times longer, so the hazard proximity number for the longer link should be 50. In other words, if we subdivided the longer link into five 2-meter-long links to make the network more homogeneous, then those five links would approximately contribute 50 to the total hazard proximity of the route.

The final stage of Algorithm 1 is to run Dijkstra's algorithm in the graph G^+ from node p with link weights $HD(e)$. It produces the safest available (p,q)-route P in the building with two criteria: distance and the number of obstructions. Note that the formula for the hazard proximity numbers is based on the weights w_1=0.5 and w_2=0.5 for $d(v,z_i)$ and $b(v,z_i)$. However, different weights can be easily incorporated in the formula.

4.2 Balanced routes

The second objective is to produce a balanced (p,q)-route in a building, where an extreme event is occurring. This is achieved in Algorithm 2, where one of the input parameters is the hazard tolerance coefficient t. The hazard tolerance is a trade-off between distance and safety, and it can be equal to 0, 0.5 or 1. For example, if t=1, then the shortest route will be generated. If t=0, then the algorithm finds the safest available route. If t is not specified and there are enough routes in the set R, then a route with the 50/50 balance of distance/safety will be reported as the balanced route.

Typically, there are many (p,q)-routes. The shortest route might go through the epicentre and be dangerous, whereas the safest route might be the longest one. A member of the rescue team, who is fully protected from the hazard, may wish to use the shortest route even if it is the most dangerous one, i.e. their hazard tolerance t is equal to 1. In contrast, an unprotected person with a respiratory disease may want to use the safest evacuation route, whatever is its length, in which case the hazard tolerance t is 0. The hazard tolerance is an optional parameter, and at the moment there are only two values. If it is not given, then by default t=0.5. The default value of 0.5 simply means that the hazard tolerance has not been specified, and this number will be used as a relative weight for the distance attribute, thus the relative weight for the hazard proximity will be 0.5 too. We use one variable t in this context because the single formula for the aggregate score $AS(P_\rho)$ will be applied for all values of t. Thus, if t is not specified, then a route with a right balance of distance/proximity will be chosen.

The first part of Algorithm 2 runs the DB_Procedure to determine all parameters $d(v,z_i)$ and $b(v,z_i)$. Then, the binary search is carried out with respect to ρ. The first run is for ρ=0, producing the shortest (p,q)-route P_0 because all hazard proximity numbers for links are 100$D(e)$. 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 ρ = 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 size of R, a specified length of the widest interval, and a running time. For our test buildings, the procedure goes on until seven routes are found or the length of the widest interval is less than 0.1.

Algorithm 2: Balanced (p,q)-route in a building, where an extreme event is occurring

```
Input:  The graphs $G$ and $G^+$, which constitute the 3D model of the building. The epicentres of an
        extreme event (nodes $z_i$, $i$=1,2,…,$k$). Node $p$; node $q$ (optional; $q$ is one of the exits by default).
        The maximal propagation coefficient $\rho_{\max}$. Hazard tolerance $t$, $t$=0 or 1 (optional; $t$=0.5 by default).
Output: Balanced ($p$,$q$)-route ($t$ is not specified); safest route ($t$=0); shortest route ($t$=1)

1: Run DB_Procedure to produce $d(v,z_i)$ and $b(v,z_i)$ for all $v \epsilon V(G^+)$ and $i$=1,2,…,$k$.
2: Put $R = \varnothing$, where $R$ is a set of ($p$,$q$)-routes.
3: Carry out the binary search with respect to $\rho$, $0 \le \rho \le \rho_{\max}$, starting with $\rho$=0, $\rho_{\max}$, 0.5$\rho_{\max}$ etc. For each value of $\rho$, implement the following:
   a. Run HP_Procedure for the specified value of $\rho$ to produce $HD(e)$ for all links in $G^+$.
   b. 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 $\rho$.
   c. Put $R = R \cup \{ P_\rho \}$ if $P_\rho \notin R$. (The shortest route $P_0$ belongs to $R$.)
   Go to Step 4 if at least one of stopping criteria is satisfied.
4: For each route $P_\rho \in R$:
   a. Calculate distance of $P_\rho$: $D(P_\rho) = \sum_{e \in P_\rho} D(e)$.
   b. Compute the proximity ratios
      $r_i(e) = \frac{\sqrt{d(u,z_i) \times b(u,z_i)} + \sqrt{d(v,z_i) \times b(v,z_i)}}{2D(e)}$ for each $i$=1,2,…,$k$ and each link $e=uv$ in $P_\rho$.
   c. Calculate $r(e) = \min_{1 \le i \le k} r_i(e)$ for each link $e=uv$ in $P_\rho$.
   d. Calculate the proximity index $PI(P_\rho) = \frac{l(P_\rho)}{\sum_{e \in P_\rho} \frac{1}{r(e)}}$.
      $PI$ is the harmonic mean of $r(e)$'s, and $l(P_\rho)$ is the number of links in $P_\rho$.
5: If $R$ consists of one route, then report $P_0$ and stop the algorithm.
6: Compute the following:
   $d^- = \min_{P_\rho \in R} D(P_\rho)$;  $d^+ = \max_{P_\rho \in R} D(P_\rho)$;
   $p^- = \min_{P_\rho \in R} PI(P_\rho)$;  $p^+ = \max_{P_\rho \in R} PI(P_\rho)$.
7: For each route $P_\rho \in R$, calculate the aggregate score:
   $AS(P_\rho) = 100\,t\left(1 - \left(\frac{D(P_\rho) - d^-}{d^+ - d^-}\right)^2\right) + 100(1-t)\left(1 - \left(\frac{p^+ - PI(P_\rho)}{p^+ - p^-}\right)^2\right)$.
8: Compute $AS^* = \max_{P_\rho \in R} AS(P_\rho)$.
9: Report $P_\rho$ for which $AS(P_\rho) = AS^*$. Algorithm stops.
```

