Intelligent Emergency Evacuation Systems for Buildings

Zoi, Z.K, Kelepiri
Postgraduate, School of Informatics
Aristotle University of Thessaloniki
zoekele28@gmail.com

Athanasios, A.L, Lentzas
PhD Candidate, School of Informatics
Aristotle University of Thessaloniki
alentzas@csd.auth.gr

Dimitrios, D.V, Vrakas
Professor, School of Informatics
Aristotle University of Thessaloniki
dvrakas@csd.auth.gr

ABSTRACT

The purpose of the present work is the planning of evacuation of a building in emergency conditions, the simulation of the building and the finding of the shortest path to the safest exit. The algorithm for finding and customizing the path is written in Java. The path planning algorithm used here is Dijkstra. Respectively, the evacuation building simulation is written in JavaScript with the help of Html and CSS, which is displayed in a dynamic html web page. In the building simulation the user can add all the information about the structure of the building, which are stored in an xml file. The xml file is used to find the shortest path using java. After such an adjustment the new chosen path has the lowest possible cost to the goal.

KEYWORDS

emergency evacuation, graphs, intelligent evacuation, optimal path, dijkstra

ACM Reference Format:

Zoi, Z.K, Kelepiri, Athanasios, A.L, Lentzas, and Dimitrios, D.V, Vrakas. 2022. Intelligent Emergency Evacuation Systems for Buildings. In *26th Pan-Hellenic Conference on Informatics (PCI 2022), November 25–27, 2022, Athens, Greece.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3575879.3575966

1 INTRODUCTION

Nowadays the design of effective evacuation plans is one of the most urgent research topics. Proper emergency evacuation planning is important to ensure the integrity and efficiency of transmission networks in case of natural hazards. A good evacuation plan can save lives and prevent congestion. This study presents case-tested models, to develop effective emergency evacuation plans.

One of the most crucial challenges for a large-scale emergency evacuation is the limited number and capacity of evacuation routes, which may not be sufficient to accommodate the traffic demand into the evacuation zone. The distribution of disabled people among existing emergency shelters is also a challenge, considering the limited capacity of available shelters and the special needs of certain groups. A proper evacuation plan is even more important for the elderly, as many of them may require considerable time to get into the designated emergency shelter.

PCI 2022, November 25–27, 2022, Athens, Greece

ACM ISBN 978-1-4503-9854-1/22/11…$15.00
https://doi.org/10.1145/3575879.3575966

Recent studies on emergency evacuation are oriented towards the optimization of evacuation models. These models can improve the use of evacuation routes, reduce congestion by minimizing evacuation travel time, and help distribute evacuees among available emergency shelters. Therefore, decision models should focus on the design and development of advanced optimization-based tools and algorithms to develop effective emergency evacuation plans. The whole process aims to facilitate natural hazard preparedness work, with the main objective of minimizing the total travel time of people evacuated from a danger zone.

The structure of the paper is as follows. After a short introduction on section 1, related work is discussed on section 2 and section 3 presents our work and the experiments performed. Finally, conclusions are drawn and future work is discussed on section 4.

2 RELATED WORK

There is a plethora of approaches for solving the emergency evacuation problem. GIS (Geographic Information Systems) and BIM analysis to determine the shortest and safest paths for people at risk of fire in order to simulate their movement in the building [1] is one of them. Different scenarios were evaluated for the shortest and safest paths using the Dijkstra algorithm considering different origins and destination points in the 3D indoor environment to assist rescue operations.

A study presented by [2] investigated whether elevators can be used as an emergency alternative for evacuation in apartment buildings. The parameters were divided into three categories: human behavior, fire hazards, and functional elevator mechanism. The risk assessment approach was developed based on the Multi-Objectives Decision Analysis (MODA) method. The result of the research showed that elevators, as an alternative case of evacuation, can improve the efficiency of evacuation, especially for the elderly and people with special needs.

