SCIPEDIA Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

Revista Internacional Métodos numéricos para cálculo y diseño en ingeniería RIMNI

In cooperation with CIMNE

A Bi-Objective Evacuation Path Optimization Method Based on Meta-Heuristic Integration

Guoqing Yang* **and Jialu Chen**

School of Control and Mechanical Engineering, Tianjin Chengjian University, Tianjin, 300384, China

*Corresponding Author: Guoqing Yang. Email: tjcj2008@163.com

ABSTRACT

Urban underground public spaces are relatively enclosed, with high daily passenger flow and diverse directions. Enhancing evacuation capabilities through dynamic guidance systems is a crucial solution. Therefore, this paper proposes a double-layer computational framework based on meta-heuristic integration of a bi-objective evacuation path optimization method. Firstly, the outer layer computation explores the population by combining the genetic algorithm and simulated annealing. It uses temperature parameters to probabilistically accept worse solutions, enhancing global search capability and avoiding local optima. Secondly, the inner layer calculates the fitness value of each solution from the outer layer, considering both the shortest path and minimum risk objectives. The path optimization is carried out by A∗ algorithm on the basis of constructing the safety matrix by using breadth-first search algorithm, and the distance matrix by using Dijkstra's algorithm. Finally, this method can search for the shortest safe path, effectively guiding people safely from the starting point to the exit. The experiments show that compared to traditional evacuation path planning methods, the proposed method significantly improves path optimization capabilities, quickly planning the shortest and safest evacuation route. It can provide guidance for fire safety and emergency plans for evacuations.

OPEN ACCESS

Accepted: 14/09/2024

Submited: 19/07/2024

DOI
110.23967/
j.rimni.2024.10.56332

Keywords:
Intelligent evacuation system; genetic algorithm; simulated annealing algorithm; A algorithm; bi-objective path optimization

1 Introduction

With the development and utilization of urban underground public spaces, these areas have become increasingly complex and deeply integrated into the urban fabric. These spaces have evolved from small-scale, single-function underground projects to comprehensive urban underground complexes that integrate commercial, entertainment, recreational, transportation, and parking functions [1]. This evolution not only contributes to the sustainable development of the urban environment but also addresses the growing demands for a safer, more advanced, and modern urban lifestyle.

Urban underground complexes characterized by multifunctionality, intricate internal structures and limited exits, often become densely populated during operation. As a result, in the event of a fire, unfamiliarity with the building layout, limited visibility, and psychological panic can severely reduce

evacuation efficiency [2]. Thus, throughout both the construction and operational phases of urban underground complexes, preventing disasters in advance and ensuring safe, efficient evacuation remain significant challenges.

Emergency signs are a crucial component of the evacuation guidance systems installed within buildings. They provide pedestrians with effective path guidance and evacuation information and have been widely implemented in large buildings, including subway stations, airports, and shopping malls, to address wayfinding and evacuation issues [3]. Kobes et al. [4] analyzed the effects of smoke and emergency signage on pedestrian behavior during fire evacuations, with a specific focus on path selection. Results from 83 evacuation trials indicated that 81.8% of individuals relied on emergency signage to select their evacuation path, when smoke obscured the indoor environment. Traditional evacuation guidance systems primarily employ static, passive emergency signage that directs people to the nearest exit, irrespective of the fire location or crowd congestion. This can misdirect individuals towards the fire source, potentially resulting in chaotic evacuations and blocked exits [5]. In the face of rapidly changing dangers during emergencies, such conventional evacuation systems are no longer suitable for complex buildings. Therefore, a more flexible and intelligent path planning method is needed, capable of dynamically adjusting evacuation routes according to real-time situations to improve evacuation efficiency and ensure personnel safety.

With the advancement of technologies such as wireless sensor networks, cloud big data, and artificial intelligence, more researchers have focused on system-level designs for evacuation guidance. Examples include interactive wayfinding systems [6,7] and personal mobile devices [8]. Intelligent evacuation guidance systems [9] can dynamically generate guidance strategies based on the emergency scenario, crowd movement trends, and the internal building environment. These systems can plan optimal paths and identify safest exits during emergencies, thereby improving evacuation efficiency [10]. For instance, Al-Nabhan et al. [11] utilized IoT and wireless sensor networks to facilitate dynamic emergency evacuation. Sensor nodes detect and monitor environmental conditions, while local routing controllers and cloud systems make emergency navigation decisions, dynamically adjusting emergency signage during evacuation. Most research in evacuation guidance focuses on reduce total evacuation time, minimize average escape time, or maximize cumulative throughput at exits within a specific time frame. However, the dynamic nature of disaster environments makes it impractical to rely solely on the shortest path or minimum time for evacuation. In terms of optimizing exit selection behavior in evacuation guidance systems, Lopez-Carmona et al. [12] proposed and evaluated an adaptive guidance system that displays guidance in the form of colors on personal or wearable devices, directing evacuees to exits with corresponding color indicators. Anyway, this approach may be unreliable in actual emergencies, as it cannot guarantee that every user will have a mobile device with the necessary evacuation software installed.

Integrating monitoring control, artificial intelligence and dynamic path algorithms within intelligent evacuation guidance systems can significantly enhance evacuation efficiency during emergencies, thereby improving overall safety. Develop a core intelligent evacuation path-planning algorithm for smart evacuation systems based on IoT technology is essential. This paper proposes a double-layer computational framework for a bi-objective evacuation path optimization method, based on the integration of meta-heuristics. The outer layer employs a combination of Genetic Algorithm (GA) and Simulated Annealing Algorithm (SAA), while the inner layer utilizes A$*$ algorithm for path planning on a composite matrix. Accordingly, this method is referred to as GSA-A$*$. The shortest and safest evacuation paths can be dynamically displayed through emergency signage in future studies. The primary contributions of this work are summarized as follows:

(1) In this study, evacuation scenarios in urban underground public spaces are transformed into three-dimensional grid maps under hypothetical conditions to simplify model calculations.

(2) For a given input scenario, this approach constructs distance and safety matrices, analyzes the risk impact at each position within the matrices, and employs a bi-objective evacuation path optimization method based on meta-heuristic integration to find the shortest and safest path.

(3) Compared to traditional evacuation path planning methods, the proposed approach significantly enhances path optimization, enabling the rapid planning of the shortest and safest evacuation routes.

The remainder of this paper is organized as follows: Section 2 presents an overview of recent methods and models for evacuation route guidance. Section 3 details the proposed evacuation path optimization method. Section 4 discusses the simulation results and analysis, and Section 5 concludes the paper.

2 Literature Review

Currently, evacuation plans for urban public areas rely predominantly on pre-established static routes, typically coordinated on-site by designated personnel. As a fire progresses, these routes become increasingly vulnerable. Consequently, a growing number of researchers are focusing on intelligent evacuation guidance systems capable of recommending safe evacuation routes based on real-time indoor conditions, utilizing effective path planning algorithms to enhance safety and security. This section provides an overview of recent methods and models proposed for evacuation route guidance.

In recent years, many researchers have been combining IoT with cloud big data technologies to plan evacuation strategies. For example, Chen et al. [8] established an IoT-based time-efficient indoor navigation and evacuation framework, it plans the fastest route to any target location on smart handheld devices. Similarly, Yoo et al. [13] investigated a machine learning-based indoor augmented reality (AR) navigation and emergency evacuation system. This system estimates an individual's current location, predict disaster spread, and guide users to the best escape route. However, these approaches may prove unreliable in actual emergencies, as there is no guarantee that every user will have a mobile device with the necessary evacuation software installed and functional. In contrast, Kamoun et al. [14] tackled the evacuation challenges in multi-exit exhibition centers by proposing a dynamic intelligent evacuation model to address fire spread. The system activates intelligent panels at key corridor intersections to provide visual instructions, indicating the correct escape direction and guiding individuals to follow the optimal path.

Haghani [15] categorized evacuation optimization into three main research directions, highlighting mathematical programming methods as a key approach for optimization. These methods typically aim to design optimal path planning solutions for specific infrastructure and environmental settings, primarily based on macroscopic network models of evacuation (i.e., graph networks composed of edges and nodes). When a critical path segment is affected, the original evacuation route becomes unsuitable for the emergency, and it takes a lot of time to find a new optimal path. Deng et al. [16] explored a real-time dynamic path optimization method based on an improved A$*$ algorithm. Initially, a network topology is established, and the A$*$ algorithm is applied to generate an a priori evacuation network consisting of all optimal paths. Subsequently, the FDS software is used to simulate the dynamic spread of fire, removing nodes affected by the fire and calculating evacuation speed. Finally, the evacuation paths are recalculated by updating the weights. To reduce the search time, Lin et al. [17] depicted a graph-based indoor network simplification method, i-GIT, which reduces computation

time. With the advancements in stochastic optimization, various intelligent algorithms such as ant colony optimization [18], particle swarm optimization [19], artificial bee colony algorithm [20], genetic algorithms [21], and greedy algorithms [22] have been used to find dynamic paths.

Fang et al. [23] described the evacuation network within a stadium as a layered directed network, and proposed an improved ant colony optimization algorithm to solve the multi-objective evacuation path problem by considering three objectives: minimizing total evacuation time, minimizing total evacuation distance, and minimizing congestion. Xu et al. [24] developed a fire model for a single-level supermarket using Pathfinder software, simulating critical parameters such as temperature, CO concentration, and smoke density during a fire. They implemented optimal path planning through an Improved Ant Colony Optimization (IACO) algorithm and compared its performance to the basic Ant Colony Optimization (ACO) algorithm. Their findings indicated that IACO effectively avoids hazardous areas. Yusof et al. [25] proposed a waypoint path planning method using the Particle Swarm Optimization (PSO) algorithm, which minimizes total path cost from the initial position to the final destination based on predefined waypoints, offering path guidance for visually impaired individuals. While ACO has certain advantages in pathfinding, it heavily relies on feedback from historical information, making it susceptible to the influence of the initial solution. Conversely, although PSO converges rapidly, it may become trapped in local optima in multi-objective optimization due to insufficient diversity.

