A dynamic approach for evacuees' distribution and optimal routing in hazardous environments

Pawel Boguslawski[1,*], Lamine Mahdjoubi[2], Vadim Zverovich[2], Fodil Fadli[3]

[1] Faculty of Geoengineering, Mining and Geology, Wroclaw University of Science and Technology, Wroclaw, Poland

[2] Faculty of Environment and Technology, University of the West of England, Bristol, United Kingdom

[3] Department of Architecture and Urban Planning, College of Engineering, Qatar University, Doha, Qatar

* Corresponding author, e-mail: pawel.boguslawski@pwr.edu.pl, pboguslawski@gmail.com

Abstract

In a complex built environment, the situation changes rapidly during an emergency event. Typically, available systems rely heavily on a static scenario in the calculation of safest routes. In addition, egress route calculation and evacuation simulations are performed separately from path-finding for rescue teams. In this paper, we propose a state-of-the-art dynamic approach, which deals not only with a 3D environment, shape of spaces and hazard location, but also with the dynamic distribution of occupants during evacuation. A database of densities and information about hazard influence are generated and used to calculate optimal paths for rescue teams. Three simulation scenarios are compared in this study- namely, static with constant density values determined for subsequent stages of evacuation, semi-dynamic with densities representing an actual people distribution in a building during evacuation simulation, and dynamic with temporal distribution of evacuees stored in a database, and dynamically used in optimal path calculations. The findings revealed that static simulation is significantly different from semi-dynamic and dynamic simulations, and each type of simulation is better suited for the decision task at hand.

Keywords: indoor navigation, emergency response, evacuation simulation, 3D modelling

1. Introduction

In rapidly growing and complex urban environments, both natural and human induced, are inevitable. When an emergency situation happens in populated areas, a reliable hazard mitigation plan and rapid response should be in place in order to avoid a large number of casualties, injuries as well as economic loss. Major disaster events may result in fast environmental modifications, such as changes to the layout of a building, where plans prepared in advance are no longer applicable in a current situation. Consequently, rapid situation analysis should be performed and a new response plan swiftly developed.

Hazard analysis and routing for evacuees and first responders are essential elements of emergency support systems [13]. The main issue is to minimise time necessary to get people from a dangerous area to a safe zone [4,5]. Different areas affected by disaster are taken into consideration, for example regions, urban areas, transportation networks, or buildings [5], or combination of them [9]. Evacuation planning in complex buildings, investigated in this paper, are influenced by several

factors, including elements such as building layout, location and propagation of hazard, location of occupants and emergency personnel, location of safe places including exits, and human behaviour [5]. These factors are critical for calculating the shortest or the quickest routes, while information about hazard is used to permanently or temporally exclude some areas strongly affected by the propagation of a hazard (e.g. fire or smoke) from areas available for navigation. Information about hazard propagation and its influence on evacuation areas, is used to identify routes, which scale from areas of a higher hazard value to areas with a lower hazard value – opposite direction is avoided. Although this is deemed to be a reasonable approach, it may be, however justified to move to a more dangerous area for a short period of time, instead of remaining longer in an area with a moderate hazard level [18].

Within evacuation modelling, two major approaches may be distinguished: micro and macroscopic [5]. In the first approach, objects such as people, vehicles or interaction between them are introduced and their behaviour is modelled, while the latter approach is focused on optimisation models and no individual objects are included. Microscopic simulation is performed using agents representing individual objects. It helps to determine densities of objects in certain areas and, thus, used to detect congestions and bottlenecks [4]. In order to perform such analysis it is necessary to generate a navigable network and apply a path-finding algorithm [2,6,11,14,16,17].

In this paper, the proposed methods are scalable to various areas such as outdoors, because strong mathematical foundations for network representation, hazard analysis and path-finding algorithms were applied [2,18]. A microscopic evacuation planning is used for evacuation simulation, however, some aspects typical for macroscopic planning are adapted, such as taking into consideration maximum flow and density of evacuees, in people's movement simulation.

1.1. Problem statement

Egress routes from buildings are commonly calculated taking into consideration only limited number of factors that influence the shape and characteristics of resulting routes. Very often, the shortest or the quickest paths are calculated. Areas affected by hazards (e.g. fire or smoke) are included in calculation, but they are often excluded from navigation. In the case of evacuation simulations, where the aim is to estimate time necessary to empty a building, various aspects are taken into consideration, such as speed of evacuees' movement, depending on their age and health conditions, human and crowd behaviour.

The evacuation process affects the speed of emergency response. Congestions may be formed at bottlenecks, where too many people seek to use the same exit point, with a maximum flow capacity much lower than the actual demand. Typical values of occupancy density at different stages of evacuation may be obtained from simulations or from observations during fire drills. When access routes for search and rescue teams are calculated, densities are included in the estimation of the best possible-safe and quick- egress route. However, typical values may not reflect the real situation in various scenarios. In office buildings, various occupancy may be expected during working hours, , large events with many visitors and evenings and weekends. Therefore, the initial location of people in a building has an effect on densities during evacuation and thus, on the optimal routes for rescue teams.

In this research, agent-based evacuation simulation is performed in order to estimate occupancy density at each node of a navigable network, reconstructed from a 3D building model. The capacity of nodes and the maximum flow capacity of links are calculated and used in path-finding calculations. During simulation, transient densities are determined and stored in the network. The network with the density values is called the database of densities. The 'database' term used in this paper is understood as a method of data storage in a topologically connected structure with no formal query language, such as SQL. It is employed together with information about hazard locations and their influence on a building to dynamically calculate optimal paths that are safe and fast for search and rescue teams. The aim is to develop an effective and novel approach, where only a small set of input parameters are considered in order to achieve results comparable with other, more complex, solutions that are computationally more expensive. This makes real-time analysis and rapid decision making possible. Additionally, the proposed method can be used for building safety evaluation, where detection of bottlenecks in pathways may be detected in the early design stages.

2. Methodology

The initial stage in this research is the generation of a navigable network from a BIM model (see **Figure 1**). The method proposed by Boguslawski, et al. [2] is adopted for irregular tessellation of space and the variable density network generation. Each network node is associated with a cell of a specific area calculated in the tessellation process. A logical structure of the building, including non-navigable connections among adjacent spaces used for hazard propagation, is also available in the spatial model (for more details see [1]).The nine-floor mock model was created based on a typical floor plan of the Doha WTC building in Qatar. Location of indoor spaces is the same on all floors with following exceptions: there are exit doors on floor 1 and location of staircases on top floors is different than on lower floors; there are three staircases between floors 1 and 4, and three placed on different locations between floors 6 and 9, which meet on floor 5. The rationale for the selection of this specific staircase distribution is to introduce more options (i.e. possible exit pathways) for the path-finding algorithm, thus allowing better illustration of the method. There are five exit doors from the building on floor 1: three main doors in the main lobby and two side doors.
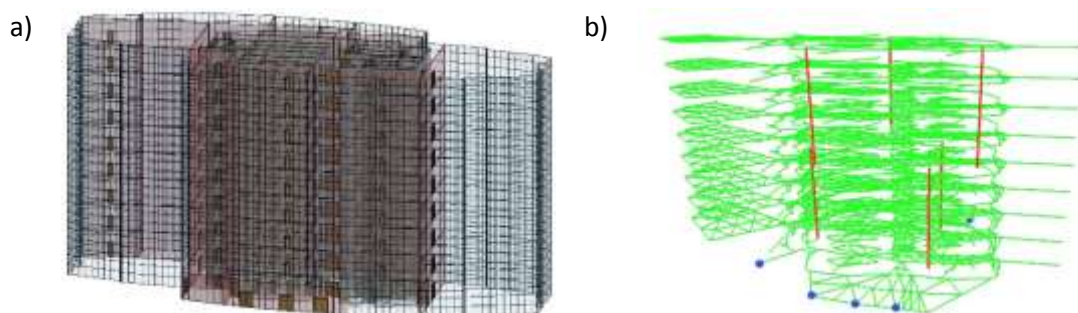

Figure 1. Generation of a navigable network from a BIM model: a) initial BIM model; b) navigable network. Red vertical lines represent staircase links, while blue nodes represent exit doors.

