Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: 0005-1144 (Print) 1848-3380 (Online) Journal homepage: https://www.tandfonline.com/loi/taut20

A multi-objective route planning model based on genetic algorithm for cuboid surfaces

Gonca Ozmen Koca, Sengul Dogan & Hicran Yilmaz

To cite this article: Gonca Ozmen Koca, Sengul Dogan & Hicran Yilmaz (2018) A multi-objective route planning model based on genetic algorithm for cuboid surfaces, Automatika, 59:1, 120-130, DOI: 10.1080/00051144.2018.1498205

To link to this article: https://doi.org/10.1080/00051144.2018.1498205

Published online: 06 Aug 2018.

AUTOMATIKA
2018, VOL. 59, NO. 1, 120–130
https://doi.org/10.1080/00051144.2018.1498205

REGULAR PAPER

OPEN ACCESS Check for updates

A multi-objective route planning model based on genetic algorithm for cuboid surfaces

Gonca Ozmen Koca[a], Sengul Dogan[b] and Hicran Yilmaz[a]

[a]Department of Mechatronic Engineering, University of Firat, Elazig, Turkey; [b]Digital Forensics Engineering, Faculty of Technology, University of Firat, Elazig, Turkey

ABSTRACT

During a natural disaster, risk management for the evacuation of people in high-rise buildings is very important for saving lives. In the case of fire, all parameters such as detection, lighting, warning systems, etc. for safety must be used interactively. Determination of evacuation conditions and different ways out are important parameters during the fire. In this study, a system is proposed for evacuating people from building with the shortest/safest route, taking into account certain factors to evaluate the current situation of the fire. Travelling Salesman Problem (TSP) may be adapted to this real-life problem to protect people in the shortest time finding optimum route. In this study, the system based on Genetic Algorithm is performed using the online information about smoke, heat and safety level, the location of fire and the potential congestion of people in order to evacuate people from the building with safety route. The system contains two- and three-dimensional surface applications to ensure evacuation with optimum distance inside/outside of the building. Results are evaluated considering the evacuation distance. Compared to other methods in the literature, the solution to this problem is improved by adding the evacuation process for the elevator and inside of the building.

ARTICLE HISTORY
Received 15 December 2016
Accepted 4 June 2018

KEYWORDS
Route planning; optimisation; genetic algorithm; travelling salesman problem

1. Introduction

A fire disaster happens at the unexpected moment and it occurs more often compared to the other disaster. Since the fire prevention is not always possible, security precautions have vital importance [1]. In various countries around the world, projects related to the fire safety are carried out with the “fire” code as well as architectural, mechanical and electrical codes. So far, fire applications have been limited by showing fire cabinets, sprinkler systems and water tanks for fire security. In order to ensure fire safety, the clustering of many disciplines is not only not sufficient but also fire-related parts of the project should be separated such as emergency lighting, detection, and warning systems. When separate fire safety issues combined, there will also arise interactions. For example, in the determination of escape routes, the lack of sprinkler in the building is an important parameter [1,2]. Human behaviour is also a very important factor concerning the fire safety. When an unexpected event is exhibited, human behaviour cannot be predicted and is also very difficult to control. During the fire, the main factor to ensure people to reach a safe area is time and evacuation process. At the same time, the number of people in the building, distribution of people by floors, their mobility and awareness of the behaviour are effective for determining the accurate evacuation time [1,3,4]. Nowadays, elevators in high-rise buildings have also become a necessity [5]. With the developing elevator technology, emergency rescue systems have been improved to provide transporting passengers automatically to the nearest floor and lifting the elevator safely [6,7]. In buildings such as hospitals, in the case of a power failure, elevators are operated by generators [8]. A recovery system placed in the elevator must be designed to be used in emergency situations and this system will require a minimum energy to reach the nearest floor and could open the door [6].

Optimisation techniques are used in many fields in the literature for the reduction of human workload, saving time, reduction of business costs, and efforts to reach the best. Although more realistic results can be obtained with the inclusion of the third dimension to real-life problems, most of the studies consider two-dimensional surfaces for optimisation of routes. Ugur and Aydin [9] present a simulation and analysis software to solve Travelling Salesman Problem applying Ant Colony Optimisation (ACO) algorithms. They test algorithms on benchmark problems. Their results are also supported with visualisation of algorithms. Costa and Baldo [10] present a method based on the genetic algorithm for the generation of road maps from trajectories collected with a smartphone. Their results provide high-quality maps which are same as reference

CONTACT Gonca Ozmen Koca gonca.ozmen@gmail.com

maps with less than 2 m of difference on average. A method based on an adaptive genetic algorithm for robot motion planning is investigated by Karami and Hasanzadeh [11]. They obtain better performance than the other methods in the literature with regard to the quality of the solution and finding an optimum path. Mahi et al. [12] develop a new hybrid method for TSP. The proposed method uses Particle Swarm Optimisation (PSO), ACO and 3-opt algorithms. These algorithms aim to improve the performance of TSP. Murray and Chu [13] introduce a new variant of the traditional TSP to determine optimal customer assignment for unmanned aerial vehicles (UAV). This system ensures faster delivery and less environmental effect. The dynamic TSP is adapted to fish aggregating devices based on the genetic algorithm by Groba et al. [14]. The main purpose of the study is to develop a simple prediction method for the dynamic route optimisation problem. Cui and Dong [15] develop a new idea for crossover and mutation operators to improve the performance of the genetic algorithm. They investigate the route planning of detecting robots with simulation studies and obtain better performance than classical approach finding the optimal path. Agarwal et al. [16] tried to obtain a minimum-length flight path for the unmanned reconnaissance aerial vehicle using a new TSP model. They present the effectiveness of the model with comparative results of real robots. An Ant Colony System algorithm is introduced for an Unmanned Reconnaissance Aerial Vehicle (URAV) by Chen et al [17]. The system is presented to expose applicable optimal route in a sparse graph within the tractable time for the URAV. Simulation results show the robustness of the proposed system with minimal risk. Ugur [18] who take into account a third dimension in the work presents an approach to collect people on the cuboid surface. This study is different from the others with the application surface which is applied for three dimension.

In this study, the main purpose is to find the best escape route and to evacuate people as quickly as possible from the high-rise building in the case of a disaster such as a fire. The cuboid optimisation method is applied for route planning inside and outside of the building in the event of a fire. The escape route, exits of the building and arrival to meeting places are important factors to determine optimal routes for this problem. The main parameters are considered as smoke and heat level, safety level, the location of fire (obstacle) and the potential congestion of people in the proposed system. TSP is applied to a cuboid-like structure surface reminded structure of the building by using a genetic algorithm optimisation method in solving the problem. Genetic algorithm provides an effective optimisation not with a single point but with a set of points in a large area. Genetic algorithm only needs the objective functions without the use of derivatives and other helpful information. Thus, it may work with many different types of continuous and/or discontinuous, linear and/or non-linear parameters or functions. Despite the increase in computation time due to the use of conventional methods in large solution space, reasonable results can be obtained by genetic algorithm in a shorter period of time.