In the context of path optimization, Cepolina [26] reformulated the path optimization problem into minimizing a cost function, which represents the movement time of the last evacuee in a given scenario, and applied the Simulated Annealing Algorithm (SAA) to find the optimal set of evacuation routes, i.e., routes with the minimal movement time for the last person. In robot path planning for unknown environments, Zhang et al. [27] enhanced robot navigation speed by integrating SAA with ACO to guide the robot in reaching specific destinations via collision-free paths. This approach proved to be highly robust and adaptable. SAA is an effective optimization technique capable of handling cost functions with characteristics such as nonlinearity, discontinuity, and stochasticity [28]. Additionally, SAA possesses advantages such as efficiency, simplicity, and versatility, making it applicable to a wide range of combinatorial optimization problems.

The Genetic Algorithm (GA) is a non-conventional meta-heuristic approach based on natural genetics and selection mechanisms. Over the past decade, GA has been widely used to solve a variety of practical problems, owing to its simplicity and global search capabilities [29]. Sriniketh et al. [30] used GA to optimize a modeling framework and proposed a path planning method for autonomous mobile robots to guide evacuees to the nearest stair exit. Simulation results showed that the proposed algorithm significantly outperformed traditional methods in terms of the distance traveled by the robot. Additionally, Xiong et al. [31], in their research on adaptive ocean sampling problems, emphasized that evolutionary algorithms like GA are also suitable for obstacle avoidance path planning for autonomous marine vehicles. GA is particularly effective in continuous optimization problems, iteratively updating control point positions to search for optimal solutions.

The diversity maintenance mechanism of GA plays a crucial role in exploring a wide range of evacuation path solutions. This mechanism enables a broad coverage of the search space, considering various potential evacuation routes. However, despite the randomness inherent in GA that promotes diversity, it may still risk becoming trapped in local optima, especially in complex optimization problems. This underscores the importance of integrating SAA. In the context of evacuation path optimization, this process is analogous to a scenario where, in the initial phase of an emergency evacuation, a crowd might select suboptimal paths to quickly avoid danger zones. As the evacuation

progresses, individuals gradually adjust their paths, ultimately converging on safer evacuation routes. The integration of GA and SAA facilitates both rapid responses to the multi-objective demands of complex evacuation scenarios and gradual optimization for greater safety and efficiency.

The aforementioned studies inspired us to develop a bi-objective dynamic path optimization method based on meta-heuristic integration, aimed at enhancing the path guidance functionality of intelligent evacuation systems. Breadth-First Search (BFS) is employed to construct a safety matrix from any node to the fire source, while Dijkstra's algorithm calculates the distance matrix from each node to its nearest exit. Finally, the dual objectives of minimizing distance and risk are linearly combined to form a single-objective path cost function, globally optimized using A$*$ algorithm.

The proposed method in this paper includes the following two characteristics in evacuation path planning:

(1) Short Path: Compared with paths that only consider the shortest distance, in this paper, the evacuation path from any node should not be excessively long, while also considering safety.

(2) Safe Path: The path planned from each node must prioritize safe, sufficiently distant from dangerous areas.

3 Method

3.1 Model Construction

This model uses the given simulated scenarios, including evacuation starting points and exit locations, as input data, this study constructs a grid map consisting of both a distance matrix and a safety matrix. Key parameters for GSA-A$*$ are then configured accordingly, including population size, crossover rate, mutation rate, initial temperature, and cooling rate. Finally, the fitness function derived from the optimized solution is evaluated and compared using performance metrics such as path length and safety level. To simplify the model calculations, this paper makes the following assumptions:

(1) Sensor nodes are uniformly deployed across public infrastructure to form a matrix of size $N = m \times n$, where m and n are the number of rows and columns, respectively.

(2) The distance between any two adjacent sensor nodes is $1m$, with each sensor positioned at the center of a grid cell.

(3) All doors remain open under normal conditions, and the exits are secure.

(4) Upon detecting a fire, sensors immediately upload data, and intelligent signage promptly responds to received instructions.

The intelligent node coordinate information is used to construct the map matrix, represented as a connected undirected graph $G(V, E)$, where $V = \{1, 2, 3, \ldots, n\}$ represents the set of nodes and E represents the set of edges. A$*$ algorithm is applied to obtain the minimum-cost path between any two intelligent nodes, recording the shortest safe path sequence and outputting the results.

3.2 Model Solution

3.2.1 Outer Layer Calculation

Genetic Algorithm

Genetic Algorithm (GA) is an optimization algorithm that emulates the principles of natural selection and genetic evolution. The core concepts of GA in this paper include the following key steps:

Step1: Individual Representation. Each chromosome represents a potential evacuation route from the starting point to the exit.

Step2: Fitness Evaluation. Determine the individual's fitness value.

Step3: Selection Operation. Individuals with higher fitness values are selected for reproduction.

Step4: Crossover Operation. A single-point crossover method is used to produce offspring (as shown in Fig. 1).

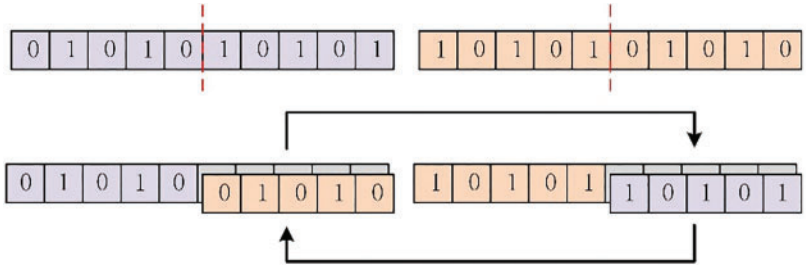

Figure 1: The single-point crossover method

Step5: Mutation Operation. The offspring undergo swap mutation (as shown in Fig. 2).

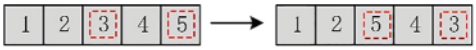

Figure 2: The swap mutation method

Step6: Replacement Strategy. Formation of a new generation of the population based on fitness values.

Simulated Annealing Algorithm

Simulated Annealing Algorithm (SAA) is a heuristic optimization algorithm inspired by the physical annealing process in solids. The key steps of SAA in this paper are as follows:

Step1: Initialization. Generate an initial solution randomly from the solution space.

Step2: Initial Temperature Setting. Set the initial temperature T_{init} and determine the exponential cooling schedule.

Step3: State Update. Generate a neighboring solution using a neighborhood search strategy.

Step4: Acceptance Criterion. Accept the neighboring solution with a certain probability P.

Step5: Cooling Process. Gradually reduce the system temperature, decreasing the probability of accepting worse solutions over time.

Step6: Termination Condition: The algorithm terminates when the temperature reaches $T_{\min}$.

GSA-A∗ Algorithm

In this study, GA is employed for the initialization, crossover, and mutation operations to achieve diverse searches and optimizations for the path planning problem. SAA is employed to balance exploration and exploitation by adjusting temperature parameters and the strategy of accepting inferior solutions, thus achieving global search and optimization for the path planning problem.

Integrating GA and SAA, this study employs A∗ algorithm to guide the fitness function updates, addressing the bi-objective problem in path planning. The flowchart of GSA-A∗ is shown in Fig. 3.

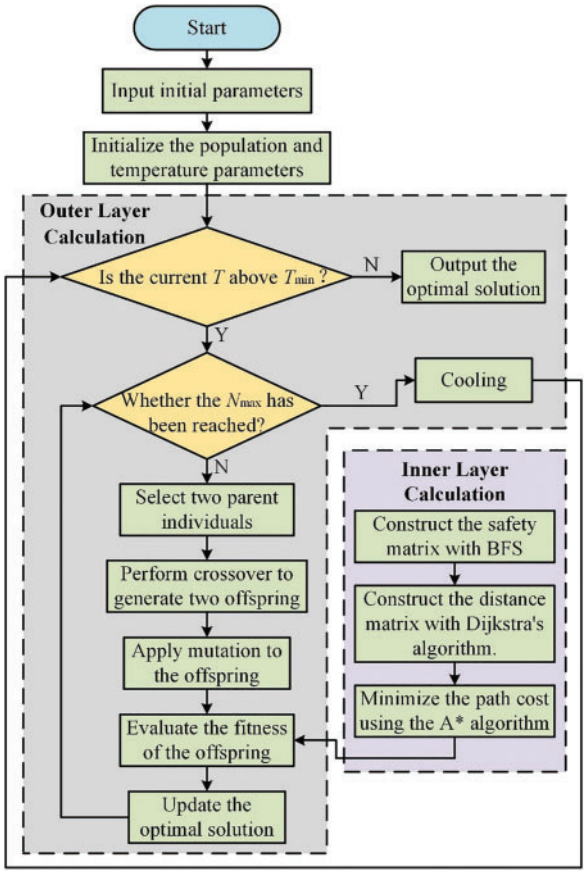

Figure 3: Diagram of the double-layer computational framework of GSA-A∗

(1) Parameter Definition and Variable Description

Several parameters are involved in the execution of GSA-A∗ algorithm. For instance, T_{init} determines the algorithm's exploratory capability and the probability of accepting suboptimal solutions at the start. If T_{init} is set too high, the early search results may be overly dispersed, potentially missing the optimal solution. Conversely, if T_{init} is too low, the algorithm is prone to getting trapped in local optima. After reaching $N_{\max}$ in each evolution cycle, the temperature decreases according to the cooling rate α. The algorithm terminates when the temperature reaches $T_{\min}$. While increasing $N_{\max}$ does not improve efficiency and may lead to excessive computational resource consumption, whereas a small $N_{\max}$ may prevent the algorithm from finding the global optimum. When α is closer to 1, the slower cooling process facilitates escaping local optima but slows convergence. Conversely, if α approaches 0, rapid cooling diminishes the algorithm's exploratory capability, leading to a decline in solution quality. If $T_{\min}$ is set too high, the algorithm may terminate prematurely without adequate optimization, whereas a lower $T_{\min}$ increases computation time. The parameter β reflects the selection pressure between superior and inferior individuals during evolution; a larger difference leads to a greater tendency to select individuals with higher fitness. The parameter N_{pop} affects the diversity of the initial solution set, while P_{mut} determines the diversity of solutions. Excessive or insufficient values for these

parameters can lead to suboptimal algorithm performance. To balance global exploration and prevent premature convergence, the parameter values used in this study are detailed in Table 1. Table 1 shows the parameter values and definitions for the algorithm proposed in this paper, and Table 2 provides the formula variables and definitions.