The navigable network is a core structure for path finding. Any graph-based algorithm may be applied to such a network. In this research, Dijkstra's algorithm is used for determining routes for evacuees. Temporal information about evacuees' density is included in calculations of optimal routes for rescue teams in hazardous indoor environments.

2.1. Agent-based evacuation simulation

In order to estimate the density of people in the navigable network during building evacuation, an agent-based evacuation simulation is performed. An initial set of agents is generated, involving the location of agents in all indoor spaces, except for main communication passageways, for example corridors, lobbies and staircases. A specific network node representing a cell is attached to each agent that defines its location – no precise location of an agent is calculated. In addition, no agent's size is considered in this research. The number of agents at a single node is then taken into consideration for calculation of the node occupancy and, consequently, agent's speed. The speed depends on the travel time calculated for all network links and local agent's density calculated for nodes.

In the next step, the quickest routes to the closest exit are generated for all agents. It should be noted that uncontrolled evacuation is considered in this research. This reflects the situation, when evacuees are acquainted with building layout and available closest exit routes, but they have no information about hazard or congestions on the pathway. If they reach a congested area, instead of waiting in one place, they may decide to use a different route to exit the building.

In order to apply the path-finding algorithm, it is necessary to calculate the travel time through network links, which is a function of the length and speed of movement:

$$T(L) = \frac{D(L)}{S(L)}$$
(1)

where T is the travel time [s] through link L, D is the link length [m], and S is the speed of movement [m/s].

The speed depends on density and it is calculated using Nelson-MacLennan-Pauls' formula [7,8]:

$$S(L) = \begin{cases} 0.856K & \text{if } 0 \leq \varDelta < 0.54 \\ (1 - 0.266\varDelta)K & \text{if } 0.54 \leq \varDelta \leq 3.75 \end{cases}$$
(2)

where $\varDelta$ is the average of the population density [people/m^2] in two cells corresponding to the bounding nodes of link L, and K is defined as follows:

K = 1.40 for horizontal movement,
K = 1.08 for moving downstairs,
K = 0.81 for moving upstairs.

The link travel time is considered by Dijkstra's algorithm as a weight, which is the cost of travelling through the link. Therefore, the route calculated for an agent is the quickest one. The quickest routes may be replaced by alternative egress paths if necessary, where different criteria, such as safety or complexity, are of higher importance. For example, the safest routes can be calculated using the method proposed by Zverovich, et al. [19]. However, in this research, the central issue is evacuees' density estimation and, thus, different criteria for agent's egress routes are not investigated.

Once routes for agents are calculated, simulation of their movement can be performed. It is an iterative process, where, in each step, the actual location of agents is determined based on their speed, which in turn is influenced by the actual density at network nodes and the flow through links.

After each iteration, the actual densities at network nodes are stored and used later. In order to apply this procedure, it is necessary to calculate the maximum capacity of nodes and maximum flow for links. Because nodes represent spaces of irregular shape and different area, the maximum capacities of nodes are calculated based on the corresponding space areas. Equation 3 determines the maximum number of agents that can be accommodated at any given time. It should be noted that density of 3.75 people/m² (see max. value of Δ in Equation 2) causes that people stop moving because of lack of space. In order to avoid such a situation, a constant 3.5 is used.

$$CAP_N = 3.5 \, A_N \qquad (3)$$

where: CAP_N is the maximum capacity of node N, A_N is the area of the cell associated with node N.

Then, the maximum flow CAP_L is determined for links connecting nodes, associated with adjacent cells. Algorithm 1 is used to determine an average number of agents that can move from one cell to another within 1 second.

Algorithm 1: Maximum link flow calculation

Input: Link L
Output: Maximum link flow CAP_L
1. If L is a door link: $\mathrm{CAP_L} = 1.3 \, (W_{Door} - 0.3)$, where W_{Door} is the door width
2. If L is a staircase link: $\mathrm{W_{Stair}} = 0.385 \sqrt{A_{Staircase}}$, where W_{Stair} is the width of stairs and $A_{Staircase}$ is the floor area of the staircase
   a. For the upstairs direction: $\mathrm{CAP_L} = 0.83(W_{Stair} - 0.3)$
   b. For the downstairs direction: $\mathrm{CAP_L} = 1.04(W_{Stair} - 0.3)$
3. If L is a link between two cells:
   a. If an edge between cells is bounded by a wall surface at two ends: $\mathrm{CAP_L} = 1.3(W_{Cell} - 0.4)$
   b. If an edge between cells is bounded by a wall surface at one end: $\mathrm{CAP_L} = 1.3(W_{Cell} - 0.2)$
   c. Otherwise: $\mathrm{CAP_L} = 1.3(W_{Cell})$, where W_{Cell} is the width of an edge between cells connected by L
4. Report CAP_L. Algorithm stops.

The maximum flow of a link is based on the effective width, which is the actual width reduced by a some distance from walls, door frames and railings: 15 cm for each door and staircase bounding walls, and 20 cm for cells adjacent to a wall [7]. It worth noting that due to the lack information in the original model about the width of stairs, it is approximated based on the staircase area (step 2 of Algorithm 1). It is assumed that staircases have rectangular shape with ratio of 3/5 between the short and long edge. In case of a staircase, the flow is different for upstairs and downstairs directions. Constant values used to calculate CAP_L are derived from parameter K used in Equation 2 and adjusted for maximizing the flow (e.g. the density is 1.88 people/m²).

The illustration of the actual width is demonstrated in Figure 2a-c while the effective width is explained in Figure 2d using the connection between two cells.
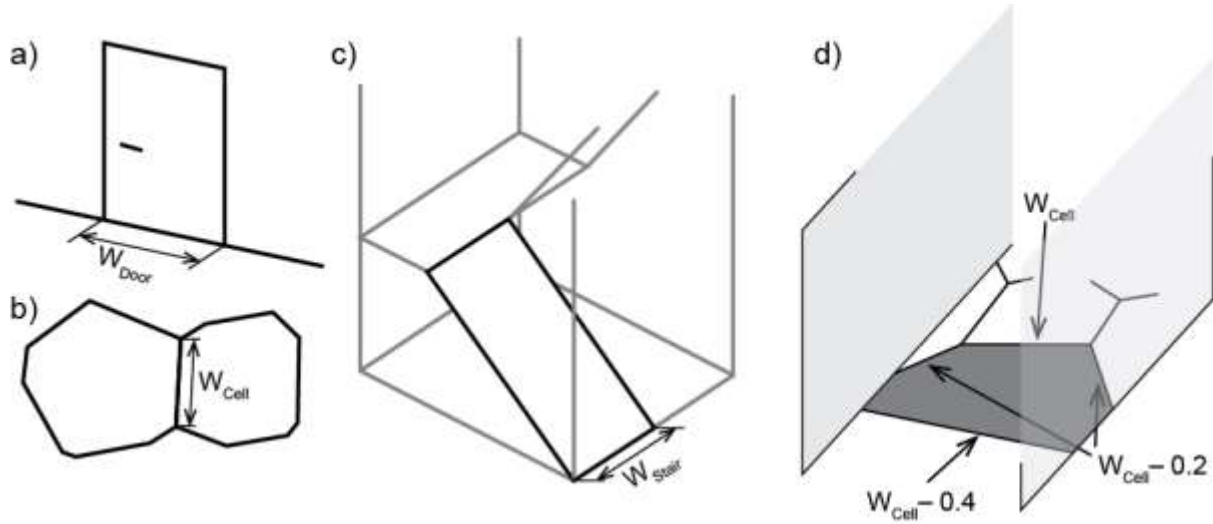
Figure 2. Actual and effective width in maximum flow calculations: a) actual width of a door W_{Door}; b) actual width of a connection between cells W_{Cell}; c) actual width of a stair flight W_{Stair}; d) effective width of connections between cells adjacent to walls.