In this paper, optimal route planning during a fire is proposed. In Section 2, a genetic algorithm is described. In Section 3, the proposed system is presented. Results of proposed systems are given in Section 4. Finally, the conclusion of this paper is present in Section 5.

2. Genetic algorithm

Genetic algorithm (GA) encoding input parameters biologically which are effective in the solution is a search and optimisation method. This method does not work over the entire solution set; it works in a specific section randomly selected. Thus, the solution is reached in a shorter time [19].

In the first step of GA, parameters are coded and chromosomes generating the random population are created. Then, the fitness value of the initial populations is calculated and two parents with the best fitness value of these populations are selected. New generations are obtained by applying crossover and mutation processes. Fitness values of new generations are calculated and added to the population. This process is continued until ensuring a certain amount of improvement [19,20].

Generally, scanning wide solution spaces with the traditional search methods increases the computation time. The successful results can be obtained by using a genetic algorithm in cases where the solution spaces are wide, discontinues and complex [20,21].

2.1. Travelling salesman problem

TSP is an optimisation problem used widely in the literature. This problem expresses that the seller visits all cities with the cheapest cost in the shortest time and returns to the starting city [22]. So, this problem aims to find the best route on condition that only one pass from each of the specific points [14]. As the number of points increases, the problem becomes more complicated. In this study, in parallel with the TSP, people evacuation is aimed as quickly and safely in the emergency of a situation such as a fire in a high-rise building. In order to achieve this aim, the next three steps are followed.

Step 1: The creation of a route planning if there are any people in the elevator.

Step 2: The creation of a route planning for people in any position inside the building.

Step 3: In the shortest time, the collection of people reaching out of the building surface.

TSP can be applied individually for each of mentioned three steps. TSP is applied to be achieve the

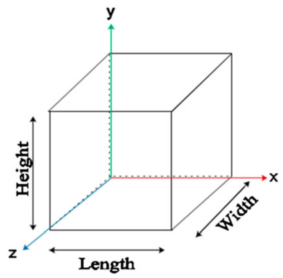


Figure 1. Three-dimensional cuboid.

most appropriate floor by the elevator and to reach exits inside the building. Cuboid TSP is applied to the collection of people from out of the building surface. In the TSP and cuboid TSP, the most appropriate route is determined as considering points on the plane and on a cuboid shape, respectively [23].

2.2. Cuboid TSP

The cuboid given in Figure 1 is a geometric shape bounded by six parallel faces. Cuboid shape is preferred because of flatness, balance and minimum space for optimisation.

In this study, the solution of TSP is obtained by using a genetic algorithm method. Points are on the cuboid surface instead of a flat plane for cuboid TSP. The distance between points for Cuboid TSP is calculated by using Euclidean distances based on x, y, z coordinates [*XX*]. Cuboid TSP can be used to solve problems such as route planning, collection of parts, placement of the component, and writing to object-shaped box [18].

2.2.1. Implementation of the cuboid TSP

All points are located on the lateral surfaces of the cuboid. To perform the optimum route planning, starting from the initial point and visiting all points on lateral surfaces and returning to the initial point is required. Considered solution for the TSP on cuboid includes:

- Find the shortest distance between pairs of points on the cuboid surface
- Solve the TSP by using GA method.

Primarily, six surfaces of the cuboid are represented as shown in Figure 2 to calculate distances between points. One corner of the cuboid is placed in the coordinate system.

When calculating distances between points, there are three possibilities to find the location of the pair of points on the cuboid surface [18]:

- Both two points on the same surface
- Points on opposite surfaces
- Points on adjacent surfaces

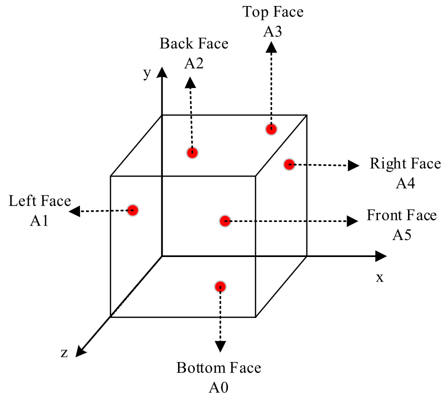


Figure 2. The representation of cuboid surfaces.

2.2.2. Distance calculation on cuboid surface

Coordinates of points in three dimension are represented as x_i, y_i, z_i and x_j, y_j, z_j where i and j are the pair of points. The distance between the pair of the points is called distance$_{i,j}$, and the calculation would be as follows:

I. If i and j are represented on the same surface, the shortest distance between two points is expressed by a straight line. Thus, distances can be detected directly with Euclidean expression.

$$distance_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2} \quad (1)$$

II. If i and j are on the opposite surface, there are four alternative ways to calculate the distance. There are three opposite surfaces on a cuboid and there are four ways of alternative calculation for each opposing surface. Distances are calculated by using these four ways and the shortest distance is selected. For example, the following four alternative ways to calculate the distance between two points on opposite top and bottom surfaces of the cuboid can be expressed as:

Case 2.1: Front surface–top surface–back surface

Case 2.2: Front surface–bottom surface–back surface

Case 2.3: Front surface–right surface–back surface

Case 2.4: Front surface–left surface–back surface

Right triangles are obtained from these edge lengths as shown in Figure 3.

Equations used in distance calculation for the front surface and the back surface of opposing surfaces are given with Equations (2–5). These equations are extracted by applying the Euclidean relation to points of the open state of the cuboid. Distances are calculated by using Equations (2–5) for Cases 2.1–2.4, respectively.

$$\begin{aligned} mx &= |x_j - x_i| \\ my &= |height - y_j| + width + |height - y_i| \\ distance_{ij} &= \sqrt{mx^2 + my^2} \end{aligned} \quad (2)$$

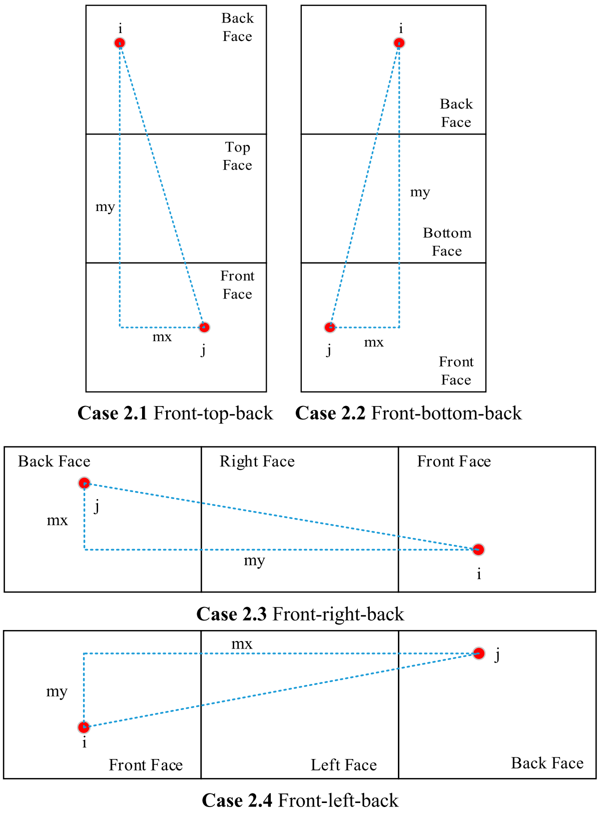


Figure 3. Distances between two points on opposite top and bottom surfaces of the cuboid.