In the next block of Algorithm 2, the total distance of each route in R is calculated. Then, the proximity ratios are computed for each link in a route. They are based on the geometric averages of parameters $d(v,z_i)$ and $b(v,z_i)$ for end-nodes of the link and its length. In contrast to hazard proximity numbers, the proximity ratios do not depend on the propagation coefficient ρ, and a small proximity ratio means a close proximity to one of the epicentres. For a given link, the final proximity ratio $r(e)$ is the smallest proximity ratio for that link: $r(e) = \min_{1 \le i \le k} r_i(e)$. The proximity index for a route P_ρ is the harmonic mean of $r(e)$'s for all links in the route:

$$PI(P_\rho) = \frac{l(P_\rho)}{\sum_{e \in P_\rho} \frac{1}{r(e)}},$$

where $l(P_\rho)$ is the number of links in P_ρ. Note that the proximity index is an average of *rates*. Also, the proximity index should not be dominated by sections of a route with large proximity ratios, and actually the impact of small proximity ratios is important. Therefore, the harmonic mean is an appropriate measure for the proximity index. Since the proximity index is independent on the propagation coefficient, it can be used for comparison of the routes from the set R.

In the final part of the algorithm, a multi-attribute decision-making technique is used to rank the routes in R and choose a balanced (p,q)-route. First of all, the maximal and minimal values of the lengths and proximity indices are calculated for all routes in R:

$$d^- = \min_{P_\rho \in R} D(P_\rho); \quad d^+ = \max_{P_\rho \in R} D(P_\rho);$$

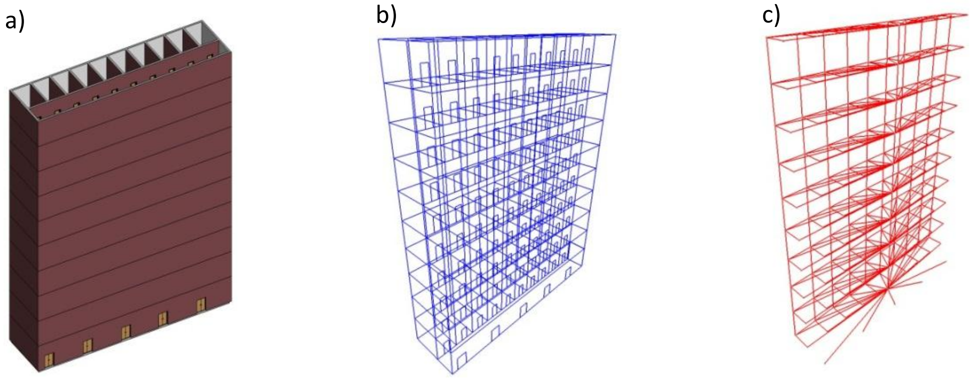


Figure 8 a) Virtual building; b) DHE reconstruction; c) graph of connections

$$p^- = \min_{P_\rho \in R} PI(P_\rho)\,; \quad p^+ = \max_{P_\rho \in R} PI(P_\rho).$$

Then, quadratic value functions are applied for rating the routes with respect to two attributes, the distance and the proximity index. Different value functions were tested, and it turned out that the most appropriate one is quadratic. For each route in *R*, the scores for these attributes are given by the following formulae, respectively:

$$S_D(P_\rho) = 100\left(1 - \left(\frac{D(P_\rho) - d^-}{d^+ - d^-}\right)^2\right)$$

and

$$S_{PI}(P_\rho) = 100\left(1 - \left(\frac{p^+ - PI(P_\rho)}{p^+ - p^-}\right)^2\right).$$

Finally, the aggregate score is calculated as a weighted average of the routes' rates, where the weights depend on the tolerance coefficient *t*. More precisely, the weights are *t* and 1–*t*. The balanced (*p*,*q*)-route is one with the highest aggregate score. For example, if *t*=1, then the shortest route is chosen. If *t*=0, then the algorithm returns the route with the highest proximity index, i.e. the safest available route. If *t* is not specified and there are enough routes in the set *R*, then a route with the 50/50 balance of distance/proximity will be reported as the balanced route.

5 TESTING THE ALGORITHMS

We start testing Algorithms 1 and 2 with a virtual building shown in Figure 8. This 10-floor building has 5 stairwells and 5 exits, and at each level there are 5 rooms connected by a long corridor as can be seen in Figure 9. The relatively large number of stairwells is needed to illustrate the behaviour of the algorithms.