Transportation systems and HVAC airflow for efficient building evacuation is presented by [3]. The paths selected for exit must be supplied with fresh air through the HVAC control system. The problem is to minimize the exit time and choose the shortest and safest evacuation paths. A solution was proposed by iteratively using the maxflow algorithm establishing the HVAC system for discharge time improvements.

A fire emergency crowd evacuation model for large-scale public buildings is proposed on [4]. The model incorporates numerical fire simulations and a shortest path search model. Navigational grids for each floor are connected to form 3D grids for complex architectural interiors. The numerical fire simulation provides smoke diffusion fluxes and temperature field distribution. The A* algorithm, with its heuristic model function, is applied to determine the safest path and the shortest evacuation route, avoiding the danger zones on different floors.

The problem of evacuation has been addressed with methods such as BFS as well [5]. More specifically a museum evacuation plan was based on best first search with genetic algorithms used for optimization. Similarly, an evacuation algorithm with multiple exits under the limiting condition of fire smoke and path capacity, was presented by [6]. Given the capacity of routes, the goal of emergency evacuation is to minimize the total evacuation time for all people. The algorithm used is based on the Dijkstra algorithm and must maintain the FIFO (First In First Out) property.

The authors of [7], advocated the use of architectures such as Multi-Agent Systems (MAS) to develop evacuation plans before an actual accident. The approach, based on model-based principles, involves using a graphical editor to design evacuation models, automatically generate the evacuation plan code, and run the generated code on a MAS platform. The results show that the approach provides increased speed, less effort, a high level of abstraction, and greater flexibility and productivity.

A robot guide that can help people evacuate is a solution that may decrease time and casualties while evacuating large buildings. The main issue is whether humans can trust a robot, and how we can improve that trust [8]. A 3D simulation environment was created and determined to what extent a human would follow a robot to emergency exits. Survey feedback and quantitative scenario results were collected on two different robot designs.

A solution for practical problems related to emergency evacuation in construction using a simulation system based on Artificial Neural Networks (ANN) was presented by [9]. Machine learning allows us to accurately simulate the behavior of all individuals during the evacuation, their reaction to obstacles and consequently, make changes to the building plans where necessary. This program is especially important for unique multifunctional facilities, for example, sports stadiums or shopping and entertainment centers.

The emergency fire evacuation problem for an existing school in the hinterland of Milan was studied by [10]. A virtual reality-based system for simulating fire emergency response will be developed starting from the BIM model and then progressing to a platform that will ensure interaction between human and virtual simulation environment. Flame and smoke propagation is based on numerical fire simulation using Computational Fluid Dynamics (CDF) software.

Post-earthquake observations reveal that evacuation affects the number of occupant casualties during earthquakes, few current studies consider occupant movements in the building in accident prediction procedures. To bridge this knowledge gap [11] presented a numerical simulation method using refined cellular automata model, which can describe various dynamic behaviors and building dimensions.

Lastly, an approach that combines a multi-agent model with fuzzy logic for smooth and successful handling of multiple attributes of everyone at exit was presented by [12]. The developed simulation system considers situations where the crowd is blocked inside a building or zone during a disaster. Each agent possesses different characteristics to realistically simulate a human by incorporating psychology, sociology, mood, reaction, etc.

3 OUR WORK

Our work consists of two important parts. The first part concerns the visualization of the building in the form of a graph, realized via a web page. The building architecture is designed with a graph, the appropriate information necessary for evacuation is added and then this information is saved in an xml file. The second part aims to solve the evacuation problem using the Dijkstra algorithm and mapping the shortest path for each node. In this part, the information generated by the first part is used, stored in the xml file, and based on this data, the algorithm is implemented.

In particular, the visualization of the building as a graph consisting of nodes and edges is realized using a web interface. The library used (vis.js), allows easy manipulation and interaction with large amounts of data. Node and edges operations such as adding, updating, and deleting are supported by the aforementioned library and made available through the website.

The implementation for designing the graph and adding information for each node and edge is designed in an HTML web page, where each user can design the building corresponding to a graph, can add the required features and can export this information to a file xml format. In addition, it can import an xml file and draw the graph based on the information contained in this file. Below is the illustration of the website (Figure 1).