$$\begin{aligned} mx &= |y_j - y_i| \\ my &= |length - x_j| + width + |length - x_i| \\ distance_{ij} &= \sqrt{mx^2 + my^2} \end{aligned} \tag{3}$$

$$\begin{aligned} mx &= |x_j - x_i| \\ my &= y_j + width + y_i \\ distance_{ij} &= \sqrt{mx^2 + my^2} \end{aligned} \tag{4}$$

$$\begin{aligned} mx &= |y_j - y_i| \\ my &= x_j + width + x_i \\ distance_{ij} &= \sqrt{mx^2 + my^2} \end{aligned} \tag{5}$$

III. If i and j are on adjacent surfaces, there are three alternative ways to calculate distances. The shortest distance is selected among them. For example, in order to calculate the distance between two points on the neighbour's front and top surfaces of the cuboid, the calculation of three alternative ways is as follows:

Case 3.1: Along a common edge of the surface containing points (bottom surface–right surface)

Case 3.2: Through the first adjoining face of the surface containing points (bottom surface–front surface–right surface)

Case 3.3: Through the first adjoining face of the surface containing points (bottom surface–back surface–right surface)

The three alternative ways are summarised in Figure 4 for the bottom and the right surface. Equations used for distance calculation in the bottom and the right surface of adjacent surfaces are given with Equations (6–8) for the Cases 3.1–3.3, respectively.

$$\begin{aligned} mx &= |z_j - z_i| \\ my &= y_j + |length - x_i| \\ distance_{ij} &= \sqrt{mx^2 + my^2} \end{aligned} \tag{6}$$

$$\begin{aligned} mx &= |width - z_j| + |length - x_i| \\ my &= y_i + |width - z_j| \\ distance_{ij} &= \sqrt{mx^2 + my^2} \end{aligned} \tag{7}$$

$$\begin{aligned} mx &= |length - x_i| + z_j \\ my &= y_j + z_i \\ distance_{ij} &= \sqrt{mx^2 + my^2} \end{aligned} \tag{8}$$

The shortest distances can be found using the genetic algorithm method in cuboid surfaces for three cases mentioned above [18].

2.2.3. *The cuboid approach to TSP based on the genetic algorithm*

After calculating the distance between points on the cuboid, the distance matrix is created. Minimum distance should be obtained by visiting all points of the cuboid. In this study, GA is applied to obtain minimum distance. The cuboid TSP process is given in Figure 5.

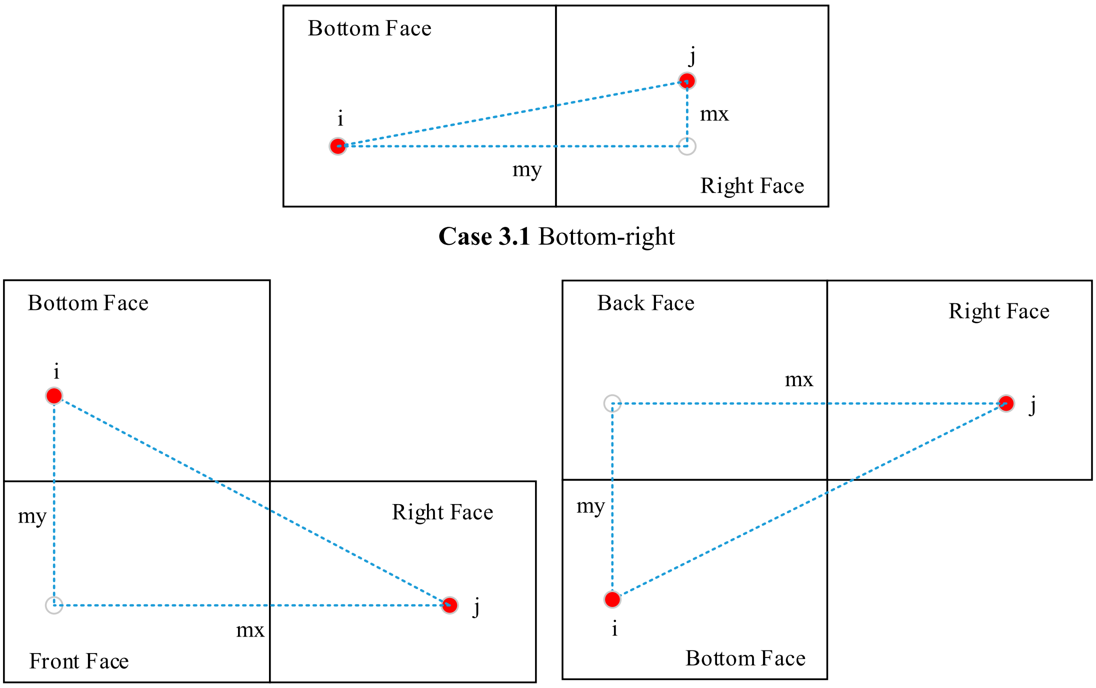
Problems would have gained a third dimension applying TSP to the surface of the cuboid. Thus, realistic results are obtained for problems in real life.

3. The proposed system

The aim of this study is to determine the optimum route for the evacuation of people in the high-rise building. The location of people is chosen as the input value of the purposed system. In this system, two different routes (G1, G2) are defined in order to canalise people to collection areas and a different route (G3) is also defined to collect people waiting for rescue in the collection area. The main purpose of the system is to find the optimum routes for all cases. Escape routes to collection areas (G1, G2) and collection route (G3) considering planes and used methods are given in Table 1.

The proposed system as applied to a high-rise building is shown in Figure 6. In the proposed system, based on Figure 6, a structure is formed to reach evacuation points from shortest and safe route by evaluating G1, G2 and G3 cases.

Process steps of the proposed system are given below and definitions of values are represented in Table 2. Guidance signals to reach exit points are expressed by



Case 3.1 Bottom-right

Case 3.2 Bottom-front-right

Case 3.3 Back-bottom-right

Figure 4. Distances between two points on opposite top and bottom surfaces of the cuboid.

