ERP: Real-time Evacuation Route Planning Algorithm for Fire Scenes in Smart Buildings

Rong-Guei Tsai [a,*], Jixin Zhao [a], Yicong Yu [a], Chengzhu Lin [a], Xiaolan Chen [b], Lijing Zheng [c], Ziqi Lin [d] and Wenwen Zou [d]

[a] *New Engineering Industry College, Putian University, Fujian, 351100, China*
[b] *Zhangzhou College of Science Technology, Fujian, 363202, China*
[c] *Fuzhou software Technology Vocational College, Fujian, 350003, China*
[d] *Information Department, Xiamen Huatian International Vocation Institute, Fujian, 361102, China*

Abstract. Fire accidents pose a serious threat to human safety. Therefore, it is necessary to develop an efficient escape route planning algorithm to cope with the challenges posed by fire accidents and ensure that personnel can escape from dangerous areas safely and quickly. This study improves the A* algorithm and proposes a path-planning algorithm called Evacuation Route Planning (ERP). Based on the location of the fire source and the congestion level of the exit, ERP can dynamically adapt to changing fire conditions and provide the best escape route to ensure the safe and timely evacuation of personnel. Additionally, to avoid overcrowding, we have also introduced the balanced queuing waiting time method to keep the load of the escape exits relatively balanced. Through this method, we can reasonably allocate the waiting time of evacuees, thereby alleviating possible congestion during the evacuation process. Finally, we conducted comparative experiments with the original A* algorithm, the RRT algorithm, and the greedy algorithm. The experimental results show that ERP outperforms the other three path-planning algorithms in terms of performance indicators such as escape rate and escape path length. This further validates the effective application of ERP in fire environments.

Keywords: Fire incidents, escape route planning, real-time, dynamic adaptation, human safety

1. Introduction

In recent years, with the rapid development of urban modernization, the importance of fire prevention has gradually gained attention. Unfortunately, when a fire occurs, people often miss the golden opportunity to escape because they are unaware of the evacuation routes within the environment. Additionally, casualties among evacuees and the inability of rescue personnel to provide real-time assistance can occur due to collapsed or congested escape routes. However, with the flourishing development of networking technology and information technology, we can utilize the Internet of Things (IoT) and smartphones to access useful information during a fire, thereby assisting in early warning and evacuation efforts, reducing the death toll, and further improving the effectiveness of fire prevention to safeguard lives and property. From 2012 to 2021, there were a total of 1.324 million residential fires in China, resulting in 11,634 deaths, 6,738 injuries, and direct property losses of 7.77 billion yuan. Among them, there were 429 significant fires causing 1,579 deaths and 329 injuries, and 2 major fires causing 26 deaths [5]. The above statistics highlight the importance of not overlooking the casualties and property losses caused by fires.

However, to respond to fires, deploy various sensors in buildings, public places, and residential areas to monitor real-time indicators such as temperature, smoke, and gas concentration in the environment. By connecting the sensors and mobile devices to the internet using IoT technology, an intelligent fire monitoring and early warning system can be established. Once the system detects signs of a fire, it will immediately sound an alarm and transmit relevant information to the appropriate departments and residents, enabling them to quickly take evacuation measures. People can access real-time information and guidance regarding the fire through mobile applications, such as fire evacuation guides and emergency contact information. These mobile devices can also serve as emergency communication tools, allowing individuals to stay in touch with rescue personnel and loved ones through voice calls or social media platforms, seeking assistance, or providing their location information.

Selecting safe and effective evacuation routes in complex indoor environments with rapidly evolving fire situations is a crucial guarantee for guiding crowd evacuation and reducing casualties. Traditional static pathfinding methods struggle to consider the dynamic changes in fire situations, leading to blind and delayed evacuation decisions. To establish a fire evacuation system, this study proposes a real-time perception-based indoor fire evacuation path dynamic optimization method called Evacuation Route Planning (ERP), which utilizes the A* algorithm and incorporates real-time fire scene status, indoor building environment status, and other fire-related information to dynamically adjust evacuation routes. Furthermore, ERP can consider congestion at evacuation exits during a fire and generate real-time evacuation routes, dispersing evacuees effectively to improve evacuation efficiency and ensure personnel safety. On the other hand, it also provides important guidance for rescue personnel to enter the fire scene for rescue operations. Simulation tests have demonstrated that ERP can accurately and reliably adjust evacuation routes dynamically based on fire evolution, significantly improving the precision of emergency evacuation.

This study summarizes the following three contributions:

(1) The study proposes a real-time perception-based indoor fire evacuation path dynamic optimization method called Evacuation Route Planning (ERP). This method is based on the A* algorithm and utilizes real-time fire scene status and indoor building environment status, among other fire-related information, to dynamically adjust evacuation routes. Compared to traditional static pathfinding methods, ERP can more accurately adapt to the rapidly evolving fire situation, improving the accuracy and timeliness of evacuation decision-making.

(2) ERP not only considers the real-time development of the fire but also considers the congestion of evacuees. It improves evacuation efficiency and ensures personnel safety by dispersing evacuees. A method called balanced queue time is proposed in this research, which allocates exits based on the shortest queue time and equivalent evacuation rate. The shortest queue time focuses on the wait time of individual evacuees, achieving a balanced distribution and reducing waiting time. The equivalent evacuation rate focuses on the fairness of overall flow, achieving a relatively balanced load distribution, and avoiding excessive congestion at evacuation exits.

(3) We validate several commonly used and well-known path-planning algorithms. Compared to other common path planning algorithms, ERP demonstrates superiority in evacuation rate and dynamic flexibility. ERP plans escape routes more accurately and faster during emergencies than the greedy algorithm, the original A* algorithm [3, 7], and the rapidly-exploring random tree (RRT) algorithm [25–26]. Furthermore, ERP is applicable to scenarios involving multiple evacuees and multiple fire sources, demonstrating its adaptability and scalability.

The remaining sections of this paper are as follows: Section 2 is related work, analyzing the advantages and disadvantages of various path planning algorithms as well as application fields. Section 3 introduces the details of the data structure and ERP operating mechanism. The fourth section is a simulation experiment. We compare several common path-planning algorithms and analyze the successful escape rate. The last section is the conclusion and future work of this study.

2. Related Works

There are two types of path planning algorithms: static path planning and dynamic path planning. These types are based on the degree of environmental changes. These two types of algorithms employ different strategies to cope with environmental variations. Static Path Planning: Static path planning algorithms are suitable for situations where the environment changes minimally or remains fixed. Static path planning algorithms assume that the environment remains unchanged during the planning process, resulting in path planning being performed only once. Typically, static path planning is used in cases where the map remains static, or the environment is known. Dynamic path planning algorithms are designed for scenarios where the environment undergoes significant changes or is unknown. These algorithms can handle environmental changes and adjust and update the path in real-time based on these changes. In the case of fire, which is a dynamic environment, static path planning is insufficient. Therefore, path planning in fire scenarios requires the application of dynamic path planning algorithms.

2.1 Fire evacuation path planning

In [1], the authors propose an indoor navigation and evacuation framework with the objective of minimizing the travel time of mobile users. In normal circumstances, the framework estimates the density and speed of mobile users in each area and plans the shortest indoor navigation path. However, in emergency situations, it considers factors such as signal propagation duration, capacity, user distribution, and congestion to accurately estimate the evacuation time of evacuees. This approach helps alleviate congestion issues and ensures balanced evacuation loads. In [2], the authors introduce an individual emergency guidance framework that models the spatiotemporal mobility of all individuals to provide each person with a dedicated path for the shortest evacuation time. The algorithm considers factors such as corridor capacity and length, exit capacity, concurrent movement, and personnel distribution to minimize evacuation time. The proposed framework accurately estimates the evacuation time for each individual and evenly distributes the evacuation load among exits, achieving the most efficient load balancing.