In each iteration, the distance covered by an agent is calculated based on the agent's actual speed and the iteration time step according to the formula:

$$D(a) = S(a)\, T_{step} \qquad\qquad (4)$$

where $D(a)$ is the distance covered by the agent, $S(a)$ is the agent's speed calculated from Eq. 2 and T_{step} is the simulation input parameter giving the iteration time step.

The distance is used to determine a new location of and agent along the route. Before the agent is placed in the new location, the occupancy and flow criteria must be met. These tests are performed only if the new location appears in a cell different from the current one. Technically, it is checked if the agent passes the midpoint of a link.

Firstly, it is checked if the current occupancy OCC_N of the next node is smaller than its maximum capacity CAP_N, where the occupancy determines the number of agents occupying the node. As it would be impossible to move through nodes when the capacity is smaller than one, an unused capacity (i.e. maximum capacity reduced by current occupancy) of neighbouring cells is temporarily added to the node before the occupancy criterion is checked. Secondly, it is checked if the actual flow of agents OCC_L through the link does not exceed the maximum allowed flow CAP_L.

If these two tests are passed, then the agent is allowed to move from one cell to another. In other words, the cell represented by the second node can accommodate the agent. The occupancy attribute OCC_N is increased by one for the second node, and reduced by one for the previous node. In addition, the actual flow attribute for the link OCC_L is increased by one. In order to determine the actual flow of agents, OCC_L for each link in the network is reduced by CAP_L multiplied by T_{step} after each iteration, when all agents are moved to the new locations. The minimum value of OCC_L is zero.

Conversely, if one of the tests is not passed, then the agent needs to wait. If the maximum waiting time T_1 specified for individual agents is achieved, then the agent is marked for rerouting. Additionally, if the agent's waiting time exceeds the global maximum waiting time T_2 defined for

simulation or if the ratio OCC_L/CAP_L of the current link associated with the agent is greater than a certain threshold T_3, the current link is temporarily marked as blocked and is not considered for path finding. New routes for the marked agents are calculated after the current iteration is completed, and at the same time the individual agent's waiting time is increased by a specific value (e.g. 5 s). Once the agent is allowed to move, the individual waiting time is reset to the initial value.

After each iteration, the database of densities is updated. The density value, OCC_N/CAP_N, is calculated for each node. Because this value is attached as an attribute to the related nodes, the navigable network is considered as a database. It is used in calculations of optimal routes for search and rescue teams.

It should be noted that in the reported version of simulation, the hazard influence is not taken into consideration in route calculations for agents. It is assumed that agents have no knowledge of hazard locations and they seek to find the quickest way to exit the building. However, routes for search and rescue teams are calculated taking into consideration hazard information, as they are expected to have access to the central management system where hazard influence in the building is mapped in a digital model.

The method described above is formalized in the following algorithms 2 and 3. Algorithm 2 is the main evacuation simulation implemented as an iterative process based on the list of agents. It stops when all the agents reach their destinations outside the building. Algorithm 3 describes a single move of an agent made within time T_{step} in step 2 of Algorithm 2. Each agent is represented as an object a with the following properties: L – current link, N – current node, R – egress route, P_{xyz} – current location, $wait$ – current waiting time, $maxWait$ – maximum waiting time, $NewRouteRequired$ – true/false.

Algorithm 2: Evacuation simulation

Input: Navigable network NN

List of agents LA

Time step T_{step}

1. Calculate routes for agents in LA
2. For each agent a in LA call MoveAgent(a, T_{step})
3. For each agent a in LA calculate new egress routes $a.R$ if $a.NewRouteRequired=True$
4. For each link L in NN update $OCC_L = max(0, OCC_L – CAP_L*T_{step})$ and if OCC_L=0 then $L.blocked$=False
5. For each node N in NN determine current density OCC_N / CAP_N
6. If any agent in LA did not reach the route end goto 2
7. Algorithm stops

Algorithm 3: Move Agent

Input: Agent a

Time step T_{step}

Agent's waiting threshold T_1

Maximum waiting time T_2

OCC/CAP threshold T_3

Output: Single move on an agent within T_{step}

1. Calculate distance to cover: $D(a) = S(a) * T_{step}$ (see Eqs. 2 and 3)
2. $L \leftarrow a.L$ (get the current link)
3. $N_{Next} \leftarrow$ next node on the route $a.R$
4. if $D(a) <$ Distance($a.P_{xyz}$, N_{Next}) then goto 10
5. $L \leftarrow$ next link on the route a.R
6. If $OCC_L >= CAP_L * T_{step}$ then $a.wait = a.wait+1$ and goto 18
7. $D(a) = D(a) -$ Distance($a.P_{xyz}$, N_{Next})
8. $a.L \leftarrow L$ and $a.P_{xyz} \leftarrow N_{Next}$
9. goto 2
10. Calculate new location P_{xyz} on the L (at distance $D(a)$ from the previous route node)
11. if ($a.N == N_{Next}$) or (Distance(P_{xyz}, N_{Next}) > Distance(P_{xyz}, $a.N$)) then goto 17
12. If $OCC_L >= CAP_L * T_{step}$ then $a.wait=a.wait+1$ and goto 18
13. If $OCC_N >= CAP_N$ then $a.wait=a.wait+1$ and goto 18
14. $OCC_N = OCC_N-1$
15. $a.N \leftarrow N_{Next}$
16. $OCC_N = OCC_N+1$ and $OCC_L = OCC_L+1$
17. $a.P_{xyz} \leftarrow P_{xyz}$, $a.wait=0$ and $a.maxWait=T_1$
18. if $a.wait * T_{step} > a.maxWait$ then $a.NewRouteRequired$=True and $a.maxWait=a.maxWait+T_1$
19. if $a.wait * T_{step} > T_2$ or $OCC_L / CAP_L > T_3$ then $L.blocked$=True
20. Algorithm stops

2.2 Agent-based simulation scenarios for route calculation

The algorithms proposed in the previous section will be used in three simulations, where optimal routes for rescue teams are calculated and different evacuees' densities are considered: a) static with constant values for four stages of evacuation; b) semi-dynamic with values obtained during evacuation simulation; c) dynamic with values stored in a database.