In what follows, we put $\rho_{\max}$=100 for Algorithms 1 and 2. However, further testing is needed to decide which values of the maximal propagation coefficient are appropriate for different buildings. The epicentre of an extreme event (labelled by a star) is on the fourth floor, as can be seen in Figure 10. The starting point (the node *p*) is located on the top floor, above the epicentre, and the node *q* is not specified. Thus, we are looking for a route from *p* to one of the exits. It is not difficult to see that Algorithm 1 is a particular case of Algorithm 2 if we put *t*=0 in the latter, i.e. the former finds a route with the highest proximity index. Thus, it is enough to test Algorithm 2 for different values of the hazard tolerance *t*.

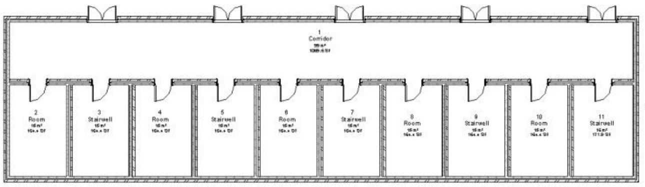


Figure 9 Plan of the ground floor

For the above scenario, the binary search of Algorithm 2 produces 5 different routes with the following propagation coefficients: ρ = 0, 7, 10, 11, 12. The lengths and the proximity indices for those routes are summarised in Table 2. As can be seen in the table, Algorithm 2 returns the shortest route P_0 if *t*=1, and the safest route P_{12} if *t*=0. If *t* is not specified, then the balanced route P_{10} is returned by the algorithm.

Table 2
Test results for Floor 9

Route	P_0	P_7	P_{10}	P_{11}	P_{12}
Distance (m.)	49.7	56.3	62.9	69.5	76.2
Proximity index	1.73	2.87	3.62	4.28	4.87
Status	Shortest	-	Balanced	-	Safest

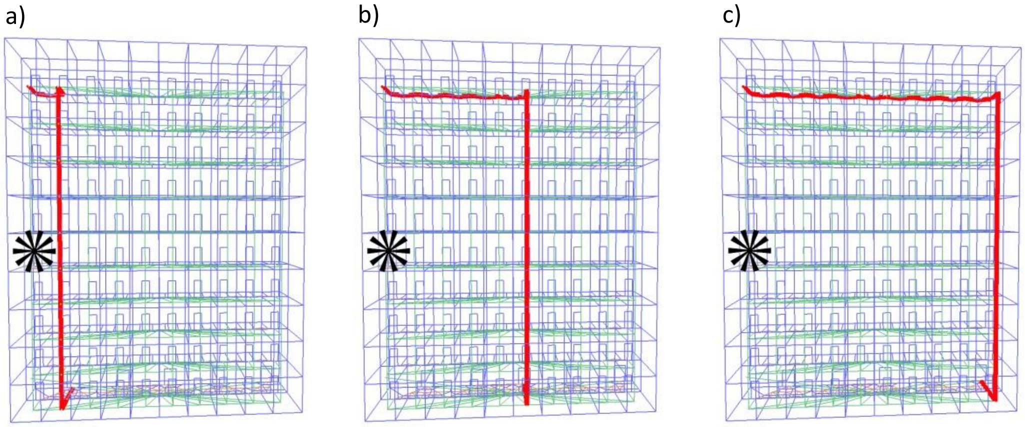


Figure 10 a) The shortest route P_0; b) the balanced route P_{10}; c) the safest route P_{12}

The route P_0 is shown in Figure 10a. It goes through the first stairwell, which is very close to the epicentre, so it is the most dangerous route, but the shortest one. The second route P_7 goes through the second stairwell, it is safer but longer. The routes P_{10}, P_{11} and P_{12} go through the third, fourth and fifth stairwells, respectively. The routes P_{10} and P_{12} are shown in Figures 10b and 10c. The scatter plot for the two parameters of the five routes is given in Figure 11. It is not surprising that there is a very strong positive correlation (at 1% significance level) between distance and proximity index. Also, the routes form a so-called 'efficient frontier' in the sense that no route is 'dominated' by another one, i.e. for any two routes one of them is safer but longer.

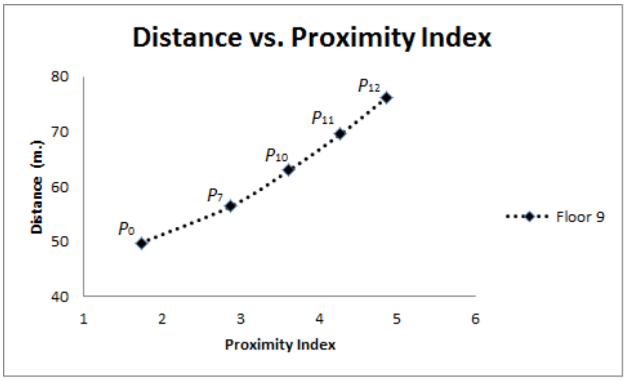


Figure 11 The scatter plot for routes P_0–P_{12}

Different floors for the starting room have been tested. In general, Algorithm 2 produces good results; however in some cases there is an unexpected behaviour. For example, if the starting point is located on the sixth floor, then the binary search finds five routes, see Table 3. The first four routes are similar to the results for the top floor; they go through the first four stairwells, respectively. However, the fifth route P_{26} does not go directly to the fifth stairwell. According to the algorithm, it is safer to first go upstairs and then use the fifth stairwell as can be seen in Figure 12. This is reflected in the corresponding proximity indices: 3.87 for P_{26} and 3.59 for the route that goes directly to the fifth stairwell. Note that P_{26} is much longer compared to other routes, so the balanced route P_{25} might be considered as a more reasonable one.