Researchers propose a dynamic path optimization method for emergency evacuation in the event of a fire on a cruise ship, utilizing an improved A* algorithm [3]. The method aims to obtain real-time optimal evacuation paths by utilizing dynamic fire spread data. It combines the improved A* algorithm with a breadth-first search strategy to plan evacuation paths, ensuring the determination of the optimal path from the current location to a safe exit and reducing the likelihood of casualties during the evacuation process. In [7], the authors present a novel fire detection and path planning algorithm. They utilize a convolutional neural network-based object detection algorithm trained on real fire images to detect fire regions in the images. The authors propose an improved A* algorithm for path planning, incorporating weighted settings for different areas and utilizing a box blur method to steer the evacuation path away from obstacles. In [12], the authors discuss the performance analysis of the A* algorithm in the Indonesian Firefighting Robot Contest (KRPAI). They evaluate the algorithm based on time complexity and space complexity. According to experimental results, the A* algorithm requires an average of 4,270.72 time units and 14,192 bytes of memory to find a solution. In [20], a multi-objective, robust optimization-based path planning method is proposed for the evacuation scenario of subway passengers in the presence of multiple fires. The method aims to optimize passenger evacuation paths and improve evacuation efficiency. The authors define the objectives as minimizing the total evacuation time, minimizing the total risk, and minimizing the total congestion cost. The NSGA-II algorithm is used to solve the model and find the Pareto-optimal solutions under certain robustness control parameters. The overall optimization level of the paths is judged by how well they minimize the total cost function.

2.2 Community and building evacuation path planning

Building fires are common disasters in our daily lives, and successfully escaping from a fire depends on the design of evacuation routes and timing. A lack of evacuation equipment or poor design of emergency routes causes most of the fire damage. In [4], the authors focus on the challenging problem of evacuation path planning. Their research objective is to quickly generate evacuation paths for communities to evacuate victims to safe areas as soon as possible. The authors model the problem as a maximum flow problem and propose a binary search algorithm based on the maximum flow algorithm as the evacuation path planning algorithm for communities. In [11], the authors present a multi-objective mathematical model for path selection in emergency evacuations. The objectives are to minimize the total travel time on the paths and minimize the complexity of the paths. Considering the real-time impact of disaster spread, we model the travel speeds on each arc as continuous decreasing functions over time. We propose an ant colony optimization algorithm to solve this model.

In [13], the authors discuss the design and implementation of building network infrastructure. They propose an evacuation guidance system that takes into account the population distribution in evacuation directions. The system collects and provides expected information to rescue teams, guiding people towards safer areas or exits. In [17], the authors address the lack of dynamic adaptability in behavioral descriptions and path preprocessing mechanisms during fire emergencies. They propose a new algorithm based on geometric methods. They establish an adaptive dynamic path query table for creating dynamically generated maps and introduce a population density model to restrict the expected evacuation speed, enabling more accurate and efficient evacuation paths.

In [18], the authors consider parameters to determine the optimal path and use fuzzy algorithms to group the dataset and various parameters. They employ an improved Dijkstra algorithm to find the best path for firefighters to reach the fire scene. This algorithm considers multiple factors, such as road conditions, traffic conditions, and the priority of emergency situations, providing fast and reliable path planning solutions. The aim is to enhance the efficiency and accuracy of fire emergency response and ensure the ability to quickly identify the optimal evacuation or rescue routes in emergency situations. In [21], the authors propose an evacuation path planning method for high-rise building fires, incorporating real-time data, risk distribution calculations, and evacuation paths.

Researchers have developed various systems to address the challenges of fire evacuation and rescue, with the aim of improving the efficiency of disaster relief. In [6], the authors establish a spatial model for emergency evacuation in buildings and design an emergency evacuation platform based on mobile GIS. By utilizing location-based services (LBS) and displaying the spatial structure in mobile terminal scenarios, the platform virtually deduces the possible development trends of events to design evacuation routes within the building. In [14], the authors utilize radio frequency identification (RFID) technology and cloud computing to develop a hybrid building fire evacuation system (HBFES) on mobile phones. This system rapidly calculates reliable evacuation routes in the event of a building fire. In [15], the authors develop a Building Fire Rescue and Evacuation Management Information System (BFREMIS). Using a constructed network model, they analyze the evacuation situation in a university teaching building using EVACNET4 software. The analysis includes total evacuation time, floor clearance time, evacuation bottlenecks, and evacuation visual paths on the MAPGIS platform. In [19], the authors propose a multi-layer path planning system for indoor search and rescue operations. The system consists of five steps: map preprocessing, path planning, graph processing, route optimization, and post-processing. Using a graph-based approach, the system efficiently solves multi-objective path planning problems in multi-story buildings. Additionally, the system employs multiple query methods to maximize response time and maintain flexibility in changing environments.

2.3. Path planning for disaster response robots

With the advancement of technology, the use of robots or drones for monitoring, search, and rescue is one feasible solution. However, to conserve power or provide more efficient rescue operations, many studies focus on the path planning problem for robots or drones. In [8], the authors acquire forest maps through satellites and propose an optimal patrol path planning model for forests. They combine techniques such as graph theory and mathematical programming, with the Dijkstra algorithm as the core. In [9], the authors present a multi-agent collaboration model where agents can cooperate to accomplish firefighting tasks, enhancing their firefighting capabilities.

In [10], the authors address the problem of leader-following systems without considering accident situations and the environment. They propose a tracking trajectory planning method for mobile robot groups that is useful in accident scenarios. In [16], the authors propose a new algorithm for multi-task path re-planning of multiple drones in environments with unexpected events. Each drone plans its own initial, optimal, or suboptimal path using Voronoi diagrams and the Dijkstra algorithm. Each drone is assigned different tasks, and their paths are re-planned accordingly.

In [22], the authors propose a mechanism for forest fire detection and extinguishing using a sensor and actuator network. The authors use temperature sensors to detect fires and deploy robots to extinguish them. They introduce a robot path planning mechanism using fuzzy Q-learning (FQL)-based trajectory mechanisms to prolong the lifespan of sensors and robots in the Wireless Sensor and Actuator Network (WSAN).

3. System model

3.1. Problem description and definition

Figure 1 shows an aerial view of the "Jinrong Teaching Building" at Putian University. The area is divided into N grids, and people move on a grid-by-grid basis. The number of emergency exits is $M = 5$. Let $\mathbf{G}$ be the set of grids in the entire area, denoted as $\mathbf{G} = \{G_i | 1 \le i \le N, i \in \mathbb{R}\}$. Et represents the set of emergency exits, denoted as $\mathbf{Et} = \{Exit_j | 1 \le j \le M, j \in \mathbb{R}\}$.