After the creation of edges and nodes, it is necessary to add the relevant metadata that will be used for the quick and safe evacuation of the building. This is achieved by the existence of a pop-up window, which contains text fields for the user to enter the information they need, as well as two buttons. The pop-up appears every time someone clicks on the node or the edge respectively (Figure 2).

Finally, users can import or export the graph by selecting the appropriate function on the website. Exporting the graph will write the relevant information on an XML file. The file contains not only the connection between the nodes but all the metadata as well. Users are also able to import an XML file, thus continuing the construction of the graph or update it if necessary.

Regarding the second part of the present implementation, the Dijkstra algorithm is used to find the shortest evacuation path, which is implemented in Java. It is worth mentioning that the data in the xml file is read with the help of a java parser. To visualize the shortest path for each node, a library provided by the java language, JUNG is used. This library comes with several built-in layout algorithms, so the building data is used and the path for each node is graphically illustrated.

Dijkstra's original algorithm finds the shortest path between two given nodes, but a more common variant uses a single node as the "source" node and finds the shortest paths from the source to all other nodes in the graph, producing a shortest path. The algorithm examines all possible paths from the initial node (room) to the final one (emergency exit – shelter) and chooses the one with the lowest cost, i.e., the shortest. Each edge has its own weight, a cost (real number), which corresponds, for example, to a time interval or distance between two nodes. So, Dijkstra sums the costs of each edge from the initial node to the final one, choosing the shortest possible path.