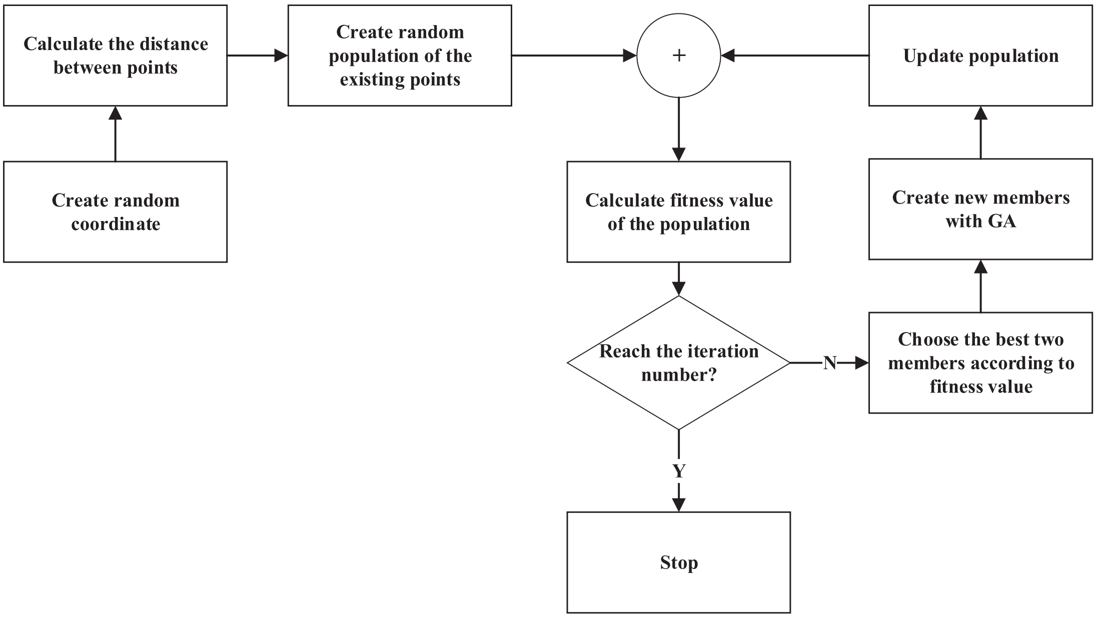


Figure 5. Process steps of the cuboid TSP.

Table 1. Routes for all cases.

Movements	Planes	Routes	Methods
Escape routes to collection areas			
Movement of the elevator	*x*–*y* plane	G1	TSP
Movement on the floor and/or between floors	*x*–*y*–*z* plane	G2	TSP-cuboid TSP
Collection route			
Collection of building surfaces	*x*–*y*–*z* plane	G3	Cuboid TSP

three different colours such as green, yellow and red. As shown in Table 3, green, yellow and red indicate safe, intermediate level safe and unsafe routes, respectively.

Definition 3.1: Positioning of the elevator in nearest floor according to heat and smoke level is the main purpose of G1 route to save people inside the elevator.

Step G1_1: *Check the elevator if there is anyone*
If no {goto step G1_3}
Step G1_2: $Level_{i+1} > Level_i$ *?*
If yes {goto $Floor_i$ *using TSP}*
If no {goto $Floor_{i+1}$ *using TSP}*
Step G1_3: *Goto step G2_1*

Definition 3.2: The main purpose of G2 route is to select points which have less heat and smoke level and

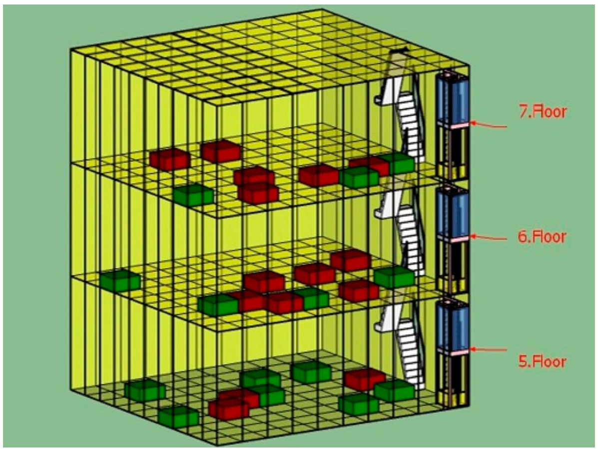


Figure 6. The representation of the inside of the building.

Table 2. Values used in process steps.

Values	Definitions	Values	Definitions
i	Current floor	δ	The safety limit value determined by the proposed system
n	The number of floor	tpn	The number of transition points
$Floor_i$	ith floor ($i = 1,2, \ldots n$)	lpc	The level of potential congestion
s	Variable indicating change in the number of floor	Γpc	Threshold value of potential congestion
$Level_i$	Smoke and heat level in the ith floor	tpl	The location of transition points
Γ_1	Threshold value for the smoke and heat level	dpn	The number of disabled points
Γ_2	Threshold value for the smoke and heat level in the stair	spn	The number of disabled points in the stair

Table 3. Exits with colours.

Exits with colours	Definitions of colours
Green exit	Safe route
Yellow exit	Intermediate level safe route
Red exit	Unsafe route

Table 4. Parameters of genetic algorithms.

Parameters of genetic algorithm	Default values
Generation size	100
Population size	250
Crossover rate	0.80
Mutation rate	0.05

direct people to the safest and shortest way. If there are two routes with the same level of security, the route may be determined according to the level of potential congestion because of congestion delays which can decrease the level of safety. Genetic algorithm process is applied to TSP to reach the outer surface of the building according to fire points. Parameters of genetic algorithm used in the proposed method are given in Table 4.

One is directed to another floor for his/her safety in case of a high number of disabled points in the current floor. In case of going another floor, the safety of stairs is considered.

Step G2_1: *Determine the number (tpn), location (tpl) of transition points according to the amount of heat and smoke level and the level of potential congestion (lpc) for various determined regions.*

Step G2_2: *Put tpl to the matrix of dis_TSP*

$$dis\_TSP = \begin{bmatrix} x_1 & z_1 \\ x_2 & z_2 \\ \vdots & \vdots \\ x_{tpn} & z_{tpn} \end{bmatrix}_{(tpnx2)}$$

Step G2_3: *Is there any obstacle in determined points?*

$$\text{If yes} \begin{Bmatrix} \text{remove disabled points from the matrix} \\ \text{of } dis\_TSP \\ \{size(dis\_TSP) = (tpn - dpn)x2\} \end{Bmatrix}$$

Step G2_4: $\Gamma_1 \geq \delta \left\{ \boldsymbol{\Gamma}_1 = \frac{dpn}{tpn} \right\}?$

$$\text{If no} \begin{Bmatrix} \text{find the shortest route using TSP} \\ \text{with green exit} \\ \text{select the path taking into account of } lpc \end{Bmatrix}$$

Step G2_5: *May one reach a collection point?*

$$\text{If yes} \begin{Bmatrix} \text{find the shortest route using TSP} \\ \text{with yellow exit} \\ \text{select the path taking into account of } lpc \end{Bmatrix}$$

Step G2_6: *(May one reach stairs?) & (spn* $< \Gamma_2$ *?)*

If no {goto window using with TSP}

Step G2_7: *May one reach exit from stairs?*

If yes {goto exit using cuboid TSP}

Step G2_8: $Level_{i+s} \geq Level_{i-s}$*?*

If yes {goto $Floor_{i-s}$ *using cuboid TSP}*

If no {goto $Floor_{i+s}$ *using cuboid TSP}*

Step G2_9: *Goto Step G2_4*

In addition, if one cannot find the safety route on the floor or between floors after trying all possibilities, she/he goes to the nearest window and a new evacuation point occurs.