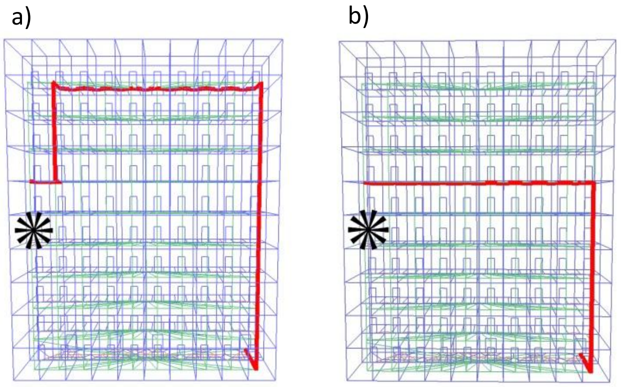


Figure 12 Starting point p on Floor 6: a) the safest route P_{26} (D=93.1 m., PI=3.87); b) the safest route P_{29} with an additional penalty for going upstairs (D=64.2 m., PI=3.59)

Table 3
Test results for Floor 6

Route	P_0	P_{12}	P_{22}	P_{25}	P_{26}
Distance (m.)	37.7	44.3	50.9	57.5	93.1
Proximity index	1.50	2.36	2.85	3.25	3.87
Status	Shortest	-	-	Balanced	Safest

The aforementioned behaviour is not necessarily unreasonable; it depends on the type of hazard. For some extreme events it might be deemed as safe, for others, e.g. fire, as unsafe. In the latter case, this problem can be rectified differently. The first approach is to include another criterion, route complexity, which will be investigated in a separate paper. The complexity of the route P_{26} would be rather high because it goes upstairs and uses two staircases, thus decreasing the likelihood that it will be eventually chosen. Another approach is to use a better model for hazard propagation in a building if the nature of hazard is known, however such models are out of scope of this paper where we consider a generic extreme event.

For the time being, we can use a simple approach based on a binary input variable x. It is equal to 1 if going upstairs is undesirable; and 0 otherwise. By default, x=1, in which case we add an additional ‘penalty’ for going upstairs in terms of distance. This penalty increases the hazard proximity numbers for links representing sections of a staircase that go up, thus making them undesirable in the routes. Note that such links are not forbidden completely, because in some cases going upstairs is unavoidable. This adjustment of Algorithm 2 produces the route P_{29} instead of P_{26}. The new route has a slightly worse proximity index (3.59 vs. 3.87), but it is much shorter (64.2 m. vs. 93.1 m.) and does not go upstairs. The adjusted algorithm returns P_{22} as a balanced route because the ‘outlier’ P_{26} was replaced by a much shorter route P_{29}.

Let us now consider the building of Figure 4, which has three stairwells. Instead of looking at simple situations with one or two epicentres, we simulate an extreme event with four epicentres as illustrated in Figure 13. This is an extremely tight situation, because the west stairwell is blocked by two epicentres at the third and seventh floors, whereas another two epicentres are located in the east and north stairwells at the fifth floor. Thus, in order to avoid the epicentres on the fifth floor, one has to use the west stairwell, which is not safe either. This means that a very safe route from the top floor to one of the exits does not exist, i.e. any route is very close to the hazard in such an extreme configuration of epicentres. Note that the hazard proximity numbers for rooms, which are shown in red/orange/yellow hues, correspond to the hazard proximity numbers for nodes in the dual model.

Table 4
Test results

Route	P_0	P_{11}	P_{165}
Distance (m.)	39.6	57.3	111.5
Proximity index	0.73	1.32	1.64
Status	Shortest	Balanced	Safest

The shortest, balanced and safest routes produced by Algorithm 2 are shown in Figure 14, where stars represent the epicentres, and the information about the routes is summarised in Table 4. Note that $\rho_{\max}$ is now 200 because there are many epicentres. As can be seen in Figure 14, the shortest route goes through two epicentres in the west stairwell, the balanced route uses the north stairwell with one epicentre, and the safest available route tries to stay further away from the epicentres by first using the east stairwell down to the sixth floor, then the west stairwell down to the fourth floor, and finally the north stairwell down to the ground floor.