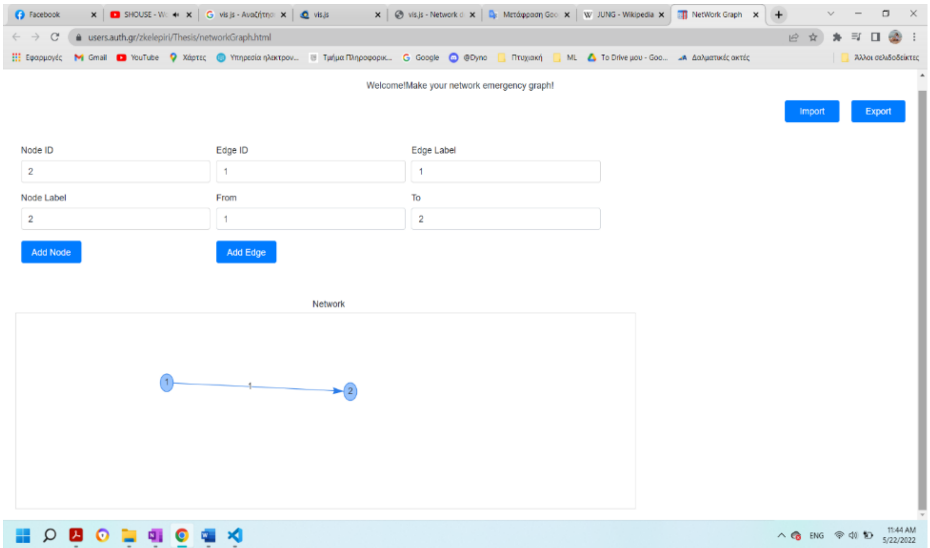

Figure 1: Implementation of website

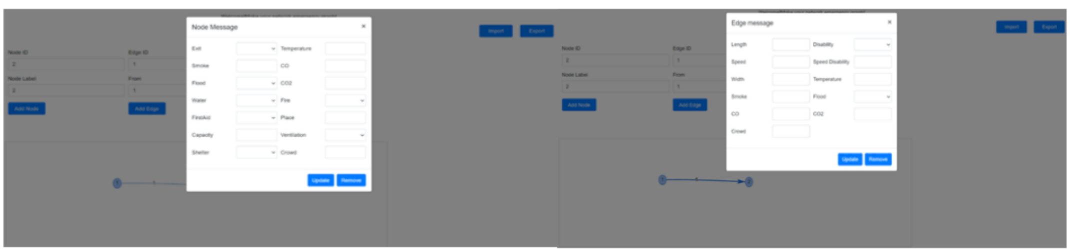

Figure 2: Pop up window for each node, edge

Regarding the evacuation implementation of a building, each room and corridor has some specific characteristics that the Dijkstra algorithm should consider finding the best solution (Table 1). Initially, it is necessary to know whether the building has an emergency exit or a shelter. If the building has an emergency exit, i.e. there is a node corresponding to the safest evacuation point, then the destination for each node based on the Dijkstra algorithm will be that one. In case there is no emergency exit but a shelter, then the final pathfinding destination will be the shelter. If there is none of the above the algorithm cannot find application and some other way of evacuating the building should be found.

In addition, for each node it should be considered whether it is in an emergency i.e, if there is a fire, a flood, or if carbon monoxide (CO), carbon dioxide (CO2) or the existence of a water supply are detected. In this case, the specific room should be excluded, the incoming edges at that node are deleted and only the outgoing edges remain, as it is considered unsuitable, since the people that pass through will be in danger. Therefore, the algorithm should not calculate the existence of the specific room and choose the path that will not need to visit the room, that is in an emergency state. Additionally, since there may be more than one safer place, a check is made as to which distance from each end node is the smallest and thus the nearest exit is selected. Possibly the path ultimately chosen will be longer than it would have been with the hazard node in place, but it will be the safest one for humans.

With reference to the edges (corridors) it is necessary to calculate the weight to accurately find the path to the exit. To calculate the weight, we need the average speed with which the runway can be crossed as well as its length. In this way, the time that takes someone to move from one room to another can be calculated for each corridor. So, Dijkstra will calculate the path based on the time required and therefore the shortest path will also be the one that takes the shortest travel time. It is worth mentioning that for each corridor it should be known whether it can be crossed by people with disabilities. If this is possible, then the average runway crossing speed will be different and the time interval will be calculated based on it.

3.1 Experiments

In order to evaluate our proposal, different scenarios were simulated. Scenarios used included both simple and complex evacuation plans. In the first experiment the user a relatively simple floor plan was used as a proof of concept (Figure 3). Selecting node 1 results in coloring the shortest path in red and printing it to the program console. Yellow is the selected node and green is the safest exit

Table 1: Characteristics for each node, edge

Node	Definition	Edge	Definition
Node Id	the id of each node	Edge Id	the id of each edge
Exit	existence of exit	From	start node
Shelter	existence of shelter	To	End node
Temperature	the temperature of each room (node)	Length	length of corridor
Smoke	existence of smoke	Width	width of corridor
CO	detection of CO	People with disabilities	existence of people with disabilities
CO_2	detection of CO_2	Speed	speed of each person
Flood	detection of flood	Disability Speed	speed of each disability person
Water	detection of water	Temperature	temperature of each corridor (edge)
Fire extinguisher	existence of extinguisher	Smoke	existence of smoke
First Aid	existence of first aid	CO	detection of CO
Place	the coordinates of room	CO_2	detection of CO_2
Capacity	capacity of the room	Flood	existence of smoke
Ventilation	existence of smoke	Crowd	number of people
Crowd	number of people		

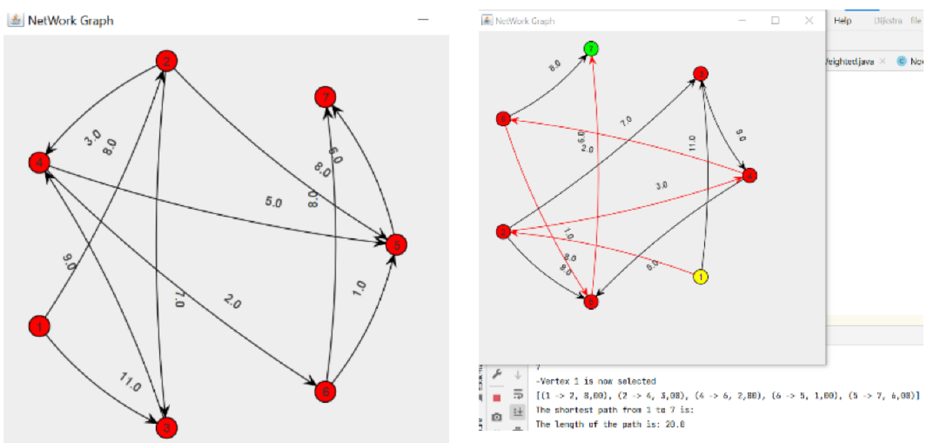

Figure 3: Representation of graph for a building and selected node 1

(Figure 3). It has been examined as mentioned above that node 7 corresponds to the emergency exit.