Definition 3.3: Collected people reaching the outer surface of the building with the shortest route is the main purpose of G3 route. The cuboid TSP based on the genetic algorithm is implemented in this step.

Step G3_1: *Determine the number (epn) and location (epl) of evacuation points.*

Step G3_2: *Put epl to the matrix of dis_CTSP*

$$dis\_CTSP = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_{epn} & y_{epn} & z_{epn} \end{bmatrix}_{(epnx3)}$$

Step G3_3: *Find the shortest route using cuboid TSP*

Step G3_4: *Pursue the route*

4. Experimental results

The aim of this study is to ensure delivering people located in the building to the collection area in the

Table 6. Selected disable points inside building for cuboid surface.

Number of cities	Randomly selected fire points
10	8 disable point (10, 9, 7, 6, 5, 4, 3, 2)
20	13 disable point (19, 18, 16, 15, 13, 12, 11, 10, 8, 6, 4, 3, 1)
30	18 disable point (23, 22, 21, 20, 19, 18, 17, 15, 14, 12, 11, 8, 7, 6, 5, 3, 2, 1)
40	23 disable point (33, 31, 29, 28, 27, 25, 24, 23, 20, 19, 18, 17, 15, 14, 12, 11, 10, 9, 7, 5, 4, 3, 1)
50	28 disable point (50, 49, 48, 46, 44, 42, 39, 37, 36, 33, 32, 30, 28, 26, 25, 23, 20, 19, 17, 15, 14, 12, 11, 10, 7, 4, 3, 1)

Table 5. Selected disable points inside building for flat surface.

Number of cities	Randomly selected fire points
10	4 disable point (2, 4, 5, 7)
20	6 disable point (1, 4, 8, 10, 13, 15)
30	8 disable point (2, 5, 8, 11, 14, 17, 20, 23)
40	10 disable point (4, 9, 14, 19, 20, 23, 25, 27, 29, 31)
50	12 disable point (1, 3, 5, 10, 15, 20, 25, 30, 32, 33, 36, 39)

Table 7. Obtained optimal distances for flat surface selected at different sizes ($S\_1 = 1000 \times 1000$, $S\_2 = 750 \times 500$, $S\_3 = 250 \times 250$).

		Genetic algorithm generation size				
Number of cities	Size	20	40	60	80	100
10	S_1	1649.3	1087.8	828.3	642.0	597.6
	S_2	759.3	503.1	378.9	284.6	232.1
	S_3	448.9	259.0	187.2	135.4	122.7
20	S_1	4758.8	4573.6	4283.8	3949.5	3801.8
	S_2	2941.1	2769.2	2494.0	2434.2	2405.1
	S_3	1049.4	961.4	897.7	881.3	859.8
30	S_1	8003.5	7653.9	7344.8	7046.5	6998.3
	S_2	5018.4	4592.9	4583.5	4315.6	4249.6
	S_3	2288.3	2222.3	2189.0	2141.1	2110.4
40	S_1	11,151	10,802	10,331	10,106	9839.5
	S_2	6529.8	6194.3	6080.3	5906.8	5897.0
	S_3	6737.4	6304.9	6032.0	5948.4	5831.6
50	S_1	14,244	14,157	13,723	13,585	13,319
	S_2	8001.8	7769.4	7586.2	7440.5	7165.5
	S_3	8446.7	8023.0	7779.8	7511.8	7511.4

shortest time and from the safest way. This problem is adapted to TSP and GA for flat and cuboid surfaces to optimise the route by using the safest way. The method is implemented using MATLAB2014a on a PC with Core i5 CPU and 8GB of RAM.

In the proposed system for indoor surface of the building, randomly selected disable points for the selected number of (10–50) cities are given in Tables 5 and 6 for flat and cuboid surfaces, respectively. If people cannot proceed in the current floor due to the high number of fire points, they make transitions between floors and the cuboid TSP is used to solve this problem. In the case of equal security level of floors, the human density of the floor is taken into account to select the accurate floor which has the least human density in order to prevent delay time. Thus, safety is also ensured by preventing the potential congestion of people on the floor.

Obtained optimum distances according to disabled points are given in Tables 7–10 for flat surface, potential congestion of people on the floor, between floors and outside surface of the building, respectively. At the same time, average distances between the current point and the nearest available collection point on the current

Table 8. Obtained optimal distances at different sizes in the case of the potential congestion of people on the floor ($S\_1 = 1000 \times 1000 \times 1000$, $S\_2 = 750 \times 500 \times 500$, $S\_3 = 250 \times 250 \times 250$).

		Genetic algorithm generation size				
Number of cities	Size	20	40	60	80	100
10	S_1	1077.3	979.9	901.5	891.7	863.4
	S_2	1031.3	911.3	879.1	844.8	832.8
	S_3	890.1	799.5	748.9	706.2	693.8
20	S_1	4470.3	4007.0	3652.3	3612.4	3440.6
	S_2	2988.5	2659.3	2413.7	2300.1	2284.1
	S_3	2121.0	1969.4	1830.5	1654.9	1637.4
30	S_1	8906.4	8570.3	8206.4	7928.6	7531.2
	S_2	5259.1	4955.7	4750.3	4628.0	4538.1
	S_3	3327.3	3026.2	2815.8	2733.7	2718.9
40	S_1	11,400	10,720	10,236	9896.4	9868.8
	S_2	6575.6	6245.5	5949.1	5684.4	5586.7
	S_3	4469.0	4279.9	3969.1	3920.6	3901.9
50	S_1	13,582	12,737	12,509	12,189	11,893
	S_2	7799.8	7491.6	7297.8	7069.1	6920.1
	S_3	5297.0	4985.2	4802.4	4652.3	4537.0

Table 9. Obtained optimal distances between floors selected at different sizes ($S\_1 = 1000 \times 1000 \times 1000$, $S\_2 = 750 \times 500 \times 500$, $S\_3 = 250 \times 250 \times 250$).

		Genetic algorithm generation size				
Number of cities	Size	20	40	60	80	100
10	S_1	1042.4	931.0	907.9	831.6	785.6
	S_2	974.9	901.9	866.8	840.1	804.0
	S_3	867.1	793.4	679.0	658.0	648.1
20	S_1	4226.8	3773.4	3591.9	3286.3	3045.4
	S_2	3203.5	2829.9	2680.9	2501.9	2271.5
	S_3	2101.9	1964.1	1718.2	1700.0	1636.2
30	S_1	8731.5	8356.3	7889.7	7661.0	7299.3
	S_2	5186.9	4891.0	4748.8	4549.8	4474.7
	S_3	3670.5	3418.2	3187.5	3143.3	3029.2
40	S_1	11,497	10,705	10,220	9878.8	9734.0
	S_2	6531.3	6311.7	5884.7	5777.2	5417.6
	S_3	6993.2	6652.5	6551.0	5965.7	5932.9
50	S_1	13,575	12,843	12,287	12,081	11,774
	S_2	7752.6	7531.2	7127.4	7002.4	6896.4
	S_3	7150.3	6756.2	6564.4	6478.6	6336.0

floor are illustrated in Figure 7. Average distances in the case of potential congestion of people on the floor and moving between floors are also presented in Figures 8 and 9, respectively. In Figure 10, average distances in the case of collected people reaching evacuation points are given.