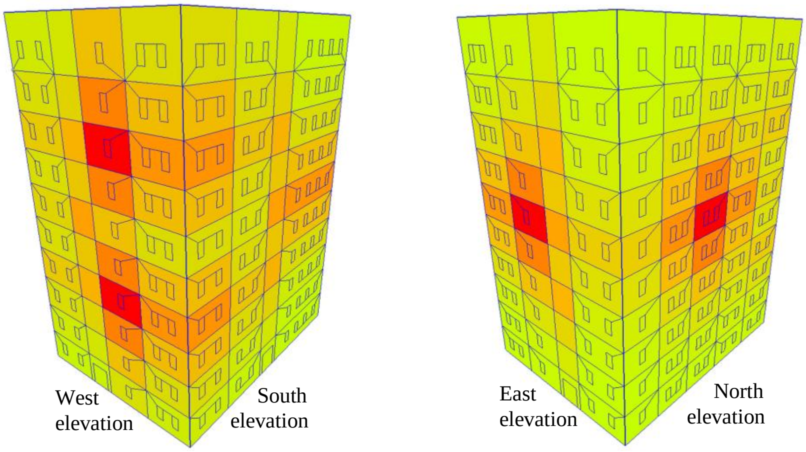


Figure 13 Simulation of an extreme event with four epicentres

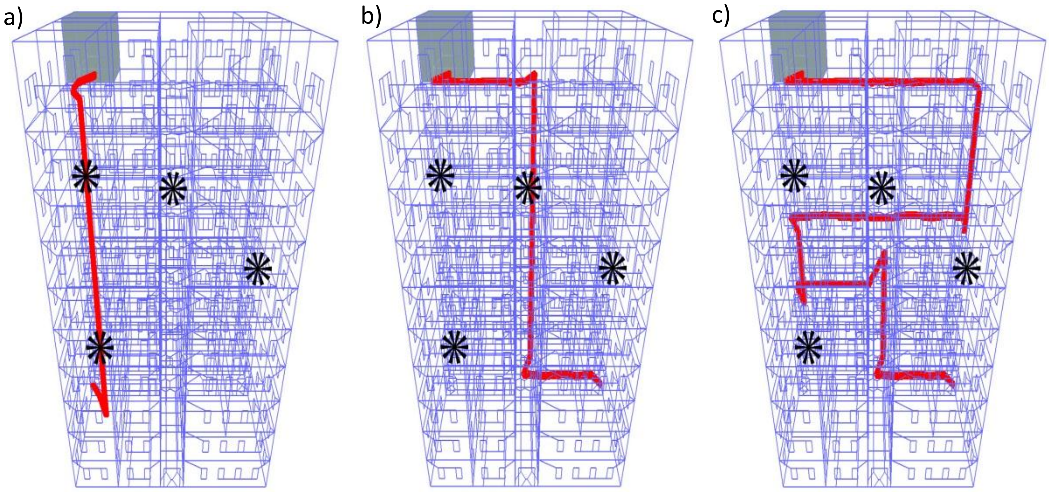


Figure 14 a) The shortest route P_0; b) the balanced route P_{11}; c) the safest route P_{165}

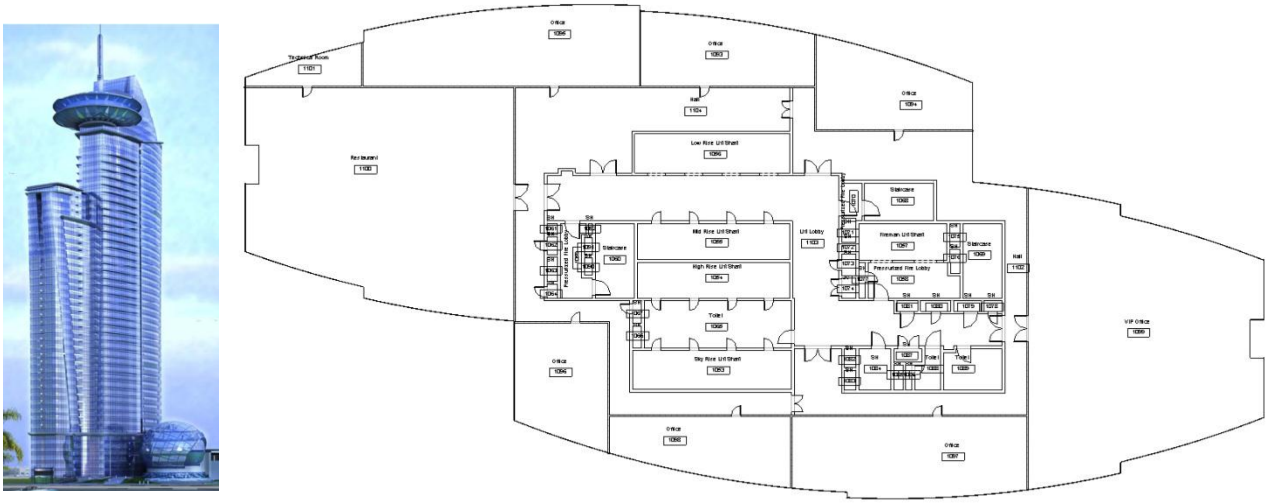
The above extreme example makes it obvious that not only the global hazard proximity of a route is important, but also a local proximity of route's nodes/links to the hazard should be taken into account. Indeed, the proximity index is a global parameter, which can be used to compare different routes. However, it does not tell us that the route is going through or very close to one of the epicentres. Some threshold values for the proximity ratios $r(e)$ could be used for this purpose if an epicentre is not located in a corridor or a large open space. Otherwise, the direct distance $d(v,z_i)$ from v to the epicentre could be a better measure of local hazard proximity of the node v. The assessment of route's local hazard proximity is out of scope of this paper. However, in the context of the above example, the proximity ratio less than 0.5 means an extreme proximity to an epicentre, whereas the proximity ratio between 0.5 and 1 represents a close proximity. Thus, if a user opts for a balanced route in our example, then (s)he should be given the information that it is extremely close to the epicentre together with the option to choose the safest available route.