In the second case, we added some nodes that are impossible to traverse, simulating rooms that passing through them will put people in immediate danger (e.g. active fire, gas leaking or some other hazard). Assuming that nodes 4 and 5 are in an emergency state, then from the original form of the graph (4 4) the incoming edges to the specific nodes are removed. Finally, the graph has the form shown in Figure 4.

In the third scenario a graph with more nodes, 15 nodes and 18 edges, is presented (Figure 5). There are an additional 3 emergency exits – shelters, while the rooms in which extreme phenomena prevail are more than 2 (Figure 5). In this way, a more complex application of the problem is presented.

4 CONCLUSIONS & FUTURE WORK

This work aimed to implement a path finding algorithm at the safe evacuation of a building in emergency conditions. For this purpose, the present work was divided into two parts, one concerning the visualization of the building as a graph with nodes and edges and a second one related to finding and visualizing the shortest path for each node separately. Although our work is preliminary and mostly serve as a proof of concept, experiments shown that representing a building as graph and finding the shortest path is a feasible solution for an emergency evacuation plan. Our proposal addresses the problem of non-passable rooms and considers different traversal speeds based on the category of the person (e.g. elder, disabilities).

Open issues that could be explored by a future extension of the present work are using more complex features for each node and edge as well as using these to find the optimal usage. Furthermore, graph visualization and shortest path finding could be performed using some API for a more complete representation of the problem

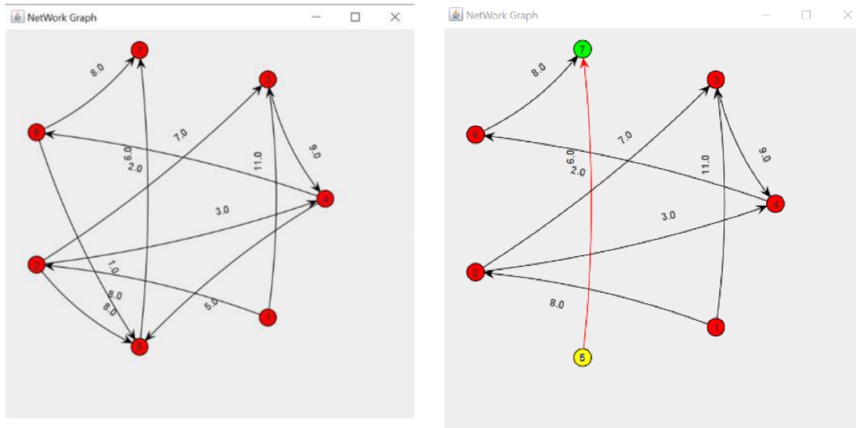

Figure 4: Building with dangerous rooms and Final representation

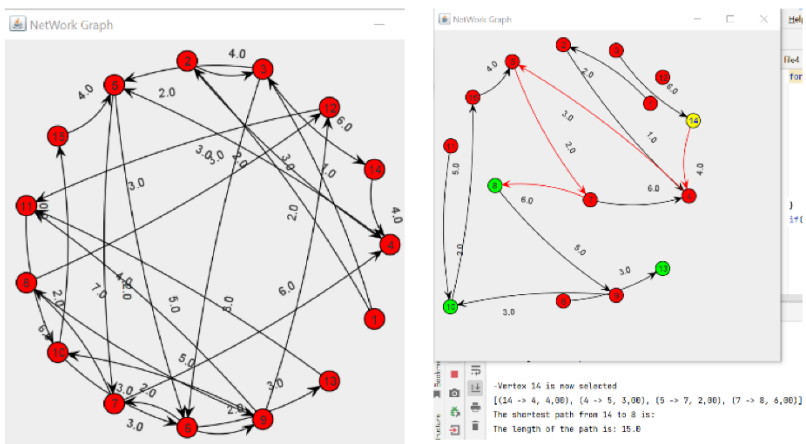

Figure 5: Graph with multiple rooms and exits/shelters and Shortest path for node 14