Distance values are separately obtained for the randomly selected flat surface, between floors and outside surface of the building and results are given in Tables 7–9.

According to selected 30 points (number of cities), the coordinates of the obtained path are given in Table 11 and average values of the distances are also found

Table 10. Obtained optimal distances of the outside surfaces selected at different sizes of the building (S_1 = 1000 × 1000 × 1000, S_2 = 750 × 500 × 500, S_3 = 250 × 250 × 250).

		Genetic algorithm generation size				
Number of cities	Size	20	40	60	80	100
10	S_1	6470.2	5952.5	5553.1	5301.5	5159.9
	S_2	3698.7	3463.3	3321.2	3201.1	3102.6
	S_3	3439.6	3063.5	2920.2	2909.6	2799.3
20	S_1	13,384	13,020	12,374	12,357	11,850
	S_2	8050.0	7599.7	7372.3	7283.1	7147.8
	S_3	7552.1	6994.9	6748.7	6483.2	6257.8
30	S_1	21,888	20,940	20,589	19,725	19,597
	S_2	12,711	12,312	11,777	11,762	11,662
	S_3	11,481	11,050	10,627	10,274	9924.1
40	S_1	29,609	28,562	28,057	27,215	26,792
	S_2	17,287	16,792	16,065	15,972	15,267
	S_3	15,850	15,229	15,065	14,045	14,269
50	S_1	36,899	36,455	35,561	34,390	34,504
	S_2	21,432	21,021	20,512	20,257	19,961
	S_3	22,901	21,761	21,423	20,473	20,338

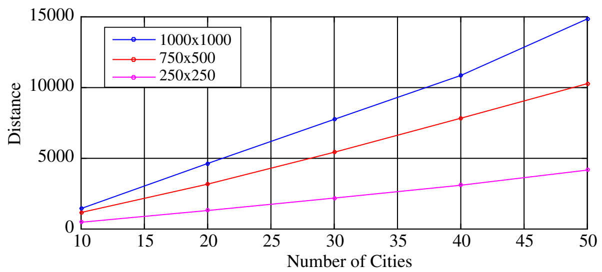


Figure 7. Average distance obtained from flat surfaces with different sizes.

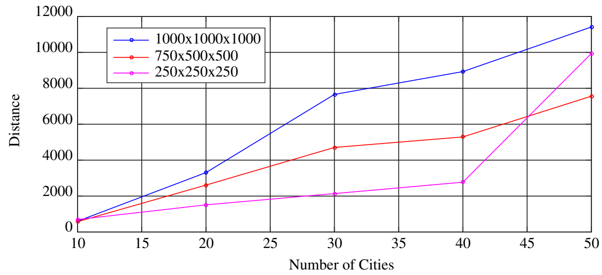


Figure 9. Average distance obtained from the inside of the building for cuboid surfaces with different sizes.

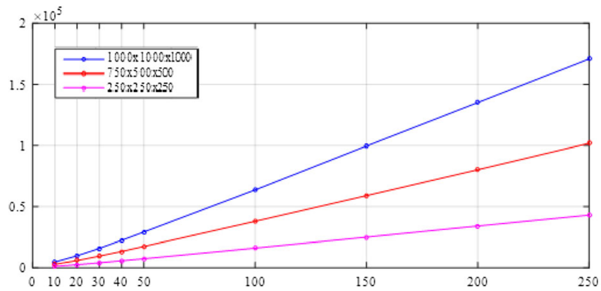


Figure 10. Average distance obtained from the outside surfaces of the building for cuboid surfaces with different sizes.

as 7409.4 for the floor size of 1000 × 1000, 7987.56 and 20,547.8 for the size of 1000 × 1000 × 1000 as the cuboid surface. By using the same number of points, average values of the distances are obtained as 4552 for the size of 750 × 500, 4770.24 and 12,044.8 for the size of 750 × 500 × 500, 2190.22, 3289.74 and 10,671.22 are also obtained for two- and three-dimensional surfaces, each with size 250. In Table 11, the first two columns represent coordinates for surface dimensions of 1000 × 1000 × 1000 from 6th to 5th and 6th to 7th floor, in the case of congestion of people on the 6th floor. One on the 6th floor might be directed to other floors (primarily the upper (7th) or the lower floor (5th), provided security level is not below the threshold of safety limit value (δ)) due to the density of people above the threshold value of potential congestion (Γpc) on the floor.

The obtained shortest distances for optimal route planning are shown in Figures 11–13. The first priority is the safety of the route while the optimal routes are investigated by using GA. The obtained optimal distances are transferred to the SketchUp 2015 program in order to illustrate optimal routes on different surfaces of the building.

The optimal route scenario for 30 randomly selected points (20 number of points for floor surface and 10 number of points for stairs) is shown in Figure 11 according to safety colour given in Table 3. Red points represent a lower security level with the high level of fire or smoke (higher than δ) and green points are safety to pass over.

In Figure 11, the route planning for reaching the exit point considering the shortest and the safest route

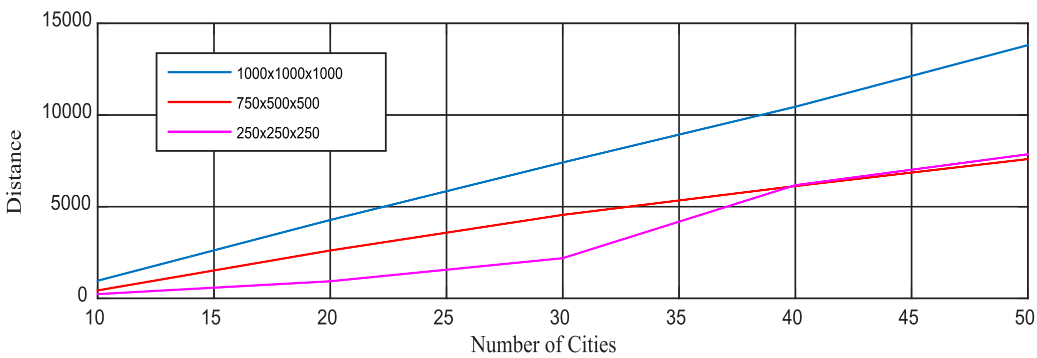


Figure 8. Obtained average distance in the case of the potential congestion of people on the floor.

Table 11. Coordinates for 1000 × 1000 (flat) and 1000 × 1000 × 1000 (cuboid).