Table 1: Parameter values and definitions for GSA-A∗

Parameter	Value	Definition
α	0.7	Cooling rate, controls the cooling speed of SAA
β	0.8	Controls the selection pressure of GA
N_{pop}	100	Population size
P_{mut}	0.1	Mutation rate
$N_{\max}$	60	Maximum number of iterations per evolution round
T_{init}	1	Initial temperature
$T_{\min}$	0.001	Minimum temperature

Table 2: Formula variables and definitions

Parameter	Definition
ω_L	Distance weight
ω_S	Safety weight
K	Path evaluation parameter determined by the number of exits
P	Acceptance probability of suboptimal solutions
Δf	Fitness delta
f_c	Offspring fitness
T	Current temperature
$S_{v_i(x,y)}$	Safety value of node i at coordinates (x, y)
$Risk_{v_i}$	Risk value of node i
$g\,(v_n)$	Cost function for distance in A∗ algorithm
$h\,(n)$	Heuristic function for distance in A∗ algorithm
$f\,(v_n)$	Evaluation function for distance in A∗ algorithm
$Total$	Objective function value

(2) Fitness Function

The fitness value refers to the measure of an individual's advantage in the survival of a population, as calculated by the fitness function. In this study, the fitness function is used to evaluate the quality of each individual. Also referred to as the evaluation function, it is derived from the objective function. The objective function calculates the total path cost by combining and weighting path distance and path safety, as demonstrated in the inner calculation below. A lower total path cost, or equivalently, a lower fitness function value, indicates a superior individual and a more optimal path. The fitness function serves two purposes: first, it provides a combined measure of path distance and

path security. Second, it guides the algorithm's optimization process by selecting superior individuals and determining whether to accept new solutions.

(3) Update Strategy

In this paper, the optimal solution is updated based on the fitness function of the offspring and the simulated annealing criteria. If the fitness value of the offspring is superior, it is accepted directly; otherwise, a random number is introduced. If the generated random number $[0, 1)$ is less than the acceptance probability of the suboptimal solution P, the latter will be accepted. The probability of accepting a suboptimal solution is determined by the difference in fitness values and the current temperature. The acceptance probability of a suboptimal solution is shown in Eq. (1).

$$P = e^{\frac{\Delta f}{T}} \tag{1}$$

$$\Delta f = f_{c1} - f_{c2} \tag{2}$$

where T is the current temperature, Δf is the fitness difference between the offspring, f_{c1} and f_{c2} are the fitness values of offspring 1 and offspring 2, respectively. Selecting the difference between the two offspring enhances the population competition.

3.2.2 Inner Layer Calculation

Safety Matrix

To ensure the evacuation path safety, BFS algorithm constructs a safety matrix by leveraging its layer-by-layer expansion feature. This matrix helps calculate the safety value of each node. As shown in Fig. 4, cells with a purple border represent exits, while cells filled in black represent obstacles. Starting from the fire source cell, which is assigned a safety value of 0, the outer layer is assigned a value of 1, the next outer layer is assigned a value of 2, and so on. BFS is employed to expand outward from the fire source layer by layer until all reachable locations are covered. Safety values increase with distance from the fire source, indicating a higher degree of safety. This value can be adjusted based on the type of incident or through sensitivity analysis to assess path safety, aiding subsequent path planning to balance safety considerations.

In this method, exponential functions and reciprocal scaling methods are applied during computation to normalize safety values. Parameter values are standardized by adjusting a constant K to control the normalization range, ensuring that the data fall within the desired intervals. Eq. (3) represents the normalized value (risk value) of each safety value.

$$Risk_{v_i} = \begin{cases} \infty, & S_{v_i(x,y)} = 0 \\ \dfrac{k}{e^{S_{v_i(x,y)}} - 1}, & S_{v_i(x,y)} > 0 \end{cases} \tag{3}$$

$S_{v_i(x,y)}$ indicates the safety value at the node's coordinates. A safety value of 0 indicates that the node is a fire source with an infinitely high risk value, while a safety value greater than 0 indicates that the node is passable, with larger safety values corresponding to smaller risk values.

Distance Matrix

In this study, the map matrix is constructed based on the assumed coordinates of the smart nodes. Each cell in this matrix is initially set to infinity. Starting from the exits, Dijkstra's algorithm is applied to calculate the distance between each node and its nearest exit. If a cell is a fire source, its value

remains infinite. Subsequently, the shortest path length from the evacuation starting point to a safe exit is selected by searching the distance matrix, as shown in Fig. 5.

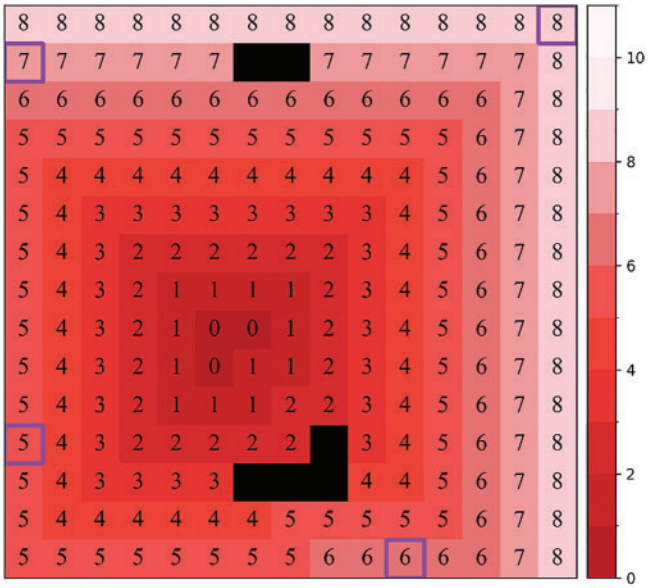

Figure 4: Safety matrix built with BFS

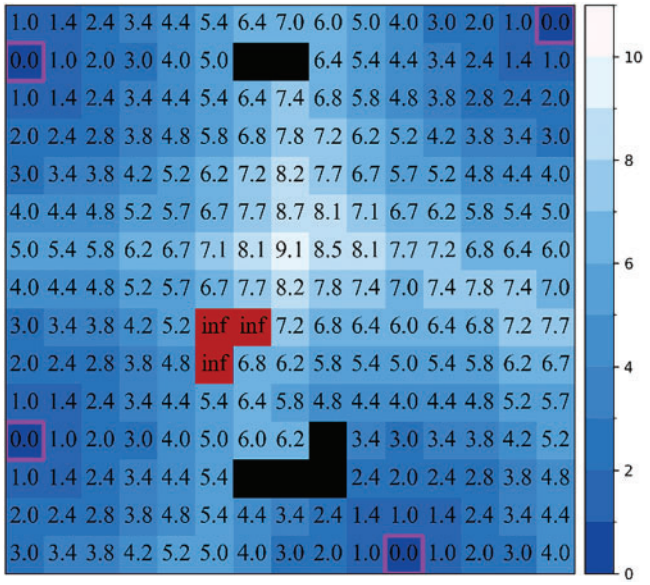

Figure 5: Distance matrix built with the Dijkstra's algorithm

Pathfinding

A∗ algorithm is a heuristic search method used to calculate the minimum path cost from the start to the endpoint. During the path search, a heuristic function estimates the cost between the current node and the target node. Common heuristic functions use node distance as a measurement unit to

guide or heuristically search the path. Common methods for distance calculation include Manhattan distance, diagonal distance and Euclidean distance. Since node expansion is required in all directions, this study selects Euclidean distance as the heuristic function, with each node's routing direction illustrated in Fig. 6.

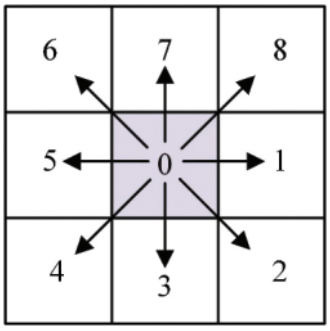

Figure 6: Eight pathfinding directions of node 0

The basic formula for A∗ algorithm is $f(v_n) = g(v_n) + h(v_n)$ where v_n represents the current node, $f(v_n)$ is the evaluation function, which is the combined cost of node v_n; $g(v_n)$ is the cost function, which is the actual cost from the start point to node v_n; $h(v_n)$ is the heuristic function, which is the estimated cost from node v_n to the endpoint. The algorithm tends to select nodes with the lowest combined cost for routing. On the grid map, the actual cost and estimated cost are calculated as shown in Eq. (4).

$$\begin{cases} g(v_n) = \sum_{i=1}^{n} \sqrt{(v_i(x) - v_{i-1}(x))^2 + (v_i(y) - v_{i-1}(y))^2} \\ h(n) = \sqrt{(v_n(x) - v_g(x))^2 + (v_n(y) - v_g(y))^2} \end{cases} \tag{4}$$

where $v_n(x)$ and $v_n(y)$ are the coordinates of the current node. $v_g(x)$ and $v_g(y)$ are the node coordinates of the end point.

The computational flow of A∗ algorithm is depicted in Fig. 7.