Temperature and smoke sensors are assumed to be deployed inside the building. In this study, the hazard level is classified based on the fire hazard classification in the NFPA 704 standard [24]. When the temperature exceeds 80 degrees Celsius or the smoke concentration reaches 12,000 ppm, the hazard level is classified as dangerous. When the smoke concentration reaches or exceeds this threshold, the sensors transmit temperature or smoke concentration information to the

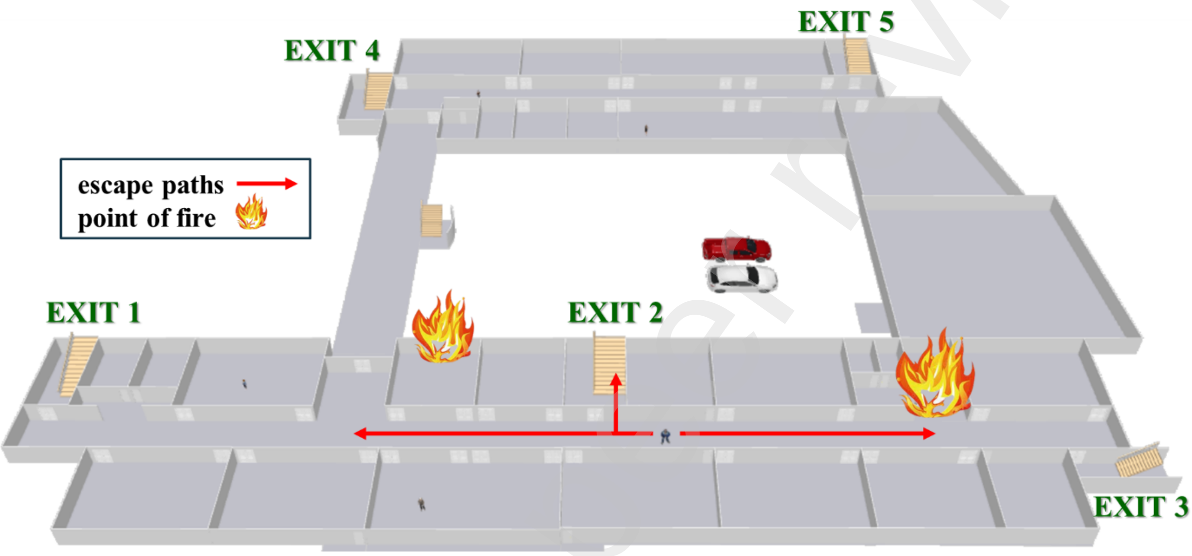


Fig. 1. Aerial view of Jinrong Teaching Building

backend server. The server utilizes the collected information from the sensors (including the location of the hazard), the total number of exits, and the congestion level of each exit to perform the ERP path planning algorithm. Finally, the server sends the results of the path planning to the users, who then evacuate the fire scene based on the instructions provided by their mobile devices.

In this study, we will design a path-planning algorithm to improve the evacuation rate of evacuees. The evacuation rate, denoted as R_E, represents the ratio of successfully evacuating individuals (NP_{suc}) to the total number of people (NP_{suc}) during a fire incident. The first goal of this study is to enhance the evacuation rate by planning appropriate paths, ensuring the safe evacuation of as many individuals as possible. Eq. (1) expresses the calculation of the evacuation rate.

$$R_E = \frac{NP_{suc}}{NP_{total}} \quad (1)$$

When the evacuation speed of evacuees remains constant, a shorter path length leads to faster evacuation time and higher efficiency. This means that evacuees require less time to escape the disaster scene. We define the path length as the distance from the starting point to the nearest safe exit. The NFPA 704 standard classifies areas based on certain criteria. In this case, when the temperature exceeds 80 degrees Celsius or the smoke concentration reaches 12,000 ppm, the area is considered a dangerous zone. Therefore, we can divide the grids in the entire area into two categories: *safe grids* (**SG**) and *dangerous grids* (**DG**). Eq. (2) shows the representation of the set of grids that constitute the dangerous zone, **D**.

$$\mathbf{D} = \bigcup_{k=1}^{|D|} (DG_k)\, , k \in \mathbb{R} \quad (2)$$

The constraint is that the path should not pass through dangerous zones, which means that none of the grids in the path should belong to the set **DG**. Let's assume a sequence of grids representing an evacuation path, denoted as $G = \{G_1, G_2, .., G_m\}$. If all the grids in the sequence belong to the set of safe grids, **SG**, it represents a safe evacuation path. Eq. (3) expresses this.

$$\mathbf{SG} - \bigcup_{k=1}^{m} (G_k) = \emptyset \quad \forall G_k \in \mathbf{SG},\ k = 1,2,3,..,m \tag{3}$$

Eq. (4) represents the time required for each individual to evacuate, with $d(G_i,G_j)$ representing the distance between grids i and j and c_{ij} being a binary variable equal to 1 if the grids are adjacent and 0 otherwise. ms represents the movement speed of the evacuee. Therefore, Eq. (4) calculates the time T_q required for a particular evacuee during the evacuation process.

$$T_q = \frac{\sum_{i=1}^{N+1} \sum_{j=1}^{N+1} d(G_i,G_j) c_{ij}}{ms} \quad \text{s.t. } c_{ij} = \{0,1\}, where\ i,j = 1,2,3,..,N \tag{4}$$

Indeed, a smaller time efficiency indicates that evacuees require less time to complete the evacuation process, resulting in faster evacuation. Therefore, the time efficiency, T_E (time efficiency), can be defined as the average time required to complete evacuation when a fire occurs. Reducing the time required for evacuation means that evacuees can quickly escape from the dangerous zone. Eq. (5) shows how to express this.

$$T_E = \frac{\sum_{q=1}^{NP_{total}} T_q}{NP_{total}} \tag{5}$$

Eq. (6) shows the study objective of the proposed evacuation path planning algorithm, ERP. Our aim is to design an efficient evacuation path planning algorithm that ensures evacuees can reach safety locations in the shortest possible time, maximizes the evacuation rate, and considers various constraints.

$$\min\left(\frac{1}{R_E}, T_E\right). \tag{6}$$

3.2. Data structure and system architecture

Figure 2 illustrates the system architecture and the process of information transmission. The system consists of three components: mobile devices, servers, and sensors. We can divide the information transmission process into four steps.

i. When a fire occurs, the sensors receive abnormal temperature or smoke concentration readings and transmit the location of the detected dangerous grid, which corresponds to the abnormal readings, to the server.
ii. The server broadcasts an **emergency package** to the users, notifying them of the presence of dangerous grids in the current area.
iii. Upon receiving the notification, users transmit their own location back to the server.
iv. The server uses the information collected from the sensors, such as the hazard location, number of exits, and congestion level of each exit, to calculate an evacuation path. Finally, the server sends an **escape instruction packet** to the users, enabling them to follow the instructions and evacuate the fire scene.

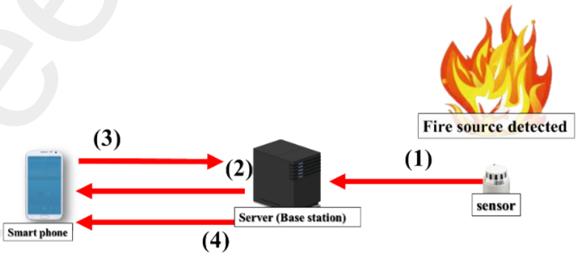


Fig. 2. System Architecture and Information Transmission Process

Figure 3 represents the format of the emergency package. The emergency package contains specific fields that provide information about the detected hazard.

Sensor ID (Sen_{ID}): This field records the ID of the sensor that generated the data packet. It identifies the specific sensor responsible for transmitting the emergency package.

Grid ID (G_{ID}): This field records the ID of the grid where the dangerous zone is located. It specifies the unique identifier of the grid associated with the detected hazard.

Temperature (A_t): This field records the temperature of the dangerous grid. It represents the temperature reading associated with the hazardous area.

Smoke Concentration (A_c): This field records the smoke concentration in the dangerous area. It indicates the level of smoke density or concentration associated with the detected hazard.