In the static simulation a), constant density values are assigned to nodes in selected areas of the navigable model. They are based on typical evacuation scenarios, where occupancy in a building is roughly estimated [18]. Four scenarios are taken into consideration: Scenario 1 is an initial stage of evacuation involving evacuees being located in corridors and staircases after leaving their rooms. Density values (people/m^2) in this scenario are as follows: 0.0 for any room, 1.0 for all staircases, 0.5 for corridors. The middle stage of evacuation is reflected in Scenario 2 with the following densities: 3.0 for staircases, 2.0 for the areas on the ground floor between staircases and entrances to the building, as well as in the areas on floor 5 where people switch staircases, and 0.0 for all other spaces. In the final stage of evacuation, Scenario 3, these values are: 3.0 for staircases between the ground floor and floor 4, 2.0 in the areas on the ground floor between staircases and entrances to the building, 0.0 for all other spaces. The empty building after completion of evacuation is Scenario 4. Figure 3 illustrates all four scenarios.
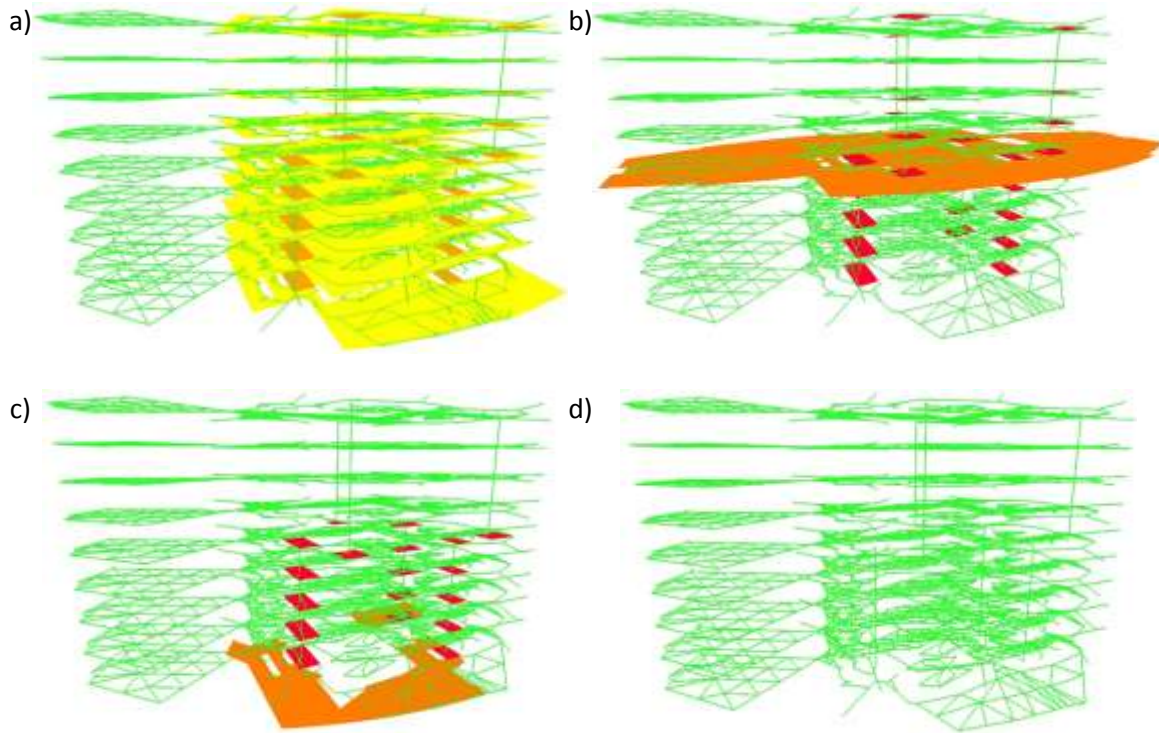
Figure 3. Constant values of evacuees' densities for different evacuation stages: a) Scenario 1 – initial stage; b) Scenario 2 – middle stage; c) Scenario 3 – final stage; d) Scenario 4 – empty building after evacuation. Different colours represent different density: from yellow (0.5 people/m^2) to red (3 people/m^2)

In the semi-dynamic simulation b), density values are calculated during evacuation simulation. These values represent a static picture of the current state of evacuation. The evacuation process is stopped and actual evacuees' densities are used in path finding for a rescue team. The values of densities do not change in the process of path calculations.

The dynamic simulation c) also uses densities calculated in evacuation simulation, which are stored in the database together with temporal information. After each iteration of evacuation simulation, densities of agents at network nodes are attached at these nodes. The density is the ratio of the number of agents associated with the node to the area represented by the node. Updates are performed every T_{step} period. This means that evacuees' densities during the evacuation process may be determined at any time. These values are used for optimal path calculations for rescue teams. A variable defining a starting time is an input parameter for the path-finding algorithm, where time 0 is the beginning of the evacuation. Densities from the database are dynamically selected depending on path calculation progress.

3. Testing setup, results and discussion

3.1. Agent-based evacuation

In the presented experiment, agent-based simulation of evacuation from a building described in Section 2 is performed with an initial density value 0.07 p/m^2. This results in the total number of agents equal to 990, which means 110 agents per floor on average. Agents associated with a certain

space are distributed evenly among network nodes belonging to that space. In case the number of agents is different than the number of nodes, the remaining agents are put at random nodes. It is important to notice that too big value of initial density will result in cell occupancy greater than 3.5 people/m^2, which may cause congestions at the initial stage of evacuation simulation.

Figure 4 shows the initial agent distribution in one of the rooms with floor area 288.88 m^2. The number of agents is equal to 21. It should be noted that agents represented as spheres may overlap and be put at the same location, because in this research the focus is put on a total cell occupancy rather than collisions among individual agents. Numbers at spheres indicate a number of agents located in a cell.
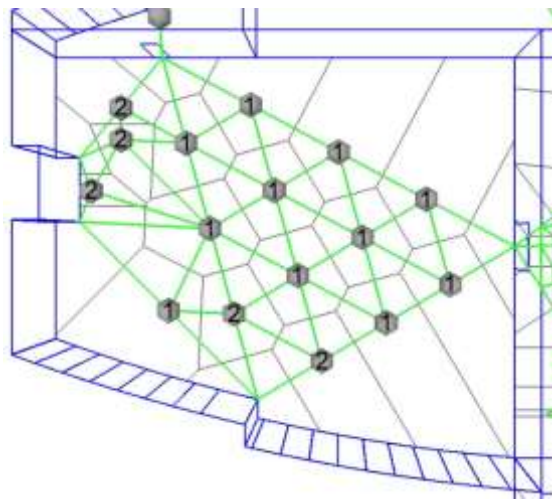
Figure 4. Initial distribution of agents in a selected room. Numbers indicate the agent occupancy of cells associated with network nodes. The navigable network, tessellation and building geometry are shown in green, grey and blue, respectively.

Other parameters are as follows: T_{step} = 1 s, T_1 = 10 s, T_2 = 30 s, T_3 = 3. They indicate that a new location for each agent is calculated every 1 s, a new path for an agent is calculated after 10 s of waiting at the same location caused by congestions, and a link is temporary marked as blocked if an agent is stuck at this link for more than 30 s or the ratio of link occupancy to maximum capacity is greater than 3. Therefore, an agent's behaviour may be considered a waiting strategy followed by obstacle avoidance in crowded areas, where some passages are too congested to follow the initial path.

The building becomes empty after 502 s. The evacuation progress is shown in Figure 5, while distribution of agents reflected in the navigable network is illustrated in Figure 6. Original locations of agents are random, but agents are evenly distributed in rooms (see Figure 6a). After some time, most evacuees are in staircases (see Figure 6b). However, at floor 5, where top-floor staircases meet the lower-floor ones, agents must move to another staircase. This causes congestions at staircase doors. New paths for agents, which cannot move, are calculated. As a result, agents move to less dense areas and change the originally selected staircase (see Figure 6c). Subsequently, most of the agents use the lower-floor staircases without changing them because densities are lower and

smooth movement without congestion is possible (see Figure 6d). Finally, agents leave the staircases (see Figure 6e) and the building (see Figure 6f).
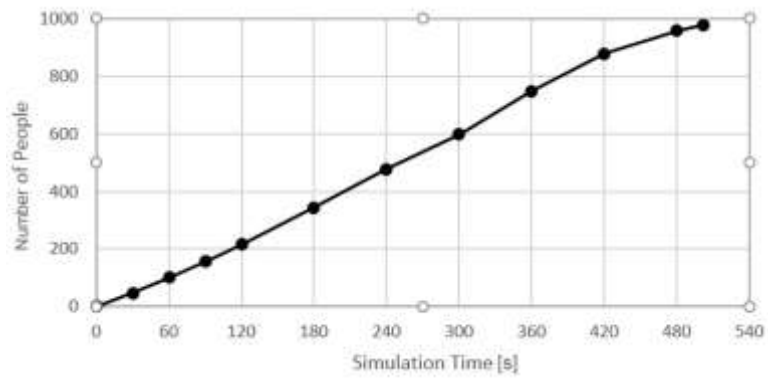