Objective Function

In this study, the path planning process takes into account both path length and safety. These factors are linearly combined into an objective function, and A∗ algorithm is used to minimize the objective function, thereby calculating the optimal path from each starting node to the exit. The objective function is shown in Eq. (5).

$$Total = \sum_{i=1}^{n} \omega_L \times f(v_i) + \omega_S \times Risk_{v_i} \tag{5}$$

where *Total* represents the total path cost. The total distance cost $\sum_{i=1}^{n} f(v_i)$ represents the path length guided by each smart node, with larger values indicating longer path distances. The total risk cost $\sum_{i=1}^{n} Risk_{v_i}$ represents the path safety guided by each smart node, with larger values indicating lower safety.

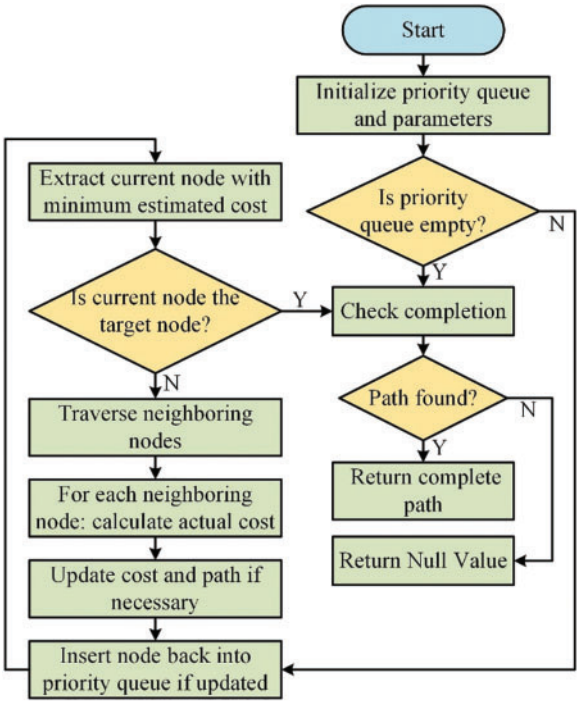

Figure 7: Computational framework of the A∗ algorithm

4 Experimental Results and Analysis

4.1 Scene Setting

This section uses the given simulation scene, personnel starting point and exit location as input data, and uses the grid map method to output the matrix model. Taking the Hongqi South Road subway station in Tianjin, China, as the prototype, the constructed 3D environment is shown in Fig. 8, which is F1 station hall level, B1 platform level and B2 platform level from top to bottom. In the figure, black squares represent obstacles, green squares represent safe exits, yellow squares indicate staircase connections, and adjacent floors are connected by stairs shown in light grey. If a fire occurs, the unit nodes turn red.

4.2 Program Settings

When a fire occurs on different floors or in various locations, the weight coefficients of the objective function can be adjusted to set different response plans. The plan settings are as follows:

Plan 1: Considers both evacuation distance and safety, with a greater emphasis on safety ($\omega_L = 0.4$, $\omega_S = 0.6$).

Plan 2: Prioritizes the shortest evacuation distance without considering the impact of the fire ($\omega_L = 1$, $\omega_S = 0$).

4.3 Scenario Analysis

As highlighted by Shi et al. [32], essential fire scenarios for metro evacuation include fires originating in trains stopped at the station, fires in public areas of the station (such as platforms, halls, and passageways), and fires occurring within railway tunnels. Furthermore, the editorial committee

emphasized the importance of conducting fire tests in station halls, platforms, and tunnel sections in China's 2023 national standard document GB/T 43392—2023. Accordingly, in this study, we strategically designed fire scenarios to evaluate evacuation strategies.

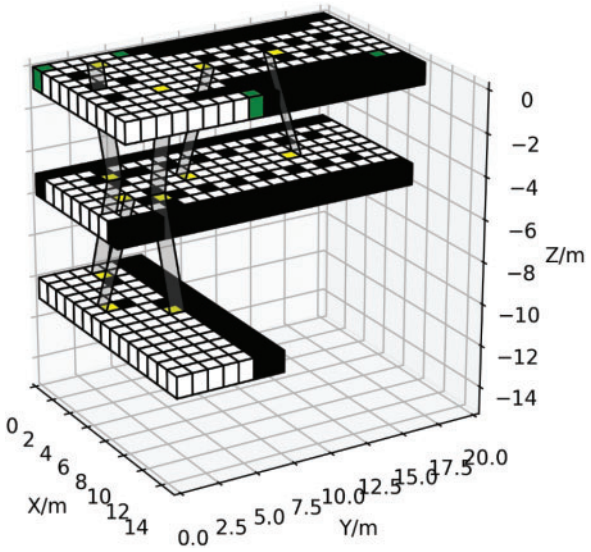

Figure 8: 3D model of Hongqi South Road Subway Station in Tianjin, China

This section analyzes different fire source locations and ranges within the 3D grid map of Hongqi South Road Subway Station. The recommended routes can subsequently be dynamically directed to evacuees through emergency signage.

4.3.1 Station Floor F1

In the first scenario, the fire source is positioned at the central cross-shaped area of the F1 station hall (shown in Fig. 9), a relatively spacious location that typically houses a service counter. This area serves as both a storage point for luggage and a location for various emergency supplies. Additionally, this zone experiences high passenger density during peak hours. Analyzing this scenario aids in understanding fire dynamic propagation patterns and developing more effective evacuation strategies.

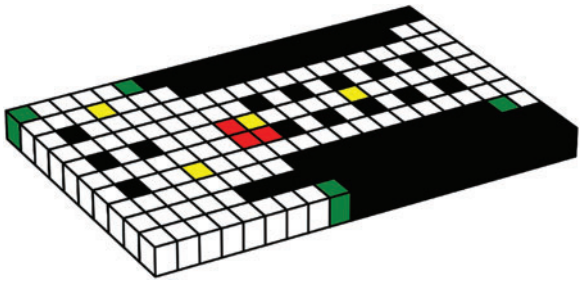

Figure 9: Fire location and fire source range on the F1 floor

The distance matrix and safety matrix are shown as (a) and (b) in Fig. 10, respectively. In Fig. 9, red squares represent the fire source, yellow squares indicate staircase connections, green squares denote

exits, and black squares represent obstacles. In Fig. 10, black cells represent obstacles, purple-bordered cells represent exits, and yellow-bordered cells indicate staircases. In the distance matrix, the value in each cell represents the distance to the nearest exit; red cells denote the fire source, where pedestrian access is prohibited, thus the distance is assigned to infinity. The closer a cell is to an exit, the smaller its distance value and the darker the fill color. In the safety matrix, the fire source is represented by a cell with a safety value of 0, radiating values to its surroundings; the further a cell is from the fire source, the higher its safety value and the lighter its fill color.

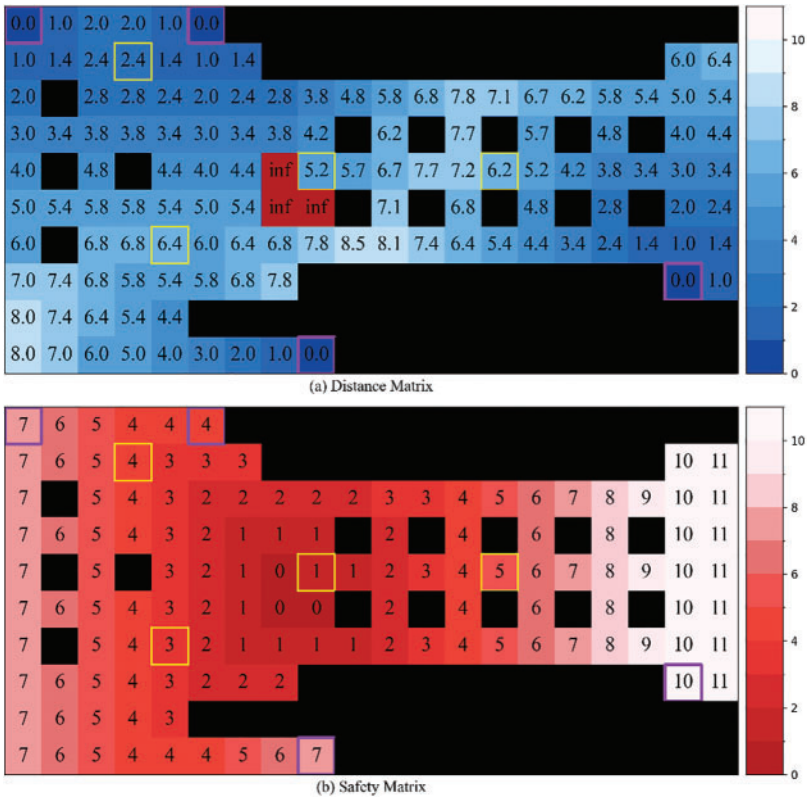

Figure 10: (a) Distance matrix and (b) safety matrix in the event of fire on the F1 floor

When a fire occurs on the F1 floor as shown in Fig. 9, Plan 1 is selected for the F1 floor. Simultaneously, to ensure efficient evacuation from the B1 and B2 floors, which are temporarily unaffected by the fire, Plan 2 is selected for these floors. In this section, four staircases on the F1 floor are selected as the starting points. GSA-A∗ algorithm proposed in this paper, along with G-A∗, SA-A∗ and ACO-A∗ as control algorithms, are used for path planning to guide personnel from the starting points to the exits. The parameters for GSA-A∗, G-A∗, and SA-A∗ are provided in Table 1. The parameters for ACO-A∗ are set as follows: the pheromone factor is $\alpha_{ACO} = 1.0$, the heuristic factor is $\beta_{ACO} = 2.5$, the pheromone evaporation rate is $\rho = 0.6$, the initial pheromone concentration is $\tau = 1.0$, and the number of iterations is 60. These parameter settings are consistently maintained in the subsequent analysis. The calculated fitness values, i.e., global path costs, are shown in Fig. 11, and the local path costs are detailed in Table 3. GSA-A∗ reaches a total of 1200 iterations, with the path cost converging to 6.89185480873227. G-A∗ completes 60 iterations, converging to a path cost of

8.49393635816158, while SA-A∗ completes 20 iterations, with a path cost of 9.69393635816158. ACO-A∗ completes 60 iterations, with a path cost of 8.57654567394356. To visually compare the differences among these algorithms, the first 150 iteration values of GSA-A∗ are uniformly selected. The path guidance results of GSA-A∗, G-A∗, SA-A∗ and ACO-A∗ are illustrated in Fig. 12.