Sen_{ID}	G_{ID}	abnormal temperature, A_t	Abnormal smoke concentration, A_c

Fig. 3. The format of the emergency package

When the server receives an emergency package from the sensors, it broadcasts the emergency package to the users. The users then transmit their own location to the server, which is responsible for the path planning. The server calculates the evacuation path based on the user's location, the location of the fire point, the total number of exits, and the congestion level of the exits. The server records the evacuation path trajectory in the escape instruction packet and sends it to the users. Figure 4 represents the format of the escape instruction packet. The timestamp field indicates the time when the escape instruction packet is issued, enabling the determination of the information's freshness. User ID (us_{ID}) represents the ID of the user, and exit_{ID} represents the ID of the exit. The path trajectory field records the trajectory of the evacuation path, from the grid G_i, where the evacuee is located, to the exit G_g. The evacuation path trajectory, from the grid G_i, where the evacuee is located, to the exit G_g, is represented as $G = \{G_i, G_{i+1}, ..., G_{g-1}, G_g\}$.

However, to have a precise understanding of the situation at the fire scene, as soon as a fire occurs, the server periodically broadcasts the escape instruction packet to the users after receiving their location information. If a new fire point is detected, the server promptly notifies the users and updates them on the situation in the entire area.

Timestamp, tp	us_{ID}	exit_{ID}	path trajectory, G

Fig. 4. The format of the escape instruction packet

3.3. Proposed path planning algorithm: Evacuation Route Planning (ERP)

Once the sensors send the location to the server, the ERP procedure executes. The ERP procedure consists of two parts: path planning and evacuation diversion mechanisms. The ERP algorithm considers actual costs and heuristic estimated costs to prioritize exploring paths that are most likely to lead to exits, thereby reducing the search space and improving search efficiency. However, in the context of a fire scenario, we must also consider the distance between evacuees and the source of the fire to ensure their safety. Therefore, the key concerns are (1) the safety of the path and (2) the congestion level of the exits.

To determine the safety of a path, the presence of dangerous grids along the evacuation path is considered, as indicated in Eq. (3). If there are any dangerous grids along the path, it is considered unsafe. In Section 3.4, we propose the Balanced Queue Waiting Time method as an evacuation diversion mechanism to address the issue of exit congestion. This mechanism takes into account both the waiting time in the queue and the load on the exits. By multiplying a weight factor with the ratio of the equivalent number of people to the capacity, it balances individual waiting time and the overall fairness of the evacuation flow when selecting exits. The exit with the minimum value is considered the optimal exit, ensuring a balanced distribution of the evacuation task. By incorporating these considerations, the ERP program aims to determine safe evacuation paths by avoiding dangerous grids and implement an evacuation diversion mechanism that balances waiting time and exit congestion for a more efficient and fair evacuation process.

When finding the shortest path between two nodes in a graph or network, the A* algorithm is a commonly used heuristic search algorithm. It combines the breadth-first search of Dijkstra's algorithm with the greedy best-first search, guiding the search direction by estimating the cost from the starting node to the target node and improving search efficiency. However, the original A* algorithm neglects the location of the fire point, potentially leading to ineffective evacuation paths. In addition, existing A* algorithms do not provide specific methods for selecting evacuation exits. The ERP algorithm proposed in this study determines the evacuation exits, followed by path planning. During the path planning process, we continuously update the estimated cost (distance) between the evacuees and the nearest fire source to avoid dangerous grids. This ensures that the evacuation paths consider the location of the fire source and dynamically adjust to avoid potential hazards. By incorporating the specific considerations of fire source location and dynamically updating the estimated cost during path planning, the ERP algorithm aims to provide more effective and safe evacuation routes for evacuees.

After receiving an emergency packet notification from the server, the user sends their own location to the server, and the server begins the path planning task for the evacuee. The ERP algorithm consists of two steps: In Section 3.4, we explain the process of determining the evacuation exit. The path-planning process begins once we determine the evacuation exit. To illustrate the path planning process of ERP, we take Figure 5 as an example. Let's assume that the exit is located at grid position (7, 11) in the bottom right corner of the figure, and the fire source is at grid position (2, 8). There are eight possible directions for evacuation (up, down, left, right, and diagonals). In this study, we utilize two data structures: the open list and the closed list. We use the open list to store the grids that we have yet to explore. The algorithm has discovered these grids but has not fully explored them. The algorithm selects the next grid to explore from the open list. The closed list stores the grids that have been fully explored, meaning it has found their optimal paths and considered all possible neighboring grids. Once added to the closed list, a grid will not undergo further exploration or expansion.

First, initialize the algorithm by setting the starting grid as the current grid G_i (current position), the target grid G_t (evacuation exit), and the dangerous grid G_f.

Step 1: Add the current grid to the open list **FS**, as shown in Eq. (7). If the current grid is the target grid, a path has been found, and the search ends.

$$\mathbf{FS} = \{G_i\},\ \mathbf{ES} = \{\emptyset\},\ \forall G_i, G_t \in G \tag{7}$$

Step 2: Look at the grids G_j that are adjacent to the current grid G_i (reachable in one step) and add G_j to the open list **FS**. Record G_i as the parent grid of G_j, as shown in Eq. (8).

$$\mathbf{FS} = \{G_j \mid dist(G_i, G_j) = 1, \forall G_j \notin \mathbf{ES}, \forall G_j \in (\tag{8}$$

G_i is removed from the open list **FS** and added to the closed list **ES**, as shown in Eq. (9), because it has already been explored.

$$\begin{aligned} \mathbf{FS} &- \{G_i\}, \forall G_i \in G \\ \mathbf{ES} &= \{G_i\}, \forall G_i \in G \end{aligned} \tag{9}$$

Step 3: Take out the grid with the minimum cost from the open list **FS** and set it as the current grid G_i, as shown in Eq. (10). After setting the new current grid G_i, return to Step 1. If the open list **FS** is empty and the target grid has not been found, there is no feasible path.

$$G_i = \min(\mathrm{cost}(\mathbf{FS})) \tag{10}$$

Eq. (11) defines the ERP cost function $C(n)$, where $AC(n)$ represents the actual cost and $EC(n)$ represents the estimated cost. The estimated cost includes two parts: the estimated cost from the grid where the evacuee is currently located to the exit grid, and the estimated cost from the grid where the evacuee is currently located to the nearest dangerous grid.

$$C(n) = AC(n) + EC(n),\ \text{where } EC(n) = dist(G_i, G_t) + \min(dist(G_i, G_f)),\ 1 \le f \le k \tag{11}$$

Eq. (12) calculates both the distance between the current grid of the evacuee and the exit, as well as the minimum distance between the current grid of the evacuee and the dangerous grid (nearest), using the Manhattan distance.

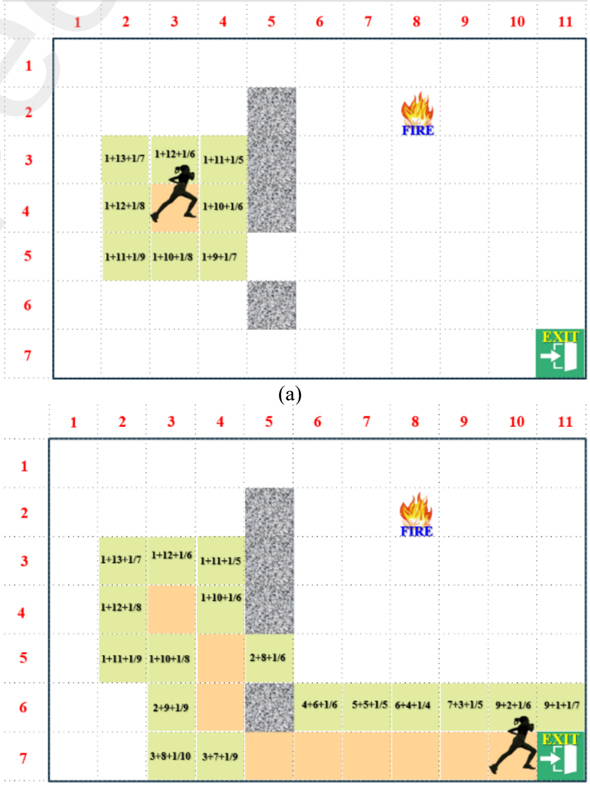
$$dist(G_i, G_j) = |x_i - x_j| + |y_i - y_j|,\text{where } 1 \le i, j \le N \tag{12}$$

In Eq. (13), the position of G_i is denoted as (x_i, y_i), and the coordinates of G_j are represented by (x_j, y_j). The equation represents the computation for selecting the current grid. We search for the grid with the minimum cost in the open list and choose it as the current grid for the next iteration.

$$\min(C(n)) \tag{13}$$

The process of adding the parent grid to the set **P** occurs every time we find it by backtracking through the parent grids (Eq. 14). Starting from the target grid, we trace back through the parent grids until we reach the starting grid to find the shortest path, **P**.

$$\mathbf{P} = \mathbf{P} \cup \{G_i\} \tag{14}$$



(a)
(b)

Fig. 5. (a) The state of the first round; (b) The state of the evacuee reaching the exit.