Finally, let us consider a more realistic example, which is based on the Doha World Trade Centre (DWTC). A typical actual floor of this building is shown in Figure 15, and its DHE reconstruction and tessellation are illustrated in Figure 16. The left and right pictures of Figure 17 demonstrate the shortest and safest routes between two rooms, respectively. Further, the 37-floor building of Figure 18 was generated using the aforementioned floor plan. In order to add even more complexity, the locations and the number of staircases were modified: there are three staircases between the ground floor and the third floor; three staircases at different locations between Floors 5 and 36; and six staircases on Floor 4 (three going up and three going down). The resulting 37-floor building is not exactly the DWTC, but it is based on the actual floor plan of the DWTC. Figure 18 illustrates the shortest and safest routes in this building, where stars represent 3 hazard epicentres.

As mentioned above, the algorithms can be used for navigation from point p to point q in the building; this is illustrated in Figure 17. Also, they can be used for evacuation from room p to one of the exits or by a rescue team to get from one of the entrances to a particular place in the building. This is illustrated in Figure 18. From the practical viewpoint, the 3D model of the building can be created in advance and kept in the cloud. On arrival, the rescue team should detect the event epicentres. For example, in case of fire, the team can scan the building with appropriate equipment or use temperature sensors in the building if they are available. Then the model in the cloud is updated with epicentres and the safest available route is found. Notice that people inside the building will also have access to the updated model in the cloud, and there is a potential for using an indoor navigator.

6 CONCLUSIONS

In this paper, we presented an algorithm for finding the safest route in a building, where an extreme event with many epicentres is occurring. Another algorithm produces a balanced route, in which a trade-off between route's length and hazard proximity is made. Note that the hazard proximity has two criteria: distance and the number of obstructions between the epicentre and a point in the building. The proposed algorithms are based on the BIM-GIS model developed by Boguslawski et al. (2015), and they essentially use the underlying 3D structure of the model. For example, the calculation of the parameters $b(v,z_i)$ is only possible because the dual graph incorporates all the necessary 3D information about the building.