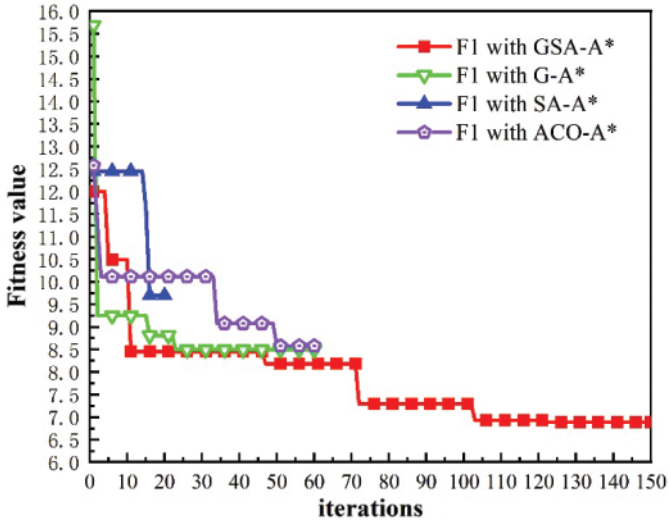

Figure 11: Comparison of fitness value (i.e., global path cost) results for the three different algorithms (GSA-A∗, G-A∗, SA-A∗ and ACO-A∗) for the F1 floor fire scenario

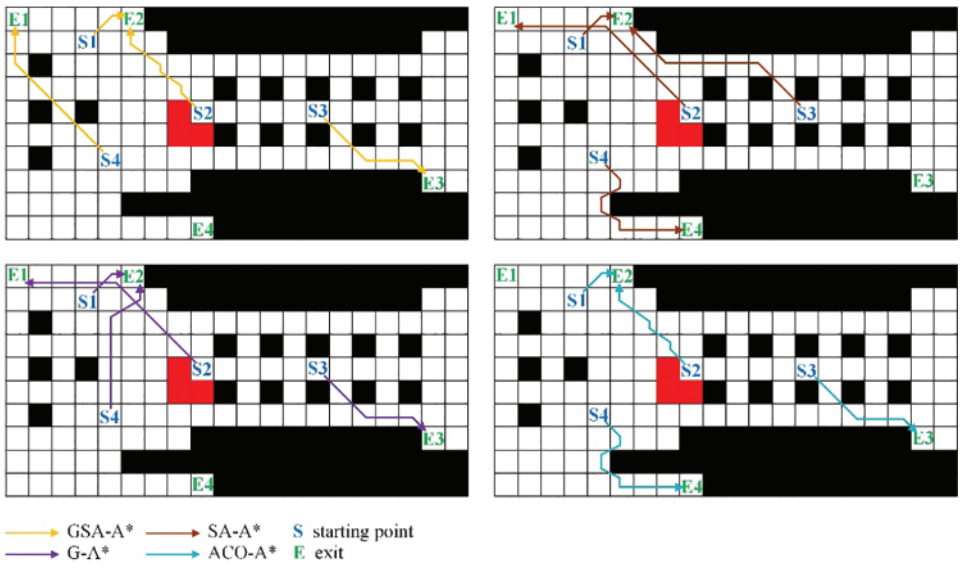

Figure 12: Comparison of path planning results for the three different algorithms (GSA-A∗, G-A∗, SA-A∗ and ACO-A∗) on the F1 floor fire scenarios

Table 3: Results of local fitness values (i.e., local path costs) for GSA-A∗, G-A∗, SA-A∗ and ACO-A∗ on the F1 floor fire scenarios

Algorithm	Starting point			
	S1	S2	S3	S4
GSA-A∗	0.844777664873058	1.644777664873060	2.000108964778420	2.402190514207730
G-A∗	1.202190514207730	1.644777664873060	3.244777664873060	2.402190514207730
SA-A∗	0.844777664873058	3.202190514207730	3.244777664873060	2.402190514207730
ACO-A∗	1.03731472072755	3.00182542850644	2.03731472072755	2.50009080398202

4.3.2 Platform Floor B1

In the second scenario, the fire source is located near a train carriage on the B1 platform floor, as shown in Fig. 13. Fires in train carriages may occur due to electrical equipment failures, intentional arson, or improper management. Researching this scenario helps develop crowd management and evacuation strategies while enhancing emergency response mechanisms.

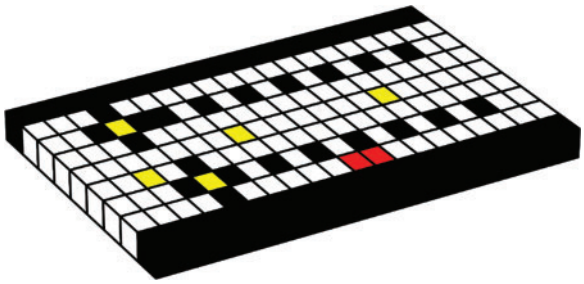

Figure 13: Fire location and fire source range on the B1 floor

The distance matrix and safety matrix are shown as (a) and (b) in Fig. 14, respectively. The color representation in Figs. 13 and 14 are the same as those described in Section 4.3.1. The passages enclosed by obstacles represent staircases on the B2 floor that directly connect to the F1 floor. Therefore, during personnel evacuation on the B1 floor, these two staircase interfaces are treated as obstacle nodes.

When a fire occurs on the B1 floor as shown in Fig. 13, Plan 1 is selected for the B1 floor. Simultaneously, to ensure the evacuation efficiency of the F1 and B2 floors, which are temporarily unaffected by the fire, Plan 2 is selected for these floors. In this section, one staircase entrance and two cells near the fire source on the B1 floor are selected as starting points. The algorithm comparison and parameter settings in this subsection remain consistent with those outlined in Section 4.3.1. The calculated fitness values, i.e., global path costs, are shown in Fig. 15, while the local path costs are detailed in Table 4. The total number of iterations is 1200 for GSA-A∗, 60 for G-A∗, and 20 for SA-A∗. The path costs for all three methods converge to 4.90765250066257, leading to consistent path guidance results. In contrast, ACO-A∗ converges to a path cost of 6.17942083443761 after 60 iterations, resulting in significantly longer evacuation distances in the path guidance outcomes. As shown in Fig. 16.

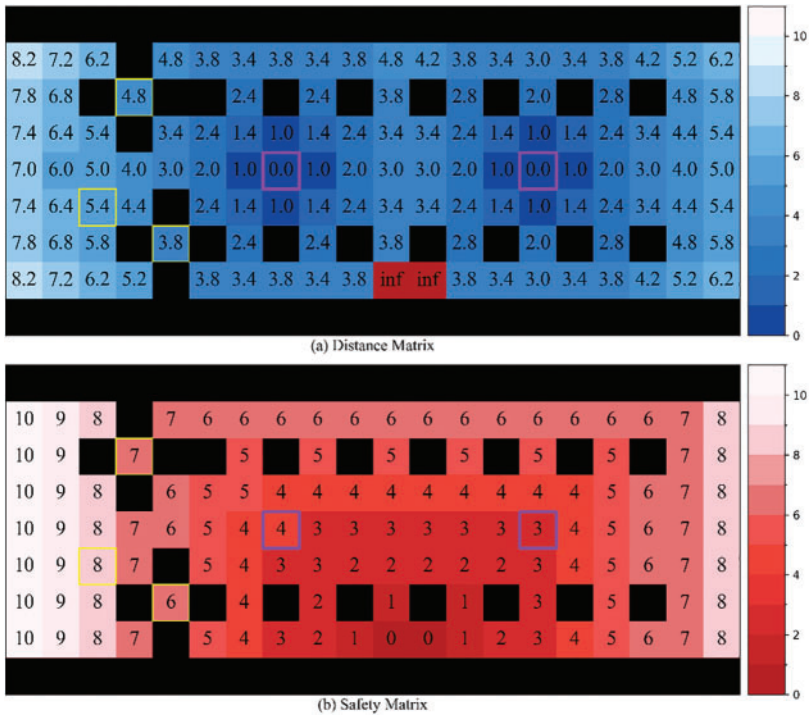

Figure 14: (a) Distance matrix and (b) safety matrix in the event of fire on the B1 floor

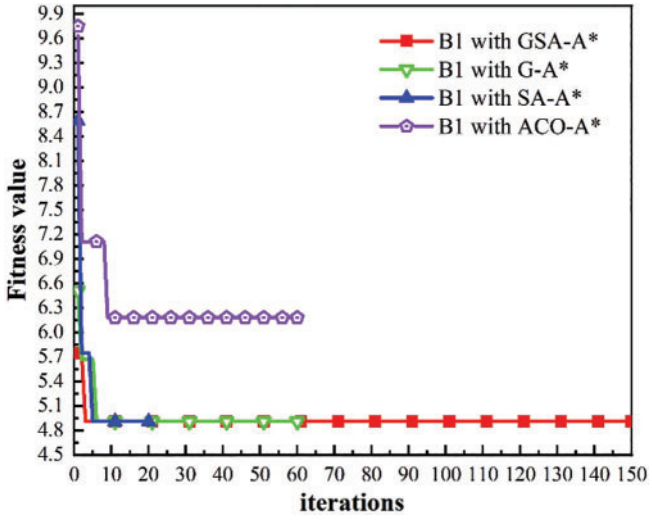

Figure 15: Comparison of fitness value (i.e., global path cost) results for the three different algorithms (GSA-A∗, G-A∗, SA-A∗ and ACO-A∗) for the B1 floor fire scenario

4.3.3 Platform Floor B2

In the third scenario, the fire source is positioned near the ends of the B2 platform (shown in Fig. 17), where facilities such as restrooms and control rooms are commonly situated. These areas are typically less frequented by passengers, which implies that fires may not be promptly detected during

their early stages. Analyzing this scenario facilitates the evaluation of evacuation route accessibility and the improvement of fire prevention measures.