In Figure 5(a), the evacuee starts from grid (4,3) and calculates the neighboring grids. The evacuee finds that grid (5,4) has the minimum cost and chooses it as the next move in the next round. In figure 5(b), it represents the state where the evacuee reaches the target grid. Algorithm Is the pseudocode used by the ERP algorithm to find the shortest path. It takes inputs such as the current grid G_i, the target grid G_t, the set of dangerous grids **DG**, and the grid set **G**. The output is the evacuation path (**P**). Algorithm II uses a subroutine pseudocode to backtrack to the current grid and determine the evacuation path P when the current grid equals the target grid.

Algorithm I. routing planning

```
Input: Starting grid Gi, target grid Gt , dangerous grids set DG, grid set G
Output: Evacuation path trajectory P
FS = {Gi};
ES = {};
P = {};
While FS is not empty
        Gi = min(cost(FS));
    If Gi== Gt
        P = reverse_trace(Gi);
    End If
    For each Gj in neighbors(Gi) and Gj not in ES:
        FS = FS ∪ {Gj};
        parent(Gj) = Gi;
    End For
    FS = FS - {Gi};
    ES = ES ∪ {Gi};
End while
```

Algorithm II. reverse_trace function

```
Subprocess reverse_trace
Function reverse_trace(Gi)
P = {Gi};
While Gi != Gi
Gi = parent(Gi)
        P = P∪ {Gi};
End while
return P;
End function reverse_trace
```

3.4. Proposed evacuation diversion mechanism: balanced queuing wait time strategy

ERP considers the congestion level of evacuation exits to divert evacuees. The goal is to prevent all evacuees from rushing towards the same exit, which can lead to congestion. Therefore, during the path planning process, the server continuously updates the status of each exit and keeps track of the number of evacuees at each exit. This study proposes the allocation of exits based on the ***shortest queue time*** and the ***equivalent evacuation rate***. The Shortest Queue Time focuses on the individual evacuee's waiting time, aiming to achieve a balanced distribution and reduce waiting time, thereby improving individual satisfaction. The Equivalent Evacuation Rate focuses on the fairness of overall diversion, achieving a relatively balanced load on evacuation exits through equal distribution of people and diversion ratios, thus avoiding excessive congestion.

In the Shortest Queue Time method, we observe the current queue wait time at each exit and choose the exit with the shortest wait time as the evacuation exit for the evacuees. By observing the wait time of each evacuee in the queue, we allocate them to the exit with the shortest wait time, aiming to achieve a balanced distribution of tasks. The queue wait time (TQW) can be defined as the maximum evacuation time minus the individual evacuation time, as shown in Eq. (15).

$$TQW = T_{ML} - T_{IE} \tag{15}$$

Where T_{IE} represents the individual evacuation time. Each person's evacuation time is recorded from the moment they receive the evacuation instruction from the server until they successfully escape the building. The longest evacuation time, T_{ML}, is determined by identifying the person with the longest evacuation time among all evacuees. This is thought to be the longest evacuation time. For example, in the event of a fire, let's say person A takes 2 minutes to evacuate, and person B takes 3 minutes to evacuate. The longest evacuation time would be 3 minutes (determined by B's evacuation time). In this case, the queue wait time for person A would be the longest evacuation time minus A's evacuation time, which is 3 minutes minus 2 minutes = 1 minute.

To achieve a relatively balanced load on evacuation exits and avoid excessive congestion, this study proposes the ***Balanced Queuing Wait Time strategy***. It takes into account both the queue wait time and the load on the exits. By multiplying a weight factor with the ratio of equivalent evacuation rate to capacity, it balances individual wait times and overall diversion fairness when selecting exits. The exit with the minimum value is considered the optimal exit for achieving a balanced distribution of tasks, as shown in Eq. (16). By observing the queue wait time and equivalent evacuation rate at each exit, the Balanced Queuing Wait Time strategy dynamically adjusts exit allocation during the evacuation process to minimize wait times and ensure a balanced load on the exits.

$$\operatorname{argmin}\left(T\,QW_i + \frac{\omega \times EP_i}{ec_i}\right) \tag{16}$$
$$\text{, where } 1 \le i \le n,\ 0 \le \omega \le 1$$

Where $T\,QW_i$ represents the queue wait time at exit i. EP_i represents the effective population at exit i, and ec_i represents the maximum capacity of each exit i. ω is a weight factor. Eq. (17) shows how to determine the calculation of the effective population.

$$EP_i = NP_{total} \times \left(\frac{ec_i}{\sum_{i=1}^{n} ec_i}\right) \quad (17)$$
$$, \text{where } 1 \leq i \leq n$$

To illustrate the calculation method of the balanced queue waiting time, let's consider a scenario where a building is on fire and there are three available evacuation exits. The total population in the building is 300 people, and the maximum capacities of the three exits are 100, 150, and 200 people, respectively. We will calculate the effective population for each exit (EP_1, EP_2, and EP_3) and then utilize the balanced queue waiting time method to select the optimal exit.

(1) Calculate the effective population of each exit.

The effective population of Exit 1:

$$EP_1 = 300 \times \left(\frac{100}{(100 + 150 + 200)}\right) \cong 22.22$$

The effective population of Exit 2:

$$EP_2 = 300 \times \left(\frac{150}{(100 + 150 + 200)}\right) \cong 50$$

The effective population of Exit 3:

$$EP_3 = 300 \times \left(\frac{200}{(100 + 150 + 200)}\right) \cong 88.89$$

Now, we can use the formula of the balanced queue waiting time method to select the optimal exit. Let's assume the queue wait times for Exits 1-3 are 5 minutes, 3 minutes, and 4 minutes, respectively. The weight factor (ω) is 0.5. According to equation (16), we calculate the balanced queuing wait time for each exit.

(2) Calculate the balanced queuing wait time for each exit.

The balanced queuing wait time for Exit 1 (minutes):

$$5 + \frac{0.5 \times 22.22}{100} \cong 5.11$$

The balanced queuing wait time for Exit 2 (minutes):

$$3 + \frac{0.5 \times 50}{150} \cong 3.17$$

The balanced queuing wait time for Exit 3 (minutes):

$$4 + \frac{0.5 \times 88.89}{200} \cong 4.22$$

According to the calculation results, we can see that Exit 2 has the smallest value, which is 3.17 minutes. Therefore, the Balanced Queuing Wait Time method selects Exit 2 as the optimal exit. This means that during the evacuation process, people should prioritize choosing Exit 2 to achieve a balanced distribution of tasks and reduce wait times. Assigning evacuees to different exits will cause the calculation results of the balanced queuing wait time to change. The Balanced Queuing Wait Time method allows for dynamic adjustment of exit allocation during the evacuation process to minimize wait times and ensure a balanced load on the exits. Algorithm III provides the pseudocode for the evacuation diversion mechanism.