Courtesy of *MZ&Partners Architectural & Engineering Consultancy*

Figure 15 The Doha World Trade Centre (DWTC) and its typical floor

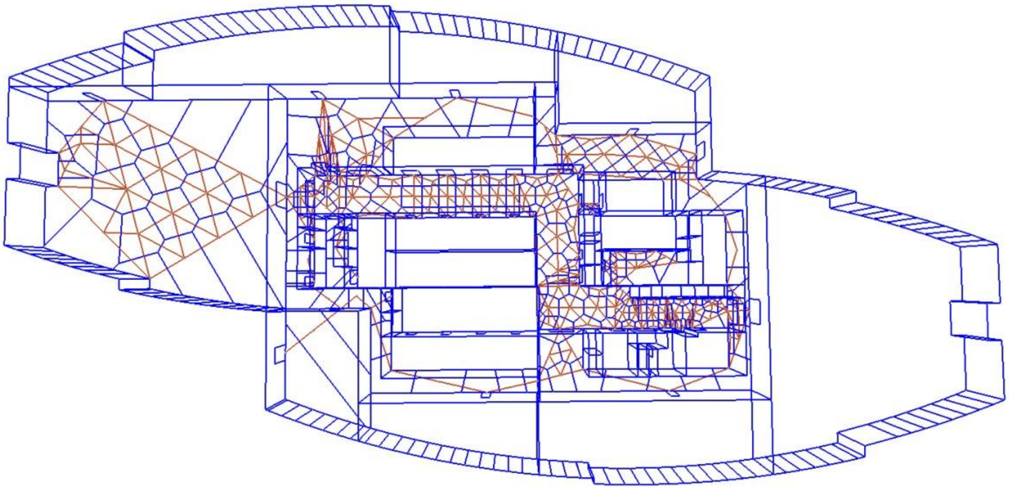


Figure 16 DHE reconstruction and tessellation of DWTC

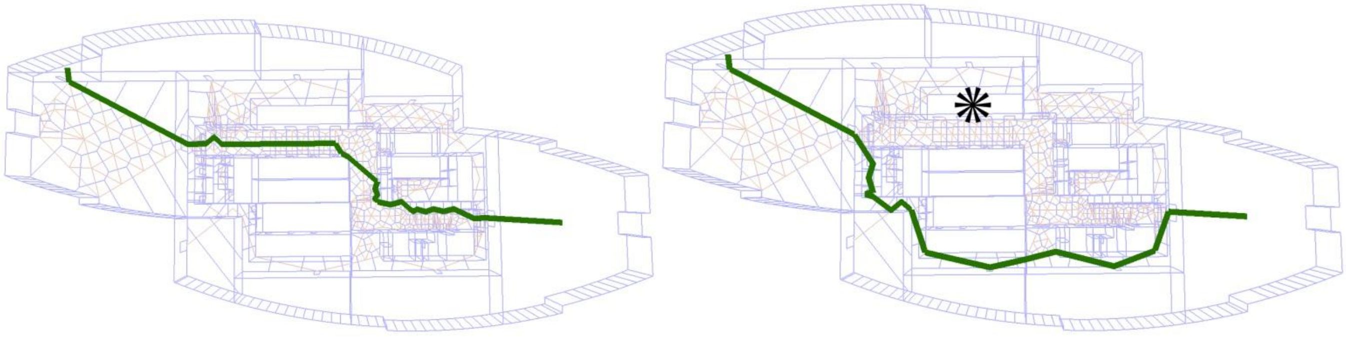


Figure 17 The shortest and safest routes between two rooms (a star represents a hazard epicentre)

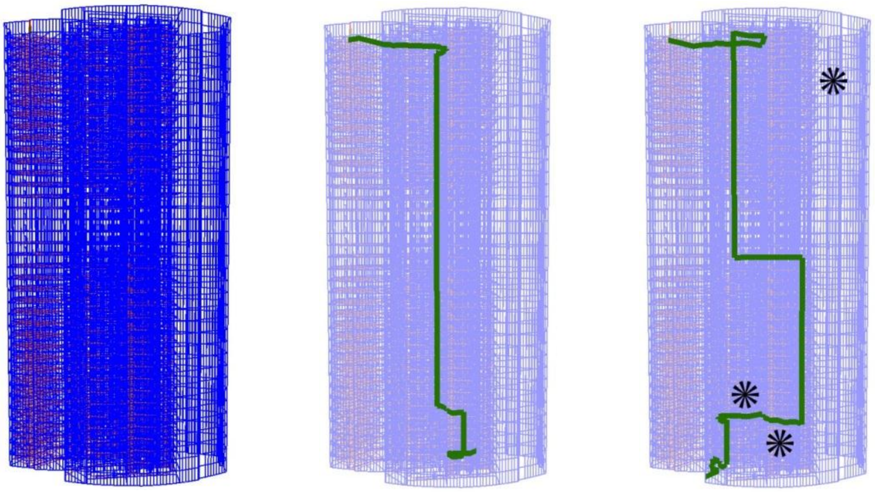


Figure 18 The 37-floor building reconstructed with DHE, shortest and safest routes

As illustrated above, the application of the algorithms is threefold: they can be used by a rescue team to get from one of the entrances to a particular place in the building; for evacuation from room *p* to one of the exits; or for navigation from point *p* to point *q* in the building. In addition to the 3D BIM-GIS model, the algorithms are based on multi-attribute decision-making technique and the introduced hazard proximity numbers, hazard propagation coefficient and proximity index for a route. The formulae for these parameters are well-justified and they have been validated by testing. Also, they can be easily adjusted to incorporate different weights for $d(v,z_i)$ and $b(v,z_i)$.

The results of testing are promising; in many instances the algorithms produce very reasonable balanced and safest routes. However, in some cases the safest available route is rather long and it goes upstairs. One possible extension to overcome this issue is to include another criterion, route complexity, which will be investigated in a separate paper. Another approach is to use a better model for hazard propagation in a building if the nature of hazard is known. For example, in the case of fire, more precise models for heat propagation, smoke spread and structural collapse can be used. A further limitation is that the algorithms do not take into account multiple agents that create route conflicts and congestion in a building. However, the capacity constraints can be easily included in the 3D model, so there is a potential for extending the algorithms in this direction.

In the future work, we will consider three criteria: distance, hazard proximity and route complexity. The multi-attribute rating technique will be applied for finding the ‘best’ route, i.e. a route which is reasonably short, safe and simple. It may be pointed out that the information about building material may be taken into account in the corresponding algorithm, as well as available sensor information (if any), for calculating the best egress route. Also, if distribution of people in a building is known, or can be predicted, then the distance criterion may be replaced by the time criterion thus taking into account multiple agents in the building. Another interesting extension would be to take into account the dynamics of the situation, which is particularly important if a hazardous event develops rapidly.

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.

The authors are grateful to 6 anonymous reviewers for very helpful comments and suggestions.

REFERENCES

Abdelghany, A., Abdelghany, K., Mahmassani, H., Al-Ahmadi, H. & Alhalabi, W. (2010), Modeling the Evacuation of Large-Scale Crowded Pedestrian Facilities, *Transportation Research Record: Journal of the Transportation Research Board*, **2198**, 152–160.

Barki, H., Fadli, F., Shaat, A., Boguslawski, P. & Mahdjoubi, L. (2015), BIM Models Generation from 2D CAD Drawings and 3D Scans: an Analysis of Challenges and Opportunities for AEC Practitioners, in Mahdjoubi, L., Brebbia, C. & Laing, R. (eds.), *Building Information Modelling (BIM) in Design, Construction and Operations*, (149) WIT Press, 369–380.