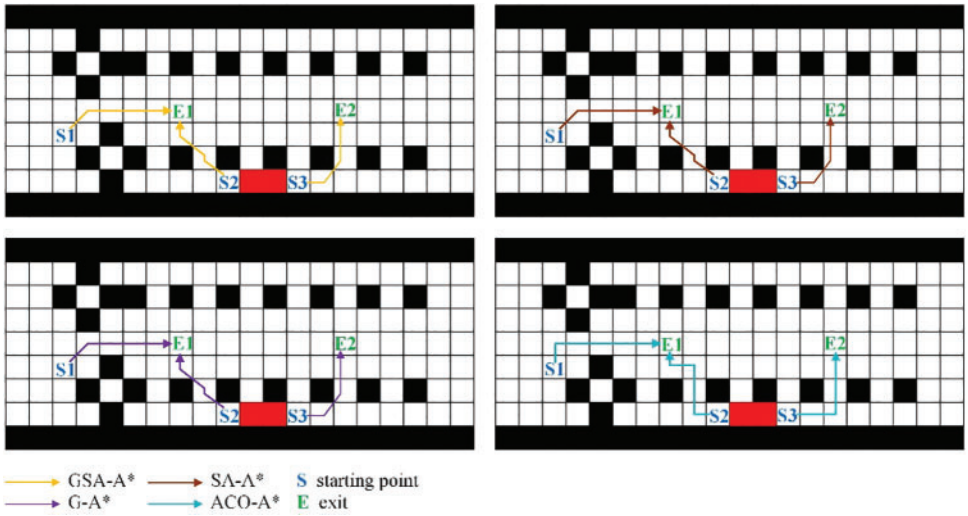

Figure 16: Comparison of path planning results for the three different algorithms (GSA-A∗, G-A∗, SA-A∗ and ACO-A∗) on the B1 floor fire scenarios

Table 4: Results of local fitness values (i.e., local path costs) for GSA-A∗, G-A∗, SA-A∗ and ACO-A∗ on the B1 floor fire scenarios

Algorithm	Starting point		
	S1	S2	S3
GSA-A∗	2.02238883243653	1.22238883243653	1.66287483578951
G-A∗	2.02238883243653	1.22238883243653	1.66287483578951
SA-A∗	2.02238883243653	1.22238883243653	1.66287483578951
ACO-A∗	2.53731472072755	1.53731472072755	2.10479139298251

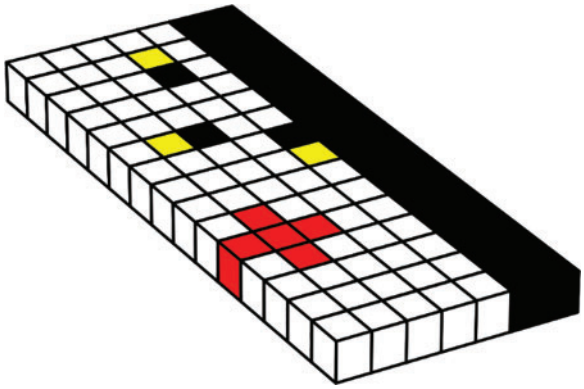

Figure 17: Fire location and fire source range on the B2 floor

The distance matrix and safety matrix are shown as (a) and (b) in Fig. 18, respectively. The color representation in Figs. 17 and 18 are consistent with those in Section 4.3.1.

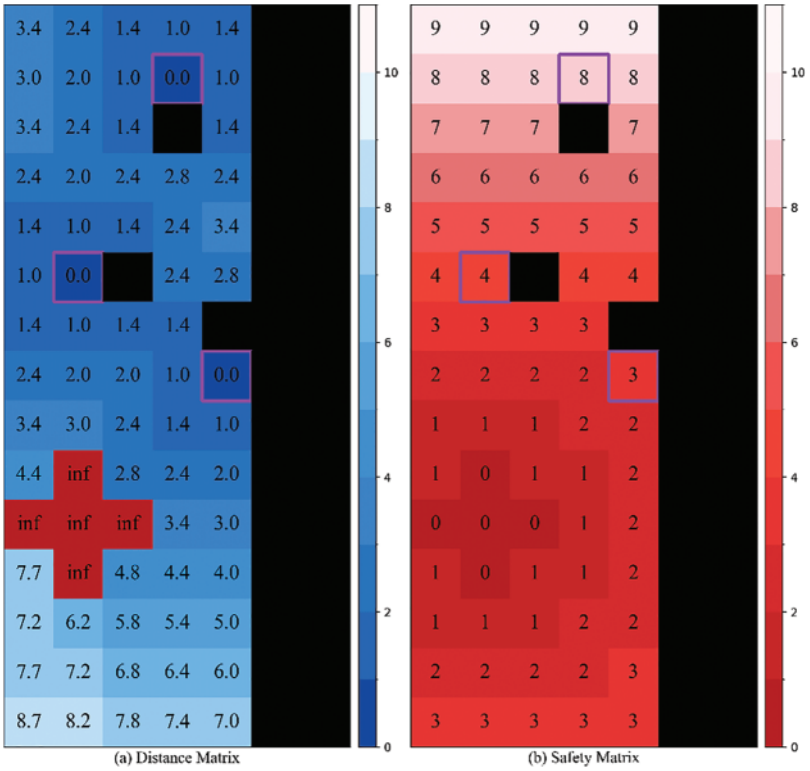

Figure 18: (a) Distance matrix and (b) safety matrix in the event of fire on the B2 floor

When a fire occurs on the B2 floor as shown in Fig. 17, Plan 1 is selected for the B1 floor. Simultaneously, to ensure the evacuation efficiency of the F1 and B2 floors, which remain temporarily unaffected by the fire, Plan 2 is selected for these floors. In this section, two cells near the stairs and two cells near the fire source on the B2 floor are selected as starting points. The algorithm comparisons and parameter settings in this subsection are consistent with those outlined in Section 4.3.1. The calculated fitness values representing global path costs, are shown in Fig. 19, while local path costs are detailed in Table 5. After 1200 iterations, GSA-A$*$ converges to a path cost of 5.42281179138483. In contrast, G-A$*$ achieves a path cost of 5.76208278635536 after 60 iterations, SA-A$*$ converges to a path cost of 5.42281179138483 after 20 iterations, and ACO-A$*$ converges to a path cost of 6.81504532934922 after 60 iterations. The path guidance results for GSA-A$*$, G-A$*$, SA-A$*$ and ACO-A$*$ are shown in Fig. 20.

4.4 Result and Discussion

As noted in Fig. 21, the convergence speed of the "F1 with GSA-A$*$" curve is notably slower compared to others, stabilizing only after approximately 100 iterations. A comparative analysis of the floor plans reveals that the F1 floor has four exits, the B1 floor has two exits, and the B2 floor has three exits. To determine whether the number of exits affects the convergence speed, simulations were conducted for the F1, B1, and B2 scenarios, with each scenario containing either two, three, or four exits, respectively. The fitness values obtained using GSA-A$*$ algorithm in this study are shown in

Fig. 22. The results indicate that the convergence speed of the fitness value curves for F1, B1, and B2 is indeed influenced by the number of exits. Specifically, as the number of exits increases, the number of potential paths from the starting points to the exits also increases. This results in a more prolonged exploration process and greater complexity, which in turn slows down the convergence speed.

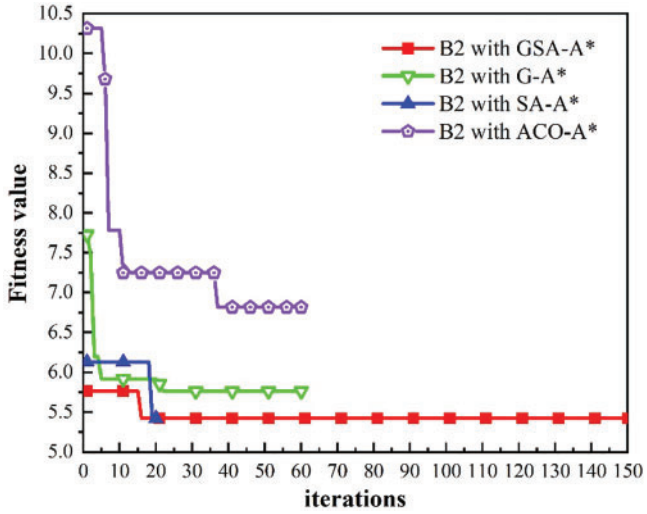

Figure 19: Comparison of fitness value (i.e., global path cost) results for the three different algorithms (GSA-A∗, G-A∗, SA-A∗ and ACO-A∗) for the B2 floor fire scenario

Table 5: Results of local fitness values (i.e., local path costs) for GSA-A∗, G-A∗, SA-A∗ and ACO-A∗ on the B2 floor fire scenarios

Algorithm	Starting point			
	S1	S2	S3	S4
GSA-A∗	1.20060403536151	0.894312253684261	1.63358324865479	1.69431225368426
G-A∗	1.20060403536151	1.23358324865479	1.63358324865479	1.69431225368426
SA-A∗	1.20060403536151	0.894312253684261	1.63358324865479	1.69431225368426
ACO-A∗	1.50067115040168	1.10479139298251	2.10479139298251	2.10479139298251

In the simulation scenario on the F1 floor, as shown in Fig. 11, the path cost calculated by GSA-A∗ not only has the lowest initial fitness value but also achieves the lowest final convergence cost. By contrast, G-A∗, SA-A∗, and ACO-A∗ exhibit slower convergence speeds, with significantly higher final path costs than the 6.89185480873227 achieved by GSA-A∗. The path guidance results are illustrated in Fig. 12, where all algorithms direct S1 to E2. Additionally, GSA-A∗ and ACO-A∗ also guide S2 to E2, whereas G-A∗ and SA-A∗ guide S2 to E1, significantly increasing the path length and reducing evacuation efficiency. Both GSA-A∗, G-A∗ and ACO-A∗ guide S3 to E3, avoiding the fire source and minimizing the path length, whereas SA-A∗ guides S3 to E2, violating the principle of the shortest safe path. GSA-A∗ and G-A∗ guide S4 to E1 and E2, respectively, while both SA-A∗ and ACO-A∗ direct S4 to E4. In this instance, aside from the highest local path cost observed with ACO-A∗, the local path costs for the other three algorithms are identical. Consequently, in this paper, the

corresponding risk values calculated from the safety matrix in Fig. 10b using Eq. (3) are approximately 0.33224, 0.66606, and 0.87044, respectively. Thus, the path planned by GSA-A∗ proves to be the safest.