Algorithm III. Escape diversion mechanism: balanced queuing wait time strategy

Input: the maximum capacities for evacuation exits, ec, with length n. total population in the building, NP_{total}. weight factor, ω.
Output: Optimal exit for evacuation, $BestExit$

```
function calculate_equivalent_population(NP_total, ec)
   EP = []; // input the effective population
   total_capacity = sum(ec); // the total capacity of all evacuation exits
   For i in range(n)
       EP_i = NP_total * (ec[i] / total_capacity);
       EP.append(EP_i);
   End for
   min_time = Infinity;
BestExit = -1;
For i in range(n)
TQW_i = T_ML - T_IE[i];
weighted_time = TQW_i + (ω * EP[i] / ec[i]);
If weighted_time < min_time:
min_time = weighted_time;
BestExit = i;
End if
    End for
return BestExit;
End calculate_equivalent_population  function
```

4. Results and discussion

4.1. Simulation Configuration Parameters

In this section, we compare the performance of the proposed ERP algorithm with the Greedy, A* [3], [7], and RRT [25]-[26] algorithms. The greedy algorithm follows human behavior by evacuating towards the nearest exit. Our study utilizes an Intel Core i9-12900k processor, 8GB of DDR4 memory, and the Windows 11 operating system. We develop the simulations using Java, perform data analysis, and plot them using Matlab and Gunplot. The National Institute of Standards and Technology (NIST) in the United States developed the Fire Dynamics Simulator (FDS), an advanced indoor fire simulation software, which we use to generate fire data. The simulations in our study are conducted at a plane height of 1.5 meters above ground level, and the entire area is divided into N (1m by 1m) grid cells. Table 1 presents the experimental environment parameters and settings.

Table 1. Experimental Parameters

Parameter	Value
area size	100^2 grids
number of origins of fire	0-9
fire spread rate	1 grid/second
rate of evacuation movements	2-11 grid/second
number of evacuees	100
number of evacuations exits	2-11
capacity of evacuations exits	20
number of obstacles	20
size of obstacles	6 grids

To evaluate the performance of the algorithm, this study compares it using three metrics: **evacuation rate**, **path length**, and **time efficiency**. Since a fire incident occurs in a dynamic environment, the evacuation rate indicates the adaptability and robustness of the algorithm in such conditions.

(1) The evacuation rate (R_E) is defined as the proportion of successfully evacuated individuals (NP_{suc}) to the total population (NP_{total}) when a fire occurs. The primary goal of this study is to improve the evacuation rate by planning appropriate paths to ensure the safe evacuation of as many people as possible. Eq. (18) shows the calculation of the evacuation rate.

$$R_E = \frac{NP_{suc}}{NP_{total}} \tag{18}$$

(2) Path Length: The path length (PL) refers to the distance required to reach a safe exit from the starting point. A shorter path length implies faster evacuation time and higher efficiency. We define the path length as the sum of the path lengths traversed by successfully evacuated individuals divided by the total number of successfully evacuated individuals. Eq. (19) shows how to calculate the path length. Here, pl_i represents the path length of evacuee i.

$$PL = \frac{\sum_{i=1}^{NP_{suc}} pl_i}{NP_{suc}} \tag{19}$$

(3) Time efficiency (T_E): T_E refers to the average time required to complete the evacuation when a fire occurs. Reducing the evacuation time means that evacuees can quickly escape from the hazardous area. Eq. (20) shows the calculation of time efficiency.

$$T_E = \frac{\sum_{q=1}^{NP_{total}} T_q}{NP_{total}} \tag{20}$$

4.2. The influence of the number of origins of fire

The purpose of this set of experiments is to verify the influence of the number of origins of fire on four algorithms. To simulate different scenarios, we randomly generated 20 obstacles, each with a size of 6 grids. We gradually increased the number of origins of fire from 0 to 9, while keeping the fire spread rate fixed at 1 grid per second, the number of evacuation exits fixed at 6 with a capacity of 20 people each, and the movement speed of evacuees fixed at 2 grids per second. The initial positions of the evacuees were also randomly generated.

According to the experimental results shown in Figure 6, as the number of origins of fire increases, the evacuation rate gradually decreases. When the number of origins of fire reaches 9, the evacuation rate of the ERP algorithm is approximately 51%, while the greedy algorithm is around 21%. Therefore, the ERP algorithm achieves a higher evacuation rate of about 30% compared to the Greedy algorithm. This experiment confirms that having a planned evacuation route helps improve the evacuation rate.

Figure 7 displays the path lengths required by evacuees during the evacuation process. As the number of origins of fire increases, the path length gradually decreases. This is because the decrease in evacuation rate leads to a calculation of path lengths only for successfully evacuated individuals. The path lengths of the ERP and A* algorithms are similar. The RRT algorithm generates many unnecessary paths due to its random movement probability, resulting in longer path lengths. The shorter path length of the Greedy algorithm is due to considering only the path lengths of successfully evacuated individuals. The experimental results confirm that the number of origins of fire has a minor impact on path length.

Based on the experimental results shown in Figure 8, when the number of origins of fire is 9, the time efficiency of the ERP and A* algorithms is similar, with values of approximately 9.64 and 10.57, respectively. The RRT algorithm has the lowest time efficiency, approximately 17.57, indicating a longer evacuation time. The Greedy algorithm exhibits the best time efficiency, approximately 7.72. This experiment confirms that the number of origins of fire has a minor impact on time efficiency.

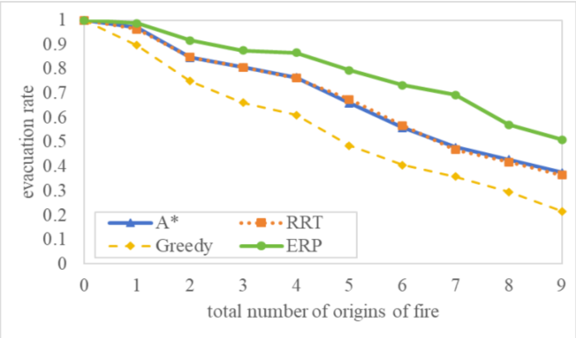


Fig. 6. influence of the number of origins of fire on the evacuation rate

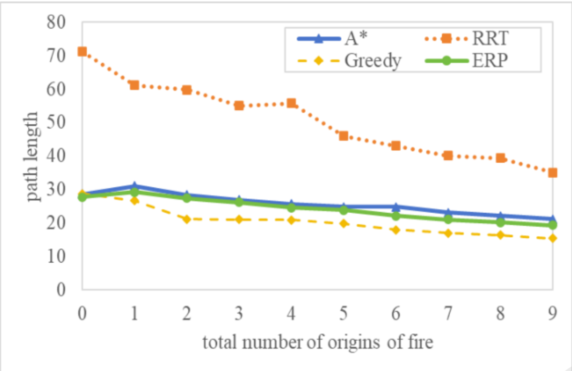


Fig. 7. influence of the number of origins of fire on the path length

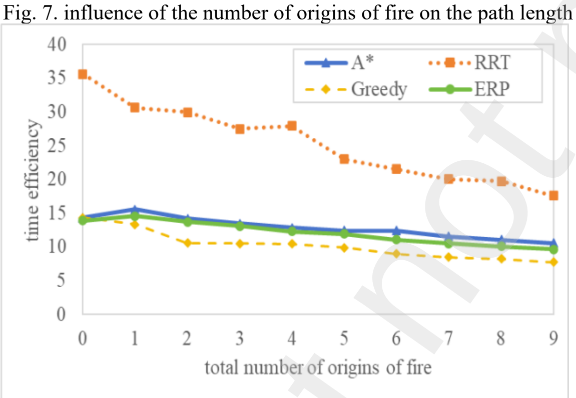


Fig. 8. influence of the number of origins of fire on the time efficiency

4.3. The influence of the rate of evacuation movement

This set of experiments aims to verify the impact of evacuees' movement speed on four algorithms. To simulate different scenarios, we randomly generated 20 obstacles, each with a size of 6 grids. We fixed the number of origins of fire at 3 and set the fire spread rate at 1 grid per second. There were six evacuation exits, with a capacity of 20 people each. Evacuees moved at speeds ranging from 2 to 11 grids per second, with their initial positions randomly generated.