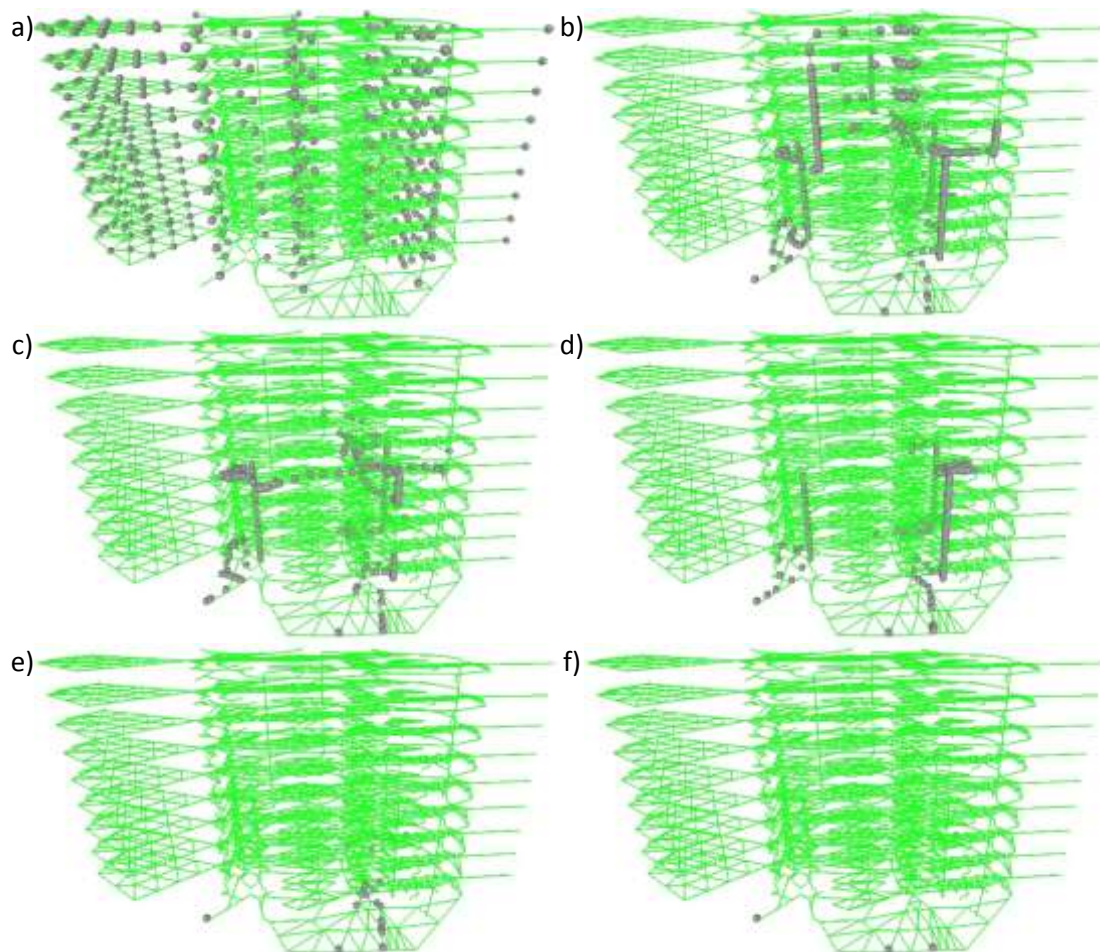Figure 5. Number of people evacuated during simulation

Figure 6. Agent distribution during evacuation simulation after: a) 0 s (original agent's distribution); b) 120 s; c) 240 s; d) 360 s; e) 480 s; f) 502 s (end of evacuation).

The evacuation time in this experiment, i.e. 502 s, compared to the results obtained in other studies [3,8] shows it is higher. Two nine-storey office buildings were evacuated in practice in 270 s and 330 s, while simulated evacuation time was 216 s and 360 s respectively [3]. The biggest difference between the buildings is the area of staircases (total staircase area per floor: 64 m2 and 38 m2 respectively), which influences the evacuation time. The total area of staircases in the building used in this paper is about 40 m^2/floor. The difference may be caused by a different building layout, especially the number and size of staircases and doors, which are critical elements influencing evacuation in high-rise buildings. Also, bottlenecks with a high density of agents were observed at doors inside the building, which was not investigated in other studies. Congestions significantly slow down the movement of people and, in result, the total evacuation time is higher.

3.2. Optimal routing for rescue teams

Optimal paths for a rescue team are calculated based on the algorithm proposed by Zverovich, et al. [18]. Five hazard locations are introduced on various floors in the model. Different starting times of the rescue operation are considered, where time 0 is the beginning of evacuation simulation. Travel times, safety levels and shapes of paths depend on agents' densities in the building, which are changing over time. In this experiment, the origin location for path calculation is outside of the building, and the rescue team has to reach a room on floor 8. Figure 7 illustrates travel times of quickest paths for three different evacuation simulations: static, semi-dynamic and dynamic, where hazard influence is not taken into consideration in path calculations.

Because in the static simulation there are only four scenarios representing evacuation stages, these scenarios were distributed over simulation time based on their description illustrated in Figure 3 and evacuees' distribution (see Figure 6). Therefore, Scenario 1 starts at time 60 s after the agents leave the rooms and they are located in corridors and staircases. Scenario 2 starts at time 180 s when agents are in staircases and on the floor, where they change staircases. Scenario 3 starts at time 360 s when all top-floor staircases are empty and agents are located in the ground-floor staircases and in areas between staircases and exits from the building. Scenario 4 is the empty building after evacuation.
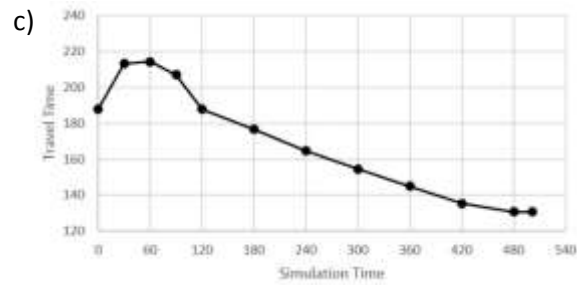
a)

b)

c)
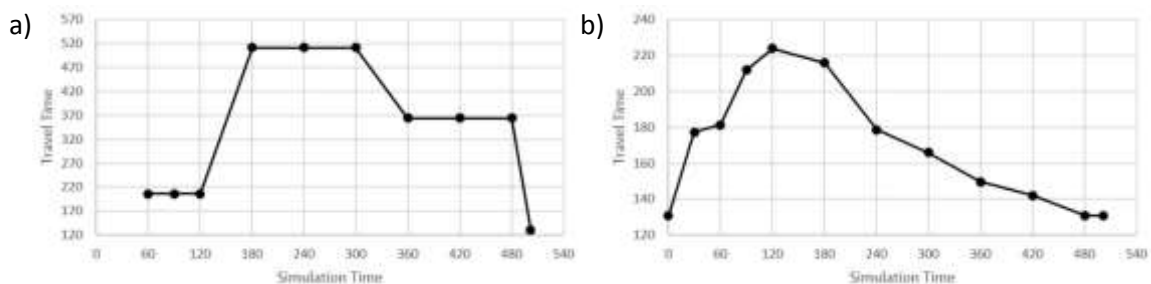
Figure 7. Quickest path travel time [s] depending on starting simulation time: a) static simulation; b) semi-dynamic simulation; c) dynamic simulation.

It should be noted that the travel time in the static simulation in Scenarios 2 and 3 is much higher (respectively 514 s and 364 s in Figure 7a) than in other simulations (maximum 224 s in semi-dynamic in Figure 7b , and 214 s in dynamic in Figure 7c), where densities are calculated in the evacuation simulation. The reason for such a difference is that in the static scenario densities in staircases are constant (3.0 p/m^2) throughout the optimal path-finding process, which in a real-life scenario is rather unrealistic. Evacuees are moving downstairs and gradually releasing spaces on higher floors. In the density database, density 3.0 p/m^2 and above are observed at some staircase nodes and only for a short period. Reducing a level of density in staircases to 2.0 p/m^2, the travel time decreases to 318 s and 253 s, respectively, for Scenarios 2 and 3 of the static simulation.