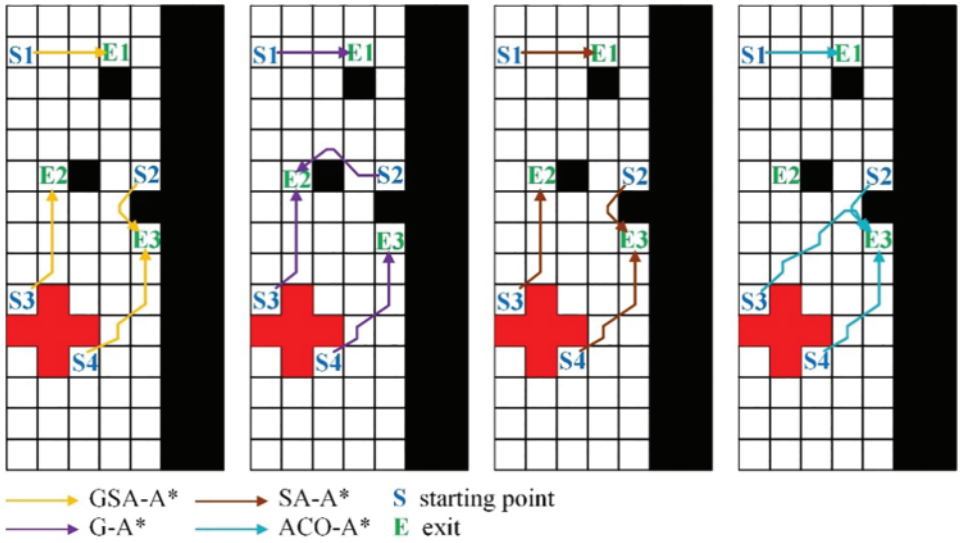

Figure 20: Comparison of path planning results for the three different algorithms (GSA-A∗, G-A∗, SA-A∗ and ACO-A∗) on the B2 floor fire scenarios

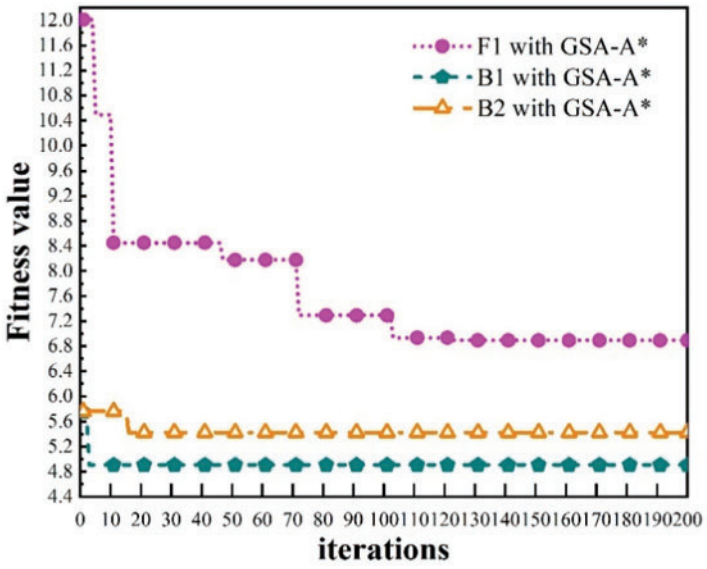

Figure 21: Comparison of fitness values (i.e., global path cost) results for GSA-A∗ for fire scenarios on the F1, B1, and B2 floors, respectively

In the B1 floor simulation scenario, the path cost calculated by ACO-A∗ is the highest. Although the other three algorithms converge to a path cost of 4.90765250066257, GSA-A∗ not only starts with the lowest initial solution but also rapidly decreases, which is not achieved by G-A∗ and SA-A∗. Since that the path costs are identical, the path guidance results, as shown in Fig. 15, are consistent among GSA-A∗, G-A∗, and SA-A∗, while the path planned by ACO-A∗ is slightly longer.

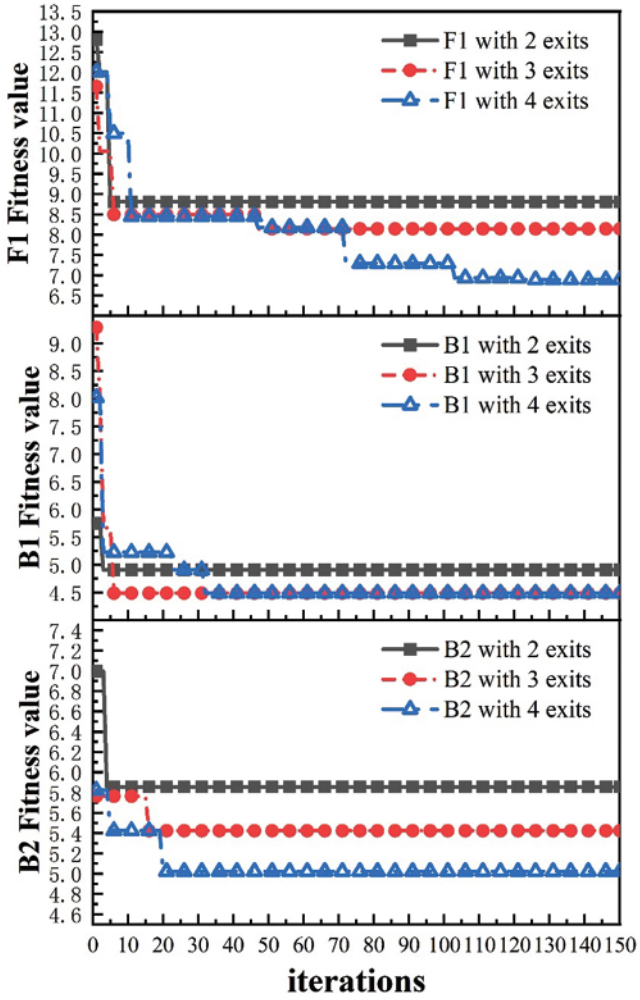

Figure 22: Comparison of fitness values (i.e., global path cost) results for GSA-A∗ for fire scenarios on the F1, B1, and B2 floors for different numbers of exits, respectively

In the B2 floor simulation scenario, ACO-A∗ exhibits both the highest initial fitness value and final convergence fitness value. In contrast, GSA-A∗ and SA-A∗ converge to a path cost of 5.42281179138483, while G-A∗ converges to 5.76208278635536, all of which significantly outperforming ACO-A∗. Although the results for GSA-A∗, G-A∗, and SA-A∗ are relatively close, GSA-A∗ demonstrates minimal fluctuation in fitness values throughout the iterations and achieves an initial path cost closer to the optimal solution. As shown in Fig. 20, the shortest safe paths planned by GSA-A∗ and SA-A∗ are identical, whereas G-A∗ guides S2 to E2, increasing the path distance. Additionally, it is evident that ACO-A∗ produces the poorest path results, directing S2, S3, and S4 to E3, which could cause exit congestion and reduce evacuation efficiency. Finally, as indicated in Table 5, GSA-A∗ achieves the lowest local path costs in this scenario.

Overall, GSA-A∗ not only demonstrates significant optimization effects in the early stages but also exhibits fast convergence and global optimal solutions in different scenarios. This demonstrates

its adaptability and effectiveness in path planning, as well as its ability to find the shortest safe paths more quickly.

5 Conclusion

This paper proposes a novel double-layer computational framework, GSA-A∗, which is a bi-objective evacuation path optimization method based on metaheuristic integration. This method efficiently identifies the shortest and safest routes in urban underground public areas during emergency events, thus enhancing the evacuation capabilities in such complexes. In this framework, the outer layer integrates GA and the SAA to explore the population through crossover and mutation, using temperature parameters to improve global search ability and avoid local optima. The inner computation then performs path optimization utilizing A∗ algorithm on the safety matrix constructed by BFS and the distance matrix constructed by Dijkstra's algorithm, balancing shortest path and minimum risk to minimize the path cost in the objective function. The safety and distance matrices constructed based on the grid map, provide the foundation for calculating the minimum path cost.

Experimental results demonstrate that in a simulated urban underground public environment, GSA-A∗ provides an intelligent evacuation path planning solution by comprehensively considering both path length and safety levels. In various scenarios, GSA-A∗ demonstrates excellent path planning capabilities, which not only optimizes significantly in the initial stage, but also shows the same advantages of fast convergence and global optimal solutions in different scenarios. The results consistently outperform traditional algorithms such as G-A∗, SA-A∗ and ACO-A∗. Although the recommended evacuation paths may have longer distances, they are safer. As a result, this framework effectively improves evacuation efficiency during emergencies.