Boguslawski, P. (2011), Modelling and Analysing 3D Building Interiors with the Dual Half-Edge Data Structure, *Ph.D. thesis*, University of Glamorgan: Pontypridd, Wales, U.K.

Boguslawski, P., Mahdjoubi, L., Zverovich, V., Barki, H. & Fadli, F. (2015), BIM-GIS Modelling in Support of Emergency Response Applications, in Mahdjoubi, L., Brebbia, C. & Laing, R. (eds.), *Building Information Modelling (BIM) in Design, Construction and Operations*, (149) WIT Press, 381–392.

Choi, J. & Lee, J. (2009), 3D Geo-Network for Agent-based Building Evacuation Simulation, in Lee, J. & Zlatanova, S. (eds.), *3D Geo-Information Sciences*, Springer, 283–299.

Daoliang, Z., Lizhong, Y. & Jian, L. (2006), Exit Dynamics of Occupant Evacuation in an Emergency, *Physica A: Statistical Mechanics and its Applications*, **363**(2), 501–511.

Deng, K., Chen, W., Mehta, P. & Meyn, S. (2008), Resource Pooling for Optimal Evacuation of a Large Building, in *The 47th IEEE Conference on Decision and Control*, 5565–5570.

Dijkstra, E. W. (1959), A Note on Two Problems in Connexion with Graphs, *Numerische Math.*, **1**, 269–271.

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), 723–737.

Hart, P. E., Nilsson, N.J. & Raphael, B. (1968), A Formal Basis for the Heuristic Determination of Minimum Cost Paths, *IEEE Transactions on Systems Science and Cybernetics*, SSC4 **4** (2), 100–107.

Huang, H.-J. & Guo, R.-Y. (2008), Static Floor Field and Exit Choice for Pedestrian Evacuation in Rooms with Internal Obstacles and Multiple Exits, *Phys. Rev.*, E **78**, 021131.

Kirchner, A. & Schadschneider, A. (2002), Simulation of Evacuation Processes Using a Bionics-Inspired Cellular Automaton Model for Pedestrian Dynamics, *Physica A: Statistical Mechanics and its Appl.*, **312**(1-2), 260–276.

Kisko, T. & Francis, R. (1985), EVACNET+: A Computer Program to Determine Optimal Building Evacuation Plans, *Fire Safety Journal*, **9**(2), 211–220.

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.

Lay, S. (2007), Alternative Evacuation Design Solutions for High-Rise Buildings, *The Structural Design of Tall and Special Buildings*, **16**, 487–500.

Lee, J. (2007), A Three-Dimensional Navigable Data Model to Support Emergency Response in Microspatial Built-Environments, *Annals of the Association of American Geographers*, **97**(3), 512–529.

Lee, J. & Zlatanova, S. (2008), A 3D Data Model and Topological Analyses for Emergency Response in Urban Areas, in Zlatanova, S. & Li, J. (eds.): *Geo-information Technology for Emergency Response*, Taylor and Francis, London, 143–168.

Lienhardt, P. (1991), Topological Models for Boundary Representation: a Comparison with *n*-Dimensional Generalized Maps, *Computer Aided Design*, **23**(1), 59–82.

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).

Liu, L. & Zlatanova, S. (2012), Towards a 3D Network Model for Indoor Navigation, in Zlatanova, S. et al. (eds.), *Urban and Regional Data Management*, CRCpress/Taylor and Francis Group, London, 77–92.

Meijers, M., Zlatanova, S. & Pfeifer, N. (2005), 3D Geo-information Indoors: Structuring for Evacuation, in Groeger, G. & Kolbe, T.H. (eds.), *The First International Workshop on Next Generation 3D City Models*, 11–16.

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*, 714–719.

Pursals, S. & Garzón, F. (2009), Optimal Building Evacuation Time Considering Evacuation Routes, *European J. of Operational Research*, **192**(2), 692–699.

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), 433–448.

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), 531–545.

Vanclooster, A., Maeyer, P. & Fack, V. (2010), On the Way of Integrating Evacuation Approaches, in *The 5th International Conference on 3D Geoinformation*, **38**(4/W15), 169–172.

Vanclooster, A., Maeyer, P., Fack, V. & Weghe, N.V. (2014), Calculating Least Risk Paths in 3D Indoor Space, in Isikdag, U. (ed.), *Innovations in 3D Geo-Information Sciences*, Springer, 13–31.

Varas, A., Cornejo, M.D., Mainemer, D. Toledo, B., Rogan, J., Muñoz, V. & Valdivia, J. A. (2007), Cellular Automaton Model for Evacuation Process with Obstacles, *Physica A: Statistical Mechanics and its Applications*, **382**(2), 631–642.

Weiler, K. (1988), The Radial Edge Structure: A Topological Representation for non-Manifold Geometric Modeling, in Wozny, M.J., McLaughlin, H. & Encarnacao, J. (eds.), *Geometric Modeling for CAD Applications*, Elsevier Science, Holland, 3–36.

Zhang, J., Song, W. & Xu, X. (2008), Experiment and Multi-grid Modeling of Evacuation from a Classroom, *Physica A: Statistical Mechanics and its Applications*, **387**(23), 5901–5909.