	Surface dimensions					
	1000 × 1000 × 1000 from 6th floor to 5th floor	1000 × 1000 × 1000 from 6th floor to 7th floor	Inside of the building for the cuboid surface	The inside of the building for the flat surface	Outside of the building (cuboid)	Number of surface
1	**400–500–300**	**400–700–300**	**400–600–300**	**400–600–300**	100–200–0	3
2	500–500–600	500–700–600	500–600–600	500–600–600	100–1000–300	4
3	**300–500–800**	**300–700–800**	**300–600–800**	300–600–800	0–100–400	2
4	**1000–500–700**	**1000–700–700**	**1000–600–700**	**1000–600–700**	1000–200–500	5
5	900–500–700	900–700–700	900–600–700	900–600–700	200–100–1000	6
6	**200–500–1000**	**200–700–1000**	**200–600–1000**	200–600–1000	100–500–0	3
7	700–500–100	700–700–100	700–600–100	700–600–100	300–1000–500	4
8	**800–500–300**	**800–700–300**	**800–600–300**	**800–600–300**	0–400–600	2
9	500–500–800	500–700–800	500–600–800	500–600–800	1000–400–700	5
10	**600–500–1000**	**600–700–1000**	**600–600–1000**	**600–600–1000**	400–300–1000	6
11	**500–500–300**	**500–700–300**	**500–600–300**	500–600–300	300–400–0	3
12	**500–500–700**	**500–700–700**	**500–600–700**	500–600–700	400–1000–600	4
13	**300–500–600**	**300–700–600**	**300–600–600**	**300–600–600**	0–300–500	2
14	1000–500–500	1000–700–500	1000–600–500	1000–600–500	1000–600–400	5
15	**800–500–700**	**800–700–700**	**800–600–700**	**800–600–700**	500–100–1000	6
16	**400–500–1000**	**400–700–1000**	**400–600–1000**	400–600–1000	400–500–0	3
17	700–500–200	700–700–200	700–600–200	700–600–200	600–1000–400	4
18	**200–500–900**	**200–700–900**	**200–600–900**	200–600–900	0–600–300	2
19	**900–500–800**	**900–700–800**	**900–600–800**	900–600–800	1000–500–100	5
20	700–500–1000	700–700–1000	700–600–1000	700–600–1000	500–300–1000	6
21	500–520–0	700–680–0	600–580–0	700–680–0	500–600–0	3
22	500–540–0	700–660–0	600–560–0	700–660–0	600–1000–700	4
23	500–560–0	700–640–0	600–540–0	700–640–0	0–500–700	2
24	500–580–0	700–620–0	600–520–0	700–620–0	1000–700–300	5
25	500–570–0	700–600–0	600–500–0	700–600–0	800–200–1000	6
26	500–530–0	700–670–0	600–570–0	700–670–0	600–200–0	3
27	500–550–0	700–650–0	600–550–0	700–650–0	700–1000–500	4
28	500–600–0	700–630–0	600–530–0	700–630–0	0–800–100	2
29	500–590–0	700–610–0	600–510–0	700–610–0	1000–400–300	5
30	500–510–0	700–630–0	600–530–0	700–630–0	700–200–1000	6

Note: Bold values represent coordinates of obstacle points.

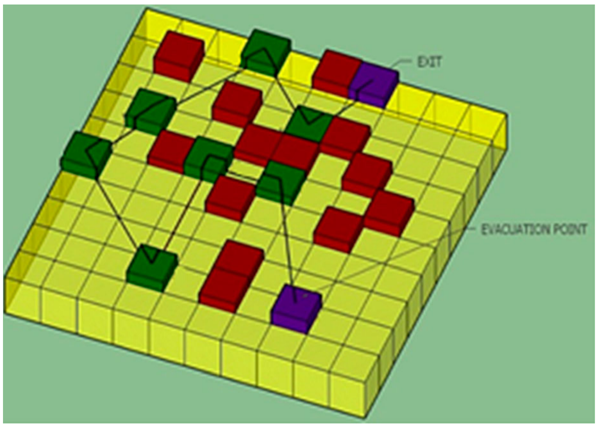


Figure 11. Side view of the optimal route planning of the inside building.

in the same floor is represented from the side view of the floor. Number 10 is the evacuation point to direct people to the planned route according to the selected disable point shown in Table 6. Figure 12 represents the optimal route planning in the case of congestion of people in the 6th floor which is detailed in Table 11. For surface dimensions of 1000 × 1000 × 1000 from 6th to 5th and 6th to 7th floor, in the case of congestion of people on the 6th floor is determined. One on the 6th floor might be directed to other floors because of the density of people above Γpc on the floor. When the case of congestion, the person is steered primarily to the upper (7th) or the lower floor (5th) provided the security level above the safety limit value (δ)).

Views from different aspects of the route scenario of collected people reaching evacuation points in the outside surface of the building are illustrated in Figure 13(a–c).

In this part of the study, randomly selected 30 number of points are used to create the scenario. Points are selected from a different side of the building as seen from the figure.

5. Conclusion

In the event of a fire, the most important issue is the evacuation of people. Therefore, the most important point in the building design stages of the project is the structure of evacuation ways and exits. When designing a building, the place of evacuation ways, fire exits, elevators and air conditioning have vital importance to remove smoke from the building as soon as possible. How far it is from outside, which safest ways can be used for leaving the building, where exit points are and if there is any delay due to human density to ensure the escape from the building from the shortest and safe way are very important factors.

The main purpose of this study is to ensure finding the safest route in the building and evacuating

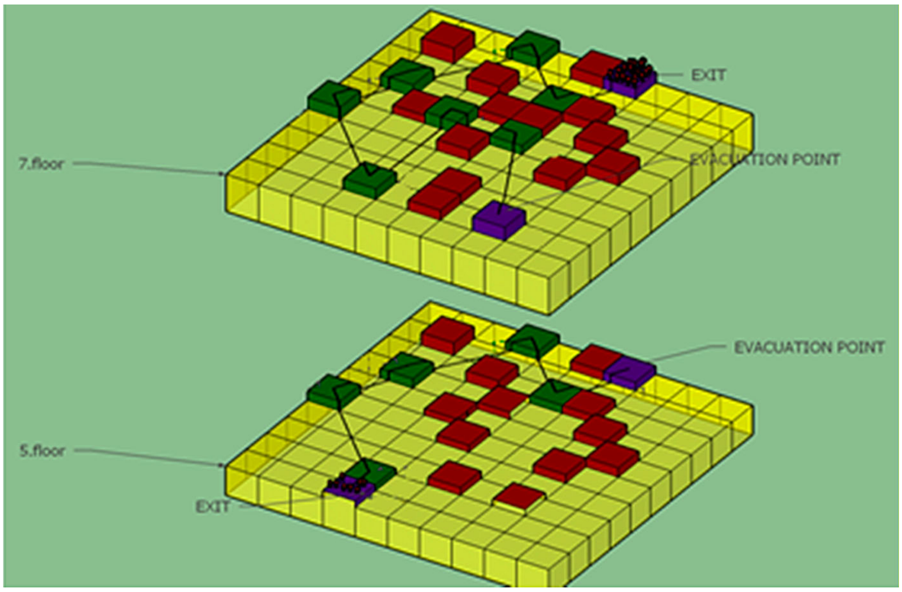


Figure 12. Side view of the optimal route planning of the inside building in the case of potential congestion of people.