According to the experimental results shown in Figure 9, as the movement speed of evacuees increases, the evacuation rate gradually rises. When the movement speed reaches 5 grids per second, the evacuation rate of the ERP algorithm is 96%. The RRT and A* algorithms achieve rates of 90% and 91%, respectively, while the Greedy algorithm reaches around 68%. Therefore, the ERP algorithm achieves a higher evacuation rate of 28% compared to the Greedy algorithm. This experiment confirms that having a planned route helps improve the evacuation rate.

Figure 10 displays the path lengths required by evacuees during the evacuation process. As the movement speed of evacuees increases, the path length gradually decreases. The path lengths of the ERP, A*, and Greedy algorithms are similar. The RRT algorithm generates many unnecessary paths due to its random movement probability. The experiment confirms that increasing the movement speed of evacuees does not necessarily shorten the path length.

Based on the experimental results shown in Figure 11, when the movement speed of evacuees reaches 11 grids per second, the ERP, A* and Greedy algorithms have similar time efficiency values of approximately 1.86 and 1.95, respectively. The RRT algorithm has the lowest time efficiency, approximately 3.64, indicating the longest evacuation time. This experiment confirms that increasing the movement speed of evacuees helps reduce the time required for evacuation.

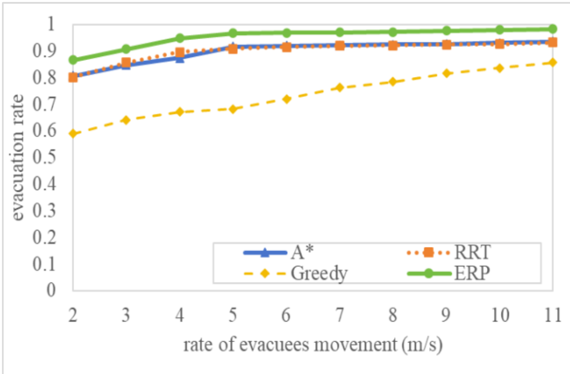


Fig. 9. influence of the rate of evacuation movement on the evacuation rate

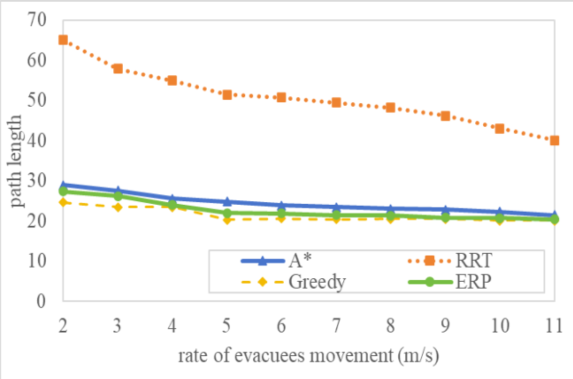


Fig. 10. influence of the rate of evacuation movement on the path length

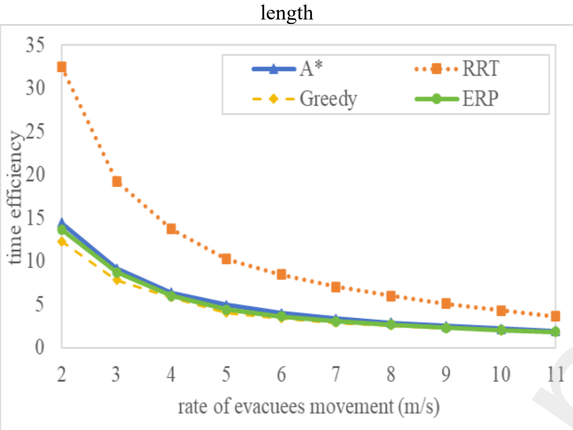


Fig. 11. influence of the rate of evacuation movement on the time efficiency

4.4. The influence the total number of the evacuation exits

This set of experiments aims to verify the impact of the total number of evacuation exits on four algorithms. We randomly generated 20 obstacles, each with a size of 6 grids, and gradually increased the number of fire points (fixed at 3) and the fire spread rate (fixed at 1 grid per second). The number of evacuation exits ranged from 2 to 11, with a capacity of 20 people each. Evacuees moved at a fixed speed of 2 grids per second, with their initial positions randomly generated.

According to the experimental results shown in Figure 12, as the number of evacuation exits increases, the evacuation rate gradually rises. When there are two evacuation exits, the evacuation rate of the ERP algorithm is 78.5%. The RRT and A* algorithms achieve rates of 72.5% and 73.7%, respectively, while the Greedy algorithm reaches approximately 37.8%. Therefore, the ERP algorithm achieves a higher evacuation rate of 40.7% compared to the greedy algorithm. When there are 11 evacuation exits, the ERP algorithm achieves a high evacuation rate of 97.7%, while the RRT and A* algorithms reach rates of 92.9% and 92.7%, respectively. The greedy algorithm achieves a rate of 74.5%. In comparison, the ERP algorithm achieves a higher evacuation rate of 23.2% compared to the greedy algorithm. This experiment confirms that having a larger number of evacuation exits helps improve the evacuation rate.

Figure 13 displays the path lengths required by evacuees during the evacuation process. As the number of evacuation exits increases, the path length gradually decreases. The path lengths of the ERP, A*, and Greedy algorithms are similar. The RRT algorithm generates many unnecessary paths due to its random movement probability. The experiment confirms that increasing the number of evacuation exits has a higher probability of finding a shorter path for evacuees.

Based on the experimental results shown in Figure 14, increasing the number of evacuation exits helps evacuees find exits closer to their locations, thus reducing the time required for evacuation. When there are two evacuation exits, the ERP, A*, and Greedy algorithms have similar time efficiency values of approximately 14.28, 14.52, and 14.47, respectively. The RRT algorithm has the lowest time efficiency, approximately 31.35. When there are 11 evacuation exits, the ERP, A*, and Greedy algorithms have similar time efficiency values of approximately 10.24, 10.7, and 10.47, respectively. The RRT algorithm has the lowest time efficiency, approximately 25.75, indicating the longest evacuation time. This experiment confirms that having a larger number of evacuation exits helps reduce the time required for evacuation.