Figure 8 shows travel times of optimal paths, where a safety criterion has the highest weight and is followed by time and complexity criteria. Similar to the quickest paths, the optimal paths obtained for the static simulation have travel time much higher than in the case of semi-dynamic and dynamic simulations. In addition, optimal travel times for the semi-dynamic and dynamic simulations are higher than for the quickest paths. The paths are longer, but they are located further from the hazard locations hence are safer. Hazard proximity indices [19] calculated for optimal paths are shown in Figure 9. The higher the proximity index, the safer the path is.
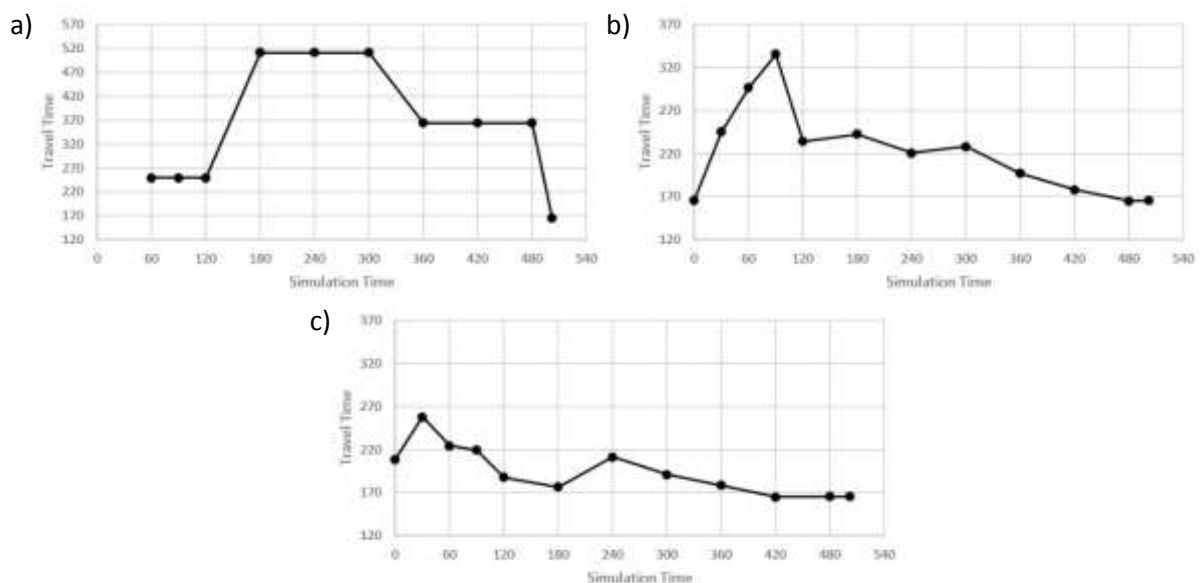
a)

b)

c)

Figure 8. Optimal path travel time depending on starting simulation time: a) static simulation; b) semi-dynamic simulation; c) dynamic simulation.

It should be noted that the safety of paths in dynamic simulation decreases after about 90 s after evacuation had started. It is caused by high density values for all staircases, which reduce the speed of rescue teams. This means that they need to stay longer in hazardous conditions. Then the safety is improved in 240 s because top floors are almost empty and the movement of rescue teams is not delayed.
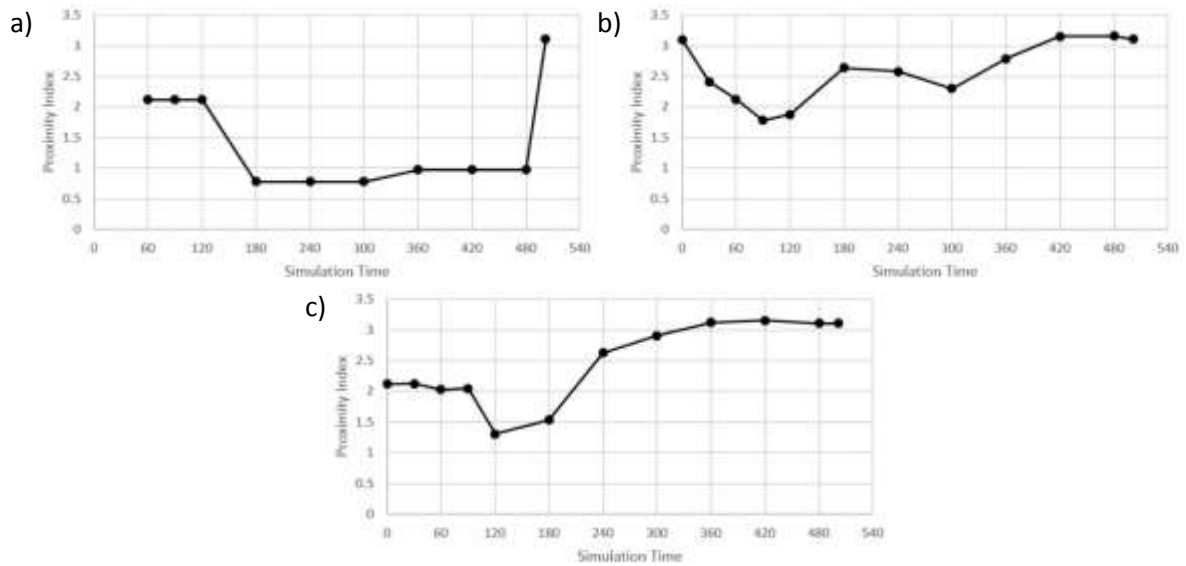

Figure 9. Optimal path hazard proximity indices depending on starting simulation time: a) static simulation; b) semi-dynamic simulation; c) dynamic simulation.

Selected optimal paths are shown in Figure 10 – static simulation, Figure 11 – semi-dynamic simulation, and Figure 12 – dynamic simulation. The shape of paths depends only on evacuees' distribution, as hazard locations and other parameters remain unchanged.
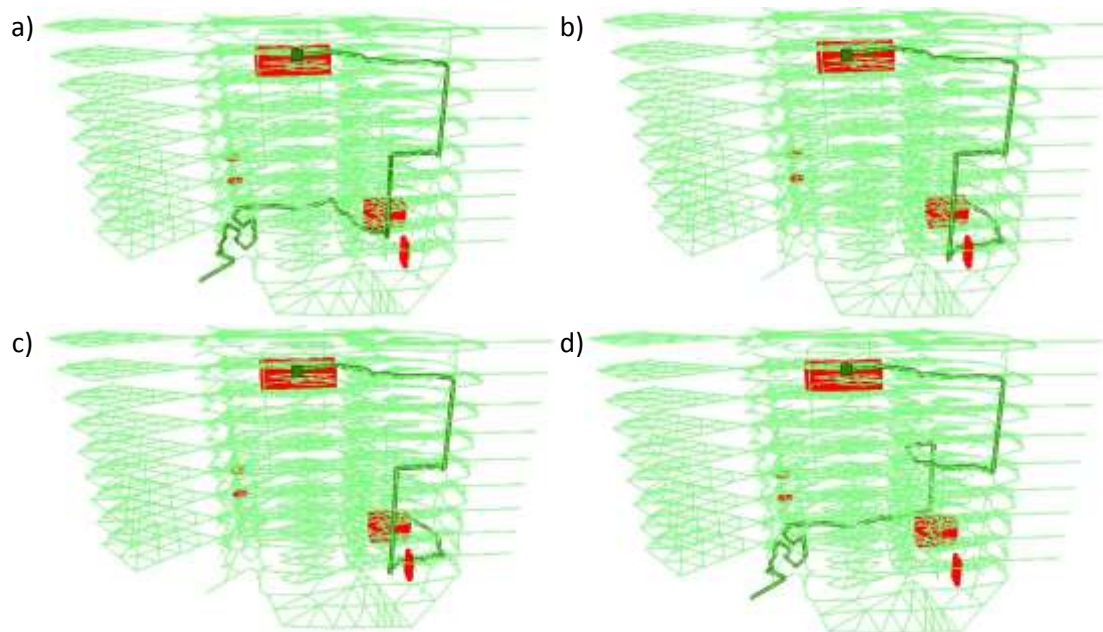