solution. That is, to carry out in the same environment both the design of the building as a graph and the application of the algorithm and its display towards the exit.

A robotic agent can be used which uses a heuristic algorithm to find its path. As a result, the points where obstacles are located are unreachable vertices. Path finding will output a series of states - vertices that the robot must go through in order to reach its goal. This path will adapt to new data as the agent discovers new information using its sensors as it explores its environment.

Regarding graph generation, it is worth exploring automated graph generation using the architectural floor plan. This is an extension we will work on in the near future, as it will make our application easier to adapt and implement. Convolutional neural networks could be employed, as they have an exceptional performance when using image data.

Finally, we currently experiment only with the Dijkstra Algorithm. In the future it is worth working with more complex algorithms. Since the complexity of the graph can increase exponentially, depending on the structure of the building, and considering that creating an evacuation plan is a time critical function, it is mandatory to evaluate several algorithms based on speed and scalability. This will ensure the robustness of the system in real life conditions.

ACKNOWLEDGMENTS

This research was carried out as part of the project «Beacon: An Intelligent Dynamic Signage System for Emergency Situations»(Project code: KMP6-0094837) under the framework of the Action «Investment Plans of Innovation»of the Operational Program «Central Macedonia 2014 2020», that is co-funded by the European Regional Development Fund and Greece.

REFERENCES

[1] H. Bayat, M. R. Delavar, W. Barghi, S.A. EslamiNezhad, P. Hanachi, and S. Zlatanova 2020, Modeling of emergency evacuation in building fire,321-327
[2] Yaping He $\kappa\alpha\iota$ Mahen Mahendran 2009, Lifts for Emergency Evacuation in Apartment Buildings
[3] Andrew See, Laurent Michel Bo Xiong, Peter B. Luh Shi Chung Chang 2006, Transportation and HVAC Systems for Building Emergency Evacuation,363-368
[4] Hongga Li and Xiaoxia Huang 2021Fire Emergency Evacuation for Large-Scale Public Buildings In 3d Gis,1-6
[5] Zhuoqun Zhang, Chen Wang, Yinzhe Wang 2020, Design of Emergency Evacuation Scheme for Louvre,1-10

[6] Chang Liua, Zhan-li Maoa,b, Zhi-min Fua 2016, Emergency evacuation model and algorithm in the building with severalexits Procedia Engineering 135 (2016) 12 – 18
[7] Hanan Daher, Samaneh HoseinDoost, Bahman Zamani, Afsaneh Fatemi 2020, A Novel Approach for Developing Emergency Evacuation Plans,1-7
[8] Paul Robinette $\kappa\alpha\iota$ Ayanna Howard 2012, Trust in emergency evacuation robots,1-7
[9] Konstantin Tkachuk, Xiao Song $\kappa\alpha\iota$ Irina Maltseva, Application of artificial neural networks for agent-based simulation of emergency evacuation from buildings for various purpose,1-7
[10] G.P. Cimellaro, M. Domaneschi, M. De Iuliis, V. Villa, C. Caldera, A. Cardoni 2019, FIRE EMERGENCY EVACUATION IN A SCHOOL BUILDING THROUGH VR,1-10
[11] Shuang Li, Xiaohui Yu, Yanjuan Zhang, Changhai Zhai 2018, A numerical simulation strategy on occupant evacuation behaviors and casualty prediction in a building during earthquakes,1238-1250
[12] Coşkun Şahin , Jon Rokne , Reda Alhajj 2019,Human behavior modeling for simulating evacuation of buildings during emergencies,1-13