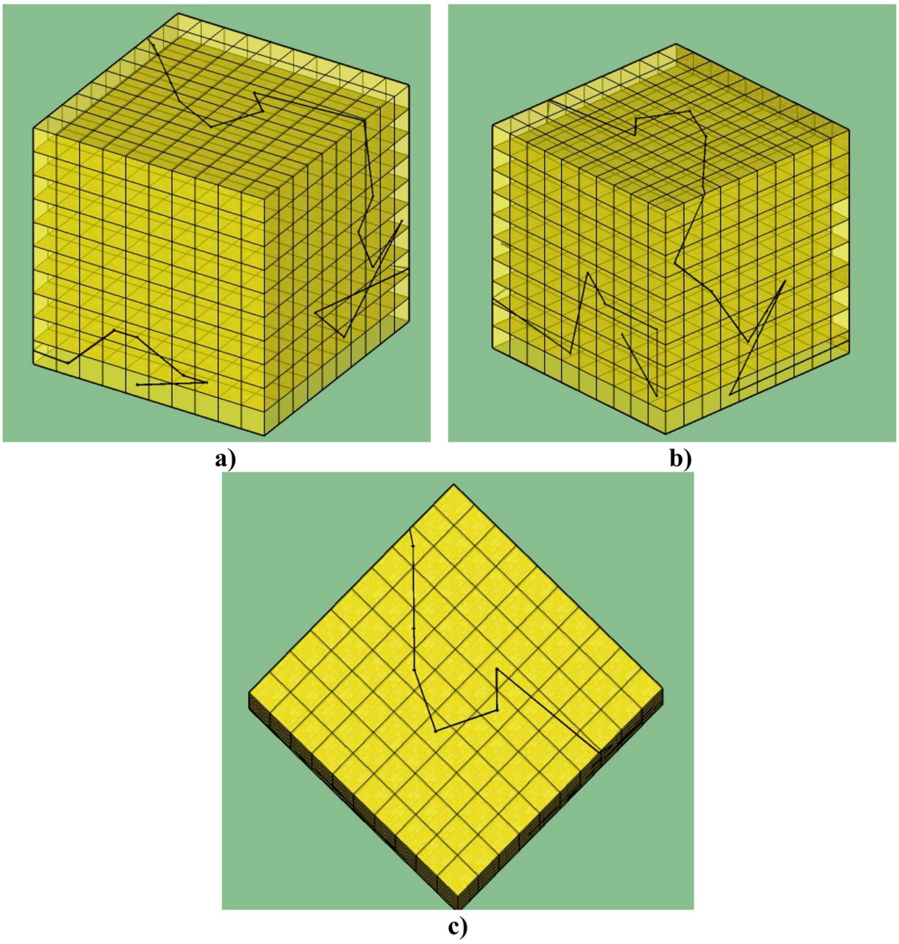


Figure 13. (a) Front–right–top view, (b) Back–left–top view, (c) Top view of the outside surface of the building.

the building in the minimum time in the case of a fire disaster. Genetic algorithm which is one of the most widely used optimisation methods is applied to the flat and cuboid surfaces in the planning of escape routes for evacuation of the building and the surface, respectively. For the realistic solution of this problem, the application of three-dimensional TSP is implemented for movement between floors and collection from building surfaces. At the same time, TSP is also used for a flat surface to evacuate people from elevators and to direct people to the collection area in the floor. In the future work, optimisation techniques will be tested to improve time concept and collection of people according to priority status who reached evacuation points will be also provided for an optimum distance.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

[1] Brown C, Milke M, Seville E. Disaster waste management: a review article. Waste Manage. 2011;31(6): 1085–1098.

[2] Onan K, Ulengin F, Sennaroglu B. An evolutionary multi-objective optimization approach to disaster waste management: a case study of Istanbul, Turkey. Expert Syst Appl. 2015;42(22):8850–8857.

[3] Ozkaya A, Sarikaya S. The guide on the protection of buildings from fire, chamber of architects. Ankara: Technical Bulletin; 2007.

[4] Basdemir H, Demirel F. A literature review of passive fire safety precautions in buildings. J Polytech. 2010;13(2):101–109.

[5] Bolat B. The simulation and optimization of lift control systems with genetic algorithms. Sigma. 2006;2:14–20.

[6] Imrak CE, Fetvaci MC. Asansörlerde acil kurtarma sistemleri ve Güç kaynaklarinin belirlenmesi. Mühendis ve Makina Dergisi. 2003;44:19–25.

[7] Schwedt H, Benczek G. Mechatronic for elevator installers and drive technicians. 2001. (Lift Report), 6, p. 94–96.

[8] Strakosch GR. Vertical transportation: elevators and escalators. 2nd ed. New York (NY): John Wiley & Sons; 1982.

[9] Ugur A, Aydin D. An interactive simulation and analysis software for solving TSP using Ant colony optimization algorithms. Adv Eng Softw. 2009;40:341–349.

[10] Costa HRG, Baldo F. Generation of road maps from trajectories collected with smartphone – a method based on genetic algorithm. Appl Soft Comput. 2015;37: 799–808.

[11] Karami AH, Hasanzadeh M. An adaptive genetic algorithm for robot motion planning in 2D complex environments. Comput Electr Eng. 2015;43: 317–329.

[12] Mahia M, Baykan ÖK, Kodaz H. A new hybrid method based on particle swarm optimization, Ant colony optimization and 3-Opt algorithms for traveling salesman problem. Appl Soft Comput. 2015;30:484–490.

[13] Murray CC, Chu AG. The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. Transportation Research Part C. 2015;54: 86–109.

[14] Groba C, Sartal A, Vázquez XH. Solving the dynamic traveling salesman problem using a genetic algorithm with trajectory prediction: An application to fish aggregating devices. Comput Oper Res. 2015;56:22–32.

[15] Cui SG, Dong JI. Detecting robots path planning based on improved genetic algorithm. 2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control; Shenyang, China; 2013.

[16] Agarwal A, Lim MH, Er MJ, et al. ACO for a new TSP in region coverage. IEEE/RSJ International Conference on Intelligent Robots and System, Aug 2–6; Edmonton. 2005; p.1717–1722.

[17] Chen XS, Lim MH, Ong YS. An ant colony system algorithm for path planning in sparse graphs. International Conference on Intelligent and Advanced Systems; Kuala Lumpur, Malaysia; 2007.

[18] Ugur A. Path planning on a cuboid using genetic algorithms. Inf Sci. 2008;178:3275–3287.

[19] Haupt RL, Haupt SE, Haupt SE. Practical genetic algorithms. 2nd ed. New York: Wiley; 1998.

[20] Chambers LD. The practical handbook of genetic algorithms: applications. Boca Raton (FL): Chapman & Hall/CRC; 2000.

[21] Kuo RJ, Syu YJ, Chen ZY, et al. Integration of particle swarm optimization and genetic algorithm for dynamic clustering. Inf Sci. 2012;195:124–140.

[22] Paul PV, Moganarangan N, Kumar SS, et al. Performance analyses over population seeding techniques of the permutation-coded genetic algorithm: an empirical study based on traveling salesman problems. Appl Soft Comput. 2015;32:383–402.

[23] Majumder AK, Bhunia AK. Genetic algorithm for asymmetric traveling salesman problem with imprecise travel times. J Comput Appl Math. 2011;235(9): 3063–3078.