14

Figure 10. Selected optimal paths in the static simulation calculated for a) Scenario 1 – initial stage; b) Scenario 2 – middle stage; c) Scenario 3 – final stage; d) Scenario 4 – empty building.
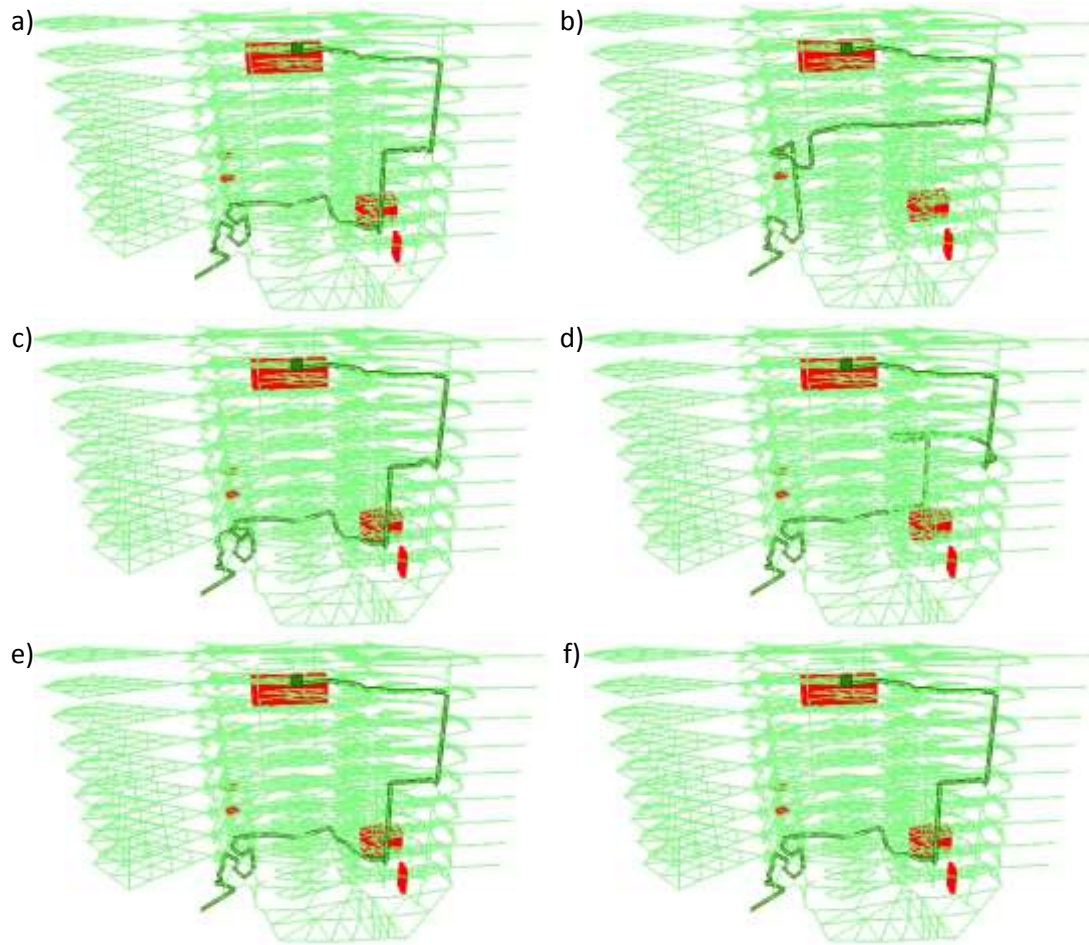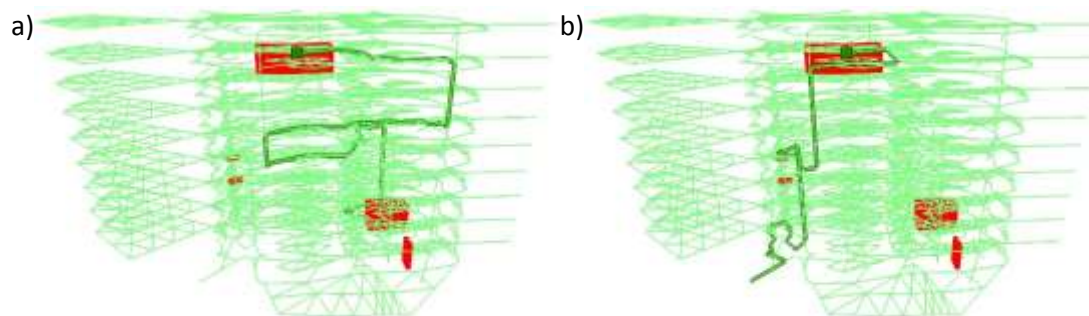

Figure 11. Selected optimal paths in the semi-dynamic simulation calculated after a) 0 s; b) 120 s; c) 240 s; d) 360 s; e) 480s; f) 502 s from the beginning of evacuation.
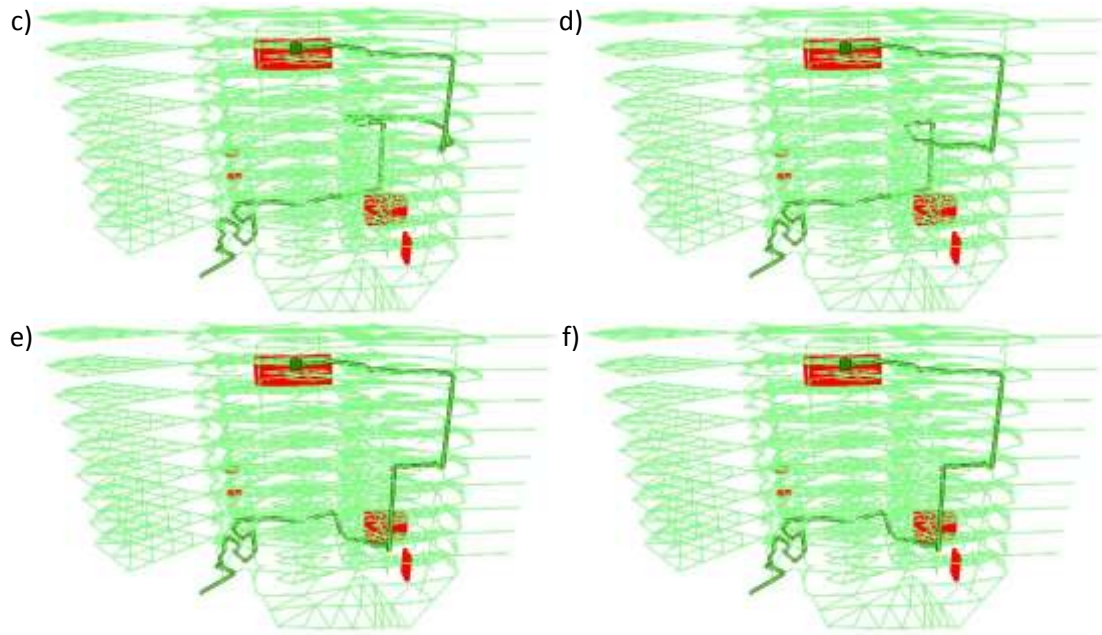
Figure 12. Selected optimal paths in the dynamic simulation calculated after a) 0 s; b) 120 s; c) 240 s; d) 360 s; e) 480s; f) 502 s from the beginning of evacuation.

Travel times obtained in Scenarios 2 and 3 of the static simulation are much higher than in semi-dynamic and dynamic simulations. Densities in these scenarios are higher in most of the critical spaces, such as staircases and near the exits, than in case of other simulations. Although similar high density values are experienced at some nodes, the occupation level in the rest of the navigable network is relatively low in comparison to the static simulation.