The research findings of this paper not only provide effective solutions for fire safety in urban underground complexes but also offer a reference for the design and optimization of future intelligent evacuation systems. By applying intelligent path algorithms, the evacuation capability of underground public areas will be observably enhanced. Future research can further incorporate more variables from real environments, such as changes in crowd density and the spread speed of fire, to improve the adaptability and practicality of the algorithm. While this paper considers the shortest and safest evacuation paths in terms of distance and risk, future research directions should also consider how these paths can minimize total evacuation time. Additionally, our future work on the hardware framework for intelligent evacuation systems is still under development. Specific implementation can be achieved by integrating IoT technology, high-quality sensors, control devices, and emergency signage displays.

Acknowledgement: The authors would like to express our sincere gratitude and appreciation to the Editors and Reviewers for their valuable insights and continuous encouragement throughout the course of this research paper.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Guoqing Yang, Jialu Chen; data collection and analysis: Jialu Chen; draft manuscript preparation: Jialu Chen. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Q. H. Qian, "Present state, problems and development trends of urban underground space in China," *Tunnelling Undergr. Space Technol.*, vol. 55, pp. 280–289, 2016. doi: 10.1016/j.tust.2015.11.007.
2. L. A. Flores-Herrera *et al.*, "3D printed scaled setup for smoke transport analysis in a subterranean passenger platform," *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, vol. 35, no. 1, 2019. doi: 10.23967/j.rimni.2018.03.005.
3. P. Wang, P. B. Luh, S. C. Chang, and K. L. Marsh, "Efficient optimization of building emergency evacuation considering social bond of evacuees," in *Proc. 2009 IEEE 5th Conf. Autom. Sci. Eng. (CASE)*, Bangalore, India, Aug. 22–25, 2009, pp. 250–255.
4. M. Kobes, I. Helsloot, B. De Vries, J. G. Post, N. Oberijé and K. Groenewegen, "Way finding during fire evacuation; an analysis of unannounced fire drills in a hotel at night," *Build. Environ.*, vol. 45, no. 3, pp. 537–548, 2010. doi: 10.1016/j.buildenv.2009.07.004.
5. H. Aydt, M. H. Lees, S. J. Turner, and W. Cai, "Toward simulation-based egress optimization in smart buildings using symbiotic simulation," in *Pedestrian and Evacuation Dynamics 2012*, U. Weidmann, U. Kirsch, M. Schreckenberg, Eds., Cham, Switzerland: Springer International Publishing, 2014, pp. 987–999.
6. H. Ran, L. Sun, and X. Gao, "Influences of intelligent evacuation guidance system on crowd evacuation in building fire," *Autom. Constr.*, vol. 41, no. 7, pp. 78–82, 2014. doi: 10.1016/j.autcon.2013.10.022.
7. S. H. Wang, W. C. Wang, K. C. Wang, and S. Y. Shih, "Applying building information modeling to support fire safety management," *Autom. Constr.*, vol. 59, no. 4, pp. 158–167, 2015. doi: 10.1016/j.autcon.2015.02.001.
8. L. W. Chen and J. X. Liu, "Time-efficient indoor navigation and evacuation with fastest path planning based on Internet of Things technologies," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 5, pp. 3125–3135, 2019. doi: 10.1109/TSMC.2019.2918233.
9. A. M. Ibrahim, I. Venkat, K. G. Subramanian, A. T. Khader, and P. D. Wilde, "Intelligent evacuation management systems: A review," *ACM Trans. Intell. Syst. Technol.(TIST)*, vol. 7, no. 3, pp. 1–27, 2016. doi: 10.1145/2842630.
10. V. Q. Nguyen, H. T. Vu, V. H. Nguyen, and K. Kim, "A smart evacuation guidance system for large buildings," *Electronics*, vol. 11, no. 18, 2022, Art. no. 2938. doi: 10.3390/electronics11182938.
11. N. Al-Nabhan, N. Al-Aboody, and A. A. Al Islam, "A hybrid IoT-based approach for emergency evacuation," *Comput. Netw.*, vol. 155, no. 4, pp. 87–97, 2019. doi: 10.1016/j.comnet.2019.03.015.
12. M. A. Lopez-Carmona and A. P. Garcia, "CellEVAC: An adaptive guidance system for crowd evacuation through behavioral optimization," *Saf. Sci.*, vol. 139, 2021, Art. no. 105215. doi: 10.1016/j.ssci.2021.105215.
13. S. J. Yoo and S. H. Choi, "Indoor AR navigation and emergency evacuation system based on machine learning and IoT technologies," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 20853–20868, 2022. doi: 10.1109/JIOT.2022.3175677.
14. F. Kamoun, M. El Barachi, F. Belqasmi, and A. Hachani, "A smart dynamic crowd evacuation system for exhibition centers," *Procedia Comput. Sci.*, vol. 184, no. 5, pp. 218–225, 2021. doi: 10.1016/j.procs.2021.04.004.
15. M. Haghani, "Optimising crowd evacuations: Mathematical, architectural and behavioural approaches," *Saf. Sci.*, vol. 128, 2020, Art. no. 104745. doi: 10.1016/j.ssci.2020.104745.
16. K. Deng, Q. Zhang, H. Zhang, P. Xiao, and J. Chen, "Optimal emergency evacuation route planning model based on fire prediction data," *Mathematics*, vol. 10, no. 17, 2022, Art. no. 3146. doi: 10.3390/math10173146.

17. W. Y. Lin and P. H. Lin, "Intelligent generation of indoor topology (i-GIT) for human indoor pathfinding based on IFC models and 3D GIS technology," *Autom. Constr.*, vol. 94, no. 4, pp. 340–359, 2018. doi: 10.1016/j.autcon.2018.07.016.
18. M. Goodwin, O. C. Granmo, and J. Radianti, "Escape planning in realistic fire scenarios with Ant Colony Optimisation," *Appl. Intell.*, vol. 42, no. 1, pp. 24–35, 2015. doi: 10.1007/s10489-014-0538-9.
19. F. Li, Y. Zhang, Y. Ma, and H. Zhang, "Modelling multi-exit large-venue pedestrian evacuation with dual-strategy adaptive particle swarm optimization," *IEEE Access*, vol. 8, pp. 114554–114569, 2020. doi: 10.1109/ACCESS.2020.3003082.
20. N. Khamis, H. Selamat, F. S. Ismail, O. F. Lutfy, M. F. Haniff and I. N. A. M. Nordin, "Optimized exit door locations for a safer emergency evacuation using crowd evacuation model and artificial bee colony optimization," *Chaos, Solition Fract.*, vol. 131, 2020, Art. no. 109505. doi: 10.1016/j.chaos.2019.109505.
21. Y. Li, W. Cai, and A. A. Kana, "Design of level of service on facilities for crowd evacuation using genetic algorithm optimization," *Saf. Sci.*, vol. 120, no. 2, pp. 237–247, 2019. doi: 10.1016/j.ssci.2019.06.044.
22. Z. Yu, D. Li, S. Zhu, W. Luo, Y. Hu and L. Yuan, "Multisource multisink optimal evacuation routing with dynamic network changes: A geometric algebra approach," *Math. Methods Appl. Sci.*, vol. 41, no. 11, pp. 4179–4194, 2018. doi: 10.1002/mma.4465.
23. Z. Fang, X. Zong, Q. Li, Q. Li, and S. Xiong, "Hierarchical multi-objective evacuation routing in stadium using ant colony optimization approach," *J. Transp. Geogr.*, vol. 19, no. 3, pp. 443–451, 2011. doi: 10.1016/j.jtrangeo.2010.10.001.
24. L. Xu, K. Huang, J. Liu, D. Li, and Y. F. Chen, "Intelligent planning of fire evacuation routes using an improved ant colony optimization algorithm," *J. Build. Eng.*, vol. 61, no. 1, 2022, Art. no. 105208. doi: 10.1016/j.jobe.2022.105208.
25. T. S. Yusof, S. F. Toha, and H. M. Yusof, "Path planning for visually impaired people in an unfamiliar environment using particle swarm optimization," *Procedia Comput. Sci.*, vol. 76, pp. 80–86, 2015. doi: 10.1016/j.procs.2015.12.281.
26. E. M. Cepolina, "A methodology for defining building evacuation routes," *Civ. Eng. Environ. Syst.*, vol. 22, no. 1, pp. 29–47, 2005. doi: 10.1080/10286600500049946.
27. Q. Zhang, J. Ma, and Q. Liu, "Path planning based quadtree representation for mobile robot using hybrid-simulated annealing and ant colony optimization algorithm," in *Proc. 2012 IEEE 10th World Cong. Int. Contr. Autom. (WCICA)*, Beijing, China, Jul. 6–8, 2012, pp. 2537–2542.
28. R. S. Tavares, T. C. Martins, and M. D. S. G. Tsuzuki, "Simulated annealing with adaptive neighborhood: A case study in off-line robot path planning," *Expert. Syst. Appl.*, vol. 38, no. 4, pp. 2951–2965, 2011. doi: 10.1016/j.eswa.2010.08.084.
29. M. N. Zafar and J. C. Mohanta, "Methodology for path planning and optimization of mobile robots: A review," *Procedia Comput. Sci.*, vol. 133, no. 3, pp. 141–152, 2018. doi: 10.1016/j.procs.2018.07.018.
30. K. Sriniketh *et al.*, "Robot-aided human evacuation optimal path planning for fire drill in buildings," *J. Build. Eng.*, vol. 72, no. 2, 2023, Art. no. 106512. doi: 10.1016/j.jobe.2023.106512.
31. C. K. Xiong, D. F. Chen, D. Lu, Z. Zeng, and L. Lian, "Path planning of multiple autonomous marine vehicles for adaptive sampling using Voronoi-based ant colony optimization," *Robot. Auton. Syst.*, vol. 115, pp. 90–103, 2018. doi: 10.1016/j.robot.2019.02.002.
32. C. Shi, M. Zhong, X. Nong, L. He, J. Shi and G. Feng, "Modeling and safety strategy of passenger evacuation in a metro station in China," *Saf. Sci.*, vol. 50, no. 5, pp. 1319–1332, 2012. doi: 10.1016/j.ssci.2010.07.017.