To conclude this section, the similarity of simulations is compared. There are three simulations resulting in three time series. This comparison is important because simulations require different information and the computational complexities of the corresponding algorithms are not the same. It is generally accepted that when measuring the similarity of two time series, a simple, robust and accurate method is the Euclidean distance (see, e.g. [15]). More importantly, Serrà and Arcos [10] showed that the Euclidean distance is the best performing measure between two time series.

For two time series x and y, let x_i and y_i denote their i-th elements, respectively, and let M be the length of the time series. Then, the Euclidean distance between the two time series can be calculated using the following formula:

$$L_2(x, y) = \sqrt{\sum_{i=1}^{M} (x_i - y_i)^2}$$

Let us denote our three time series by **S** (static), **SD** (semi-dynamic) and **D** (dynamic). Because the former time series starts at time 60, we calculate the Euclidean distances between series for the time interval [60; 502]:

$$L_2(\mathbf{S},\mathbf{SD}) = 591.5, \quad L_2(\mathbf{S},\mathbf{D}) = 651.4, \quad L_2(\mathbf{SD},\mathbf{D}) = 164.7$$

When comparing time series, it is often recommended to normalise them for tolerating different ranges. Therefore, we will also compare the normalized time series. The first pre-processing step is to subtract from each time series, its mean value. The second step is to normalise the amplitude, i.e. divide by the standard deviation of the series. This approach yields the following normalized Euclidean distances:

$$L_2(\mathbf{S_N}, \mathbf{SD_N}) = 4.5, \quad L_2(\mathbf{S_N}, \mathbf{D_N}) = 4.3, \quad L_2(\mathbf{SD_N}, \mathbf{D_N}) = 1.6$$

Both sets of Euclidean distances reveal that the static time series is rather far from two other time series, whereas the semi-dynamic time series is similar to some extent to the dynamic time series.

4. Conclusions

In this paper, a method for determining evacuees' distribution in buildings was proposed. Agent-based evacuation simulations were performed in the navigable network generated from 3D BIM model, where individual egress routes for agents are calculated and their movements are determined by actual densities and the flow in the network. The results of the simulation show that evacuation times are comparable with previous findings by other studies. It was achieved without taking into consideration parameters, which are normally included in micro-scale simulations, e.g. an agent's size or precise location. Our simulations are based on effective algorithms, which are more intuitive and simpler for implementation.

Evacuees' densities calculated in simulations are then used to generate optimal paths for rescue teams. Three simulation scenarios were considered: static with constant density values specified for four stages of evacuation, semi-dynamic with densities representing an actual people distribution in a building during evacuation simulation, and dynamic with temporal distribution of evacuees, stored in a database and dynamically used in optimal path calculation.

The static simulation offers an occupancy estimation and it is better suited for simulations, when there is no or little knowledge about building occupancy, while the semi-dynamic and dynamic simulations are based on the initial distribution of people and they offer scenarios that are more realistic. The semi-dynamic simulation is better tailored fit for emergency response scenarios, where the actual occupancy can be estimated based on occupancy sensors, and where rapid calculation of optimal paths for rescue teams is needed. The dynamic simulation may be more useful for emergency preparedness applications, where different evacuation scenarios are tested. A different building design or initial distribution of people depending on indoor space functionality may be analysed in order to achieve safer buildings, and effective building management.

In future developments, it would be desirable to use real occupancy information from sensors, which would increase the accuracy of simulations. In addition, algorithms for individual path finding for agents and their movement may be improved by adopting real-time information about people locations in buildings.

Acknowledgment

References

[1]   P. Boguslawski, C. Gold, The Dual Half-Edge—A Topological Primal/Dual Data Structure and Construction Operators for Modelling and Manipulating Cell Complexes, ISPRS International Journal of Geo-Information 5 (2) (2016) 19.

[2]   P. Boguslawski, L. Mahdjoubi, V. Zverovich, F. Fadli, Automated Construction of Variable Density Navigable Networks in a 3D Indoor Environment for Emergency Response, Automation in Construction 72 (2016) 115-128.

[3]   M. Galbreath, Time of evacuation by Stairs in High Buildings, National Research Council of Canada Division of Building Research, 1969, pp. 3-6.

[4]   H.-S. Gan, K.-F. Richter, M. Shi, S. Winter, Integration of simulation and optimization for evacuation planning, Simulation Modelling Practice and Theory 67 (Supplement C) (2016) 59-73.

[5]   H.W. Hamacher, S.A. Tjandra, Mathematical Modelling of Evacuation Problems: A State of Art, 2001.

[6]   U. Isikdag, S. Zlatanova, J. Underwood, A BIM-Oriented Model for supporting indoor navigation requirements, Computers, Environment and Urban Systems 41 (Supplement C) (2013) 112-123.

[7]   H.E. Nelson, H.A. MacLennan, Emergency Movement, in: P.J. DiNenno, C.L. Beyler, R.L.P. Custer, W.D. Walton, J.M. Watts, D. Drysdale, J.R. Hall (Eds.), SFPE handbook of fire protection engineering, Society of Fire Protection Engineers, Boston, MA, USA and National Fire Protection Association, Quincy, MA, USA, 1995, pp. 3/286-283/295.

[8]   J. Pauls, Movement of people, in: P.J. DiNenno, C.L. Beyler, R.L.P. Custer, W.D. Walton, J.M. Watts, D. Drysdale, J.R. Hall (Eds.), SFPE handbook of fire protection engineering, Society of Fire Protection Engineers, Boston, MA, USA and National Fire Protection Association, Quincy, MA, USA, 1995, pp. 3/263-263/285.

[9]   M. Saeed Osman, B. Ram, Two-phase evacuation route planning approach using combined path networks for buildings and roads, Computers & Industrial Engineering 65 (2) (2013) 233-245.

[10] J. Serrà, J.L. Arcos, An empirical evaluation of similarity measures for time series classification, Knowledge-Based Systems 67 (2014) 305-314.

[11] T.-A. Teo, K.-H. Cho, BIM-oriented indoor network model for indoor and outdoor combined route planning, Advanced Engineering Informatics 30 (3) (2016) 268-282.

[12] P. Thompson, D. Nilsson, K. Boyce, D. McGrath, Evacuation models are running out of time, Fire Safety Journal 78 (2015) 251-261.

[13] S. Tzemos, R.A. Burnett, Use of GIS in the Federal Emergency Management Information System (FEMIS), Proceeding of 1995 ESRI user conference, Palm Springs, California, 1995.

[14] A. Vanclooster, P. De Maeyer, V. Fack, N. Van de Weghe, Calculating Least Risk Paths in 3D Indoor Space, in: U. Isikdag (Ed.), Innovations in 3D Geo-Information Sciences, Springer International Publishing, 2014, pp. 13-31.

[15] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, E. Keogh, Experimental comparison of representation methods and distance measures for time series data, Data Mining and Knowledge Discovery 26 (2) (2013) 275-309.

[16] C.-H. Wu, L.-C. Chen, 3D spatial information for fire-fighting search and rescue route analysis within buildings, Fire Safety Journal 48 (Supplement C) (2012) 21-29.

[17] M. Xu, I. Hijazi, A. Mebarki, R.E. Meouche, M. Abune'meh, Indoor guided evacuation: TIN for graph generation and crowd evacuation, Geomatics, Natural Hazards and Risk 7 (sup1) (2016) 47-56.

[18] V. Zverovich, L. Mahdjoubi, P. Boguslawski, F. Fadli, Analytic Prioritization of Indoor Routes for Search and Rescue Operations in Hazardous Environments, Computer-Aided Civil and Infrastructure Engineering (2017).

[19] V. Zverovich, L. Mahdjoubi, P. Boguslawski, F. Fadli, H. Barki, Emergency Response in Complex Buildings: Automated Selection of Safest and Balanced Routes, Computer-Aided Civil and Infrastructure Engineering 31 (8) (2016) 617-632.