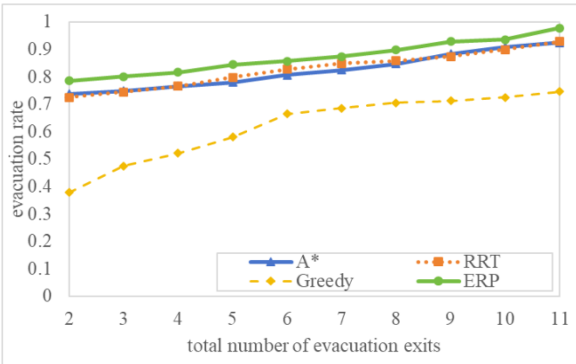


Fig. 12. The influence the total number of the evacuation exits on evacuation rate

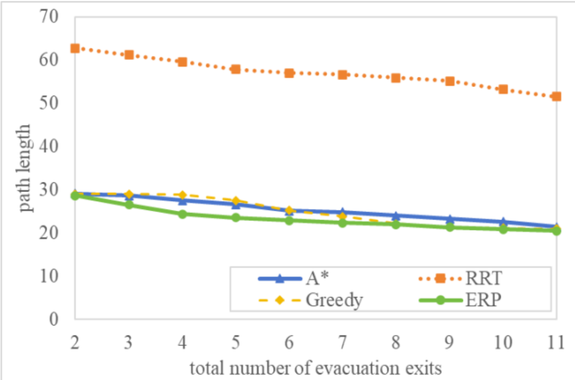


Fig. 13. The influence the total number of the evacuation exits on path length

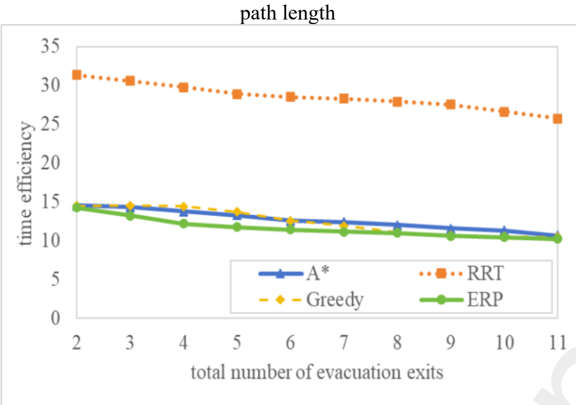


Fig. 14. The influence the total number of the evacuation exits on time efficiency

5. Conclusions

Traditional indoor emergency evacuation methods do not fully consider the impact of dynamic fire parameters, such as real-time fire propagation and smoke spreading on evacuation paths. Difficulties in ensuring the timeliness and reliability of emergency evacuation plans arise from the inability to adjust them in a timely manner based on the dynamic fire parameters. To address these issues, this study aims to propose a real-time fire evacuation path planning method called Evacuation Route Planning (ERP). By utilizing real-time fire and indoor building environment information, ERP dynamically adjusts evacuation paths to improve evacuation efficiency and personnel safety. Comparative experiments with other common path planning algorithms show that ERP can more accurately adapt to the rapidly changing fire situation and provide more accurate and timely information for evacuation decision-making. In addition, ERP considers the congestion of evacuees and improves evacuation efficiency and personnel safety by dispersing the evacuating crowd. The experimental results validate the adaptability and scalability of ERP, making it suitable for multi-person evacuation and multiple fire source scenarios. The ERP method proposed in this study offers a practical and effective solution for planning indoor fire evacuation paths. Future research can further explore the application of real-time perception-based evacuation path optimization methods in different scenarios and scales and validate them with more actual data and situations to further enhance the efficiency and safety of fire evacuation.

6. Acknowledgment

Rong-Guei Tsai was supported by the Putian City Science and Technology Project, China (Grant No. 2022GZ2001ptxy14), and the Science Foundation of the Fujian Province Project, China (Grant No. 2020J01924, Grant No. 2023J011015, Grant No. 2021J05243).

References

[1] L. -W. Chen and J. -X. Liu, Time-Efficient Indoor Navigation and Evacuation with Fastest Path Planning Based on Internet of Things Technologies, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **51**(5) (2021), pp. 3125-3135.

[2] L. -W. Chen, J. -H. Cheng and Y. -C. Tseng, Optimal Path Planning with Spatial-Temporal Mobility Modeling for Individual-Based Emergency Guiding, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **45**(12) (2015), pp. 1491-1501.

[3] Deng, Kunxiang, et al. Optimal emergency evacuation route planning model based on fire prediction data, *Mathematics*, **10**(17) (2022), pp. 3146.

[4] Liu, Li, et al. Intelligent evacuation route planning algorithm based on maximum flow, *International Journal of Environmental Research and Public Health*, **19**(13) (2022), pp. 7865.

[5] National Commercial Fire and Safety Association, Analysis of the number of fire incidents, injuries, deaths, and supply-demand of fire extinguishers in China in 2021, [Online]. Available: https://www.chyxx.com/industry/1119474.html. Accessed: October 2022.

[6] J. Huixian and Z. Shaoping, *Navigation system design of fire disaster evacuation path in buildings based on mobile terminals*, Proceedings of 11th International Conference on Computer Science & Education (ICCSE), Nagoya, Japan, 2016, pp. 327-331.

[7] J. h. Yu, S. nyeong Heo, J.-s. Shin and H.-h. Lee, *Fire Area Detection based on Convolutional Neural Network and Improved A* Path Planning*, Proceedings of International Conference on Information and

Communication Technology Robotics (ICT-ROBOT), Busan, Korea (South), 2018, pp. 1-4.
[8] J. Ma, L. Hu, B. Pan, Z. Li, Y. Tian and C. Chen, *Analysis and Decision of Optimal Path of Forest Disaster Patrol Based on Beidou Navigation*, Proceedings of 39th Chinese Control Conference (CCC), Shenyang, China, 2020, pp. 5624-5629.
[9] H. Zhang, *Research of AI Fire Fighting Robot Based on Big Data and Group Intelligence Perception*, Proceedings of Chinese Automation Congress (CAC), Shanghai, China, 2020, pp. 1625-1627.
[10] H. Amano, K. Osuka and Y. Iwano, *Tracking trajectory planning for disaster response multiple mobile robots: numerical simulation study*, Proceedings of SICE 2003 Annual Conference (IEEE Cat. No.03TH8734), Fukui, Japan, 2003, pp. 2896-2901.
[11] Y. Yuan and D. Wang, *Multi-Objective Path Selection Model and Algorithm for Emergency Evacuation*, Proceedings of IEEE International Conference on Automation and Logistics, Jinan, China, 2007, pp. 340-344.
[12] A. A. Hidayatullah, A. N. Handayani and M. J. Fuady, *Performance analysis of A* algorithm to determine shortest path of fire fighting robot*, Proceedings of International Conference on Sustainable Information Engineering and Technology (SIET), Malang, Indonesia, 2017, pp. 53-56.
[13] J. Lee, M. Choi, I. Joe and Y. Choi, *Design and Implementation of Adaptive Path Finding Based on CPS for Evacuation Guidance System in Disaster Situation*, Proceedings of International Conference on Information Science & Applications (ICISA), Seoul, Korea (South), 2014, pp. 1-2.
[14] L. Chu and S. -J. Wu, *An Integrated Building Fire Evacuation System with RFID and Cloud Computing*, Proceedings of Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Dalian, China, 2011, pp. 17-20.
[15] X. Tao, M. Guozhu, L. Xin and Z. Lin, *Building fire rescue with evacuation management information system and its application*, Proceedings of 16th International Conference on Industrial Engineering and Engineering Management, Beijing, China, 2009, pp. 2085-2090.
[16] B. -b. Meng, X. Gao and Y. Wang, *Multi-mission Path Re-planning for Multiple Unmanned Aerial Vehicles Based on Unexpected Events*, Proceedings of International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 2009, pp. 423-426.
[17] X. Chen and Y. Li, *Geometry-Based Virtual Simulation for Fire Escape in Emergency Environment*, Proceedings of Fourth International Conference on Networking and Distributed Computing, Los Angeles, CA, USA, 2013, pp. 56-59.
[18] G. S. Rahayuda and N. P. Linda Santiari, *Fire Incident Emergency Response Plan using Hybrid Fuzzy Dijkstra*, Proceedings of 1st International Conference on Cybernetics and Intelligent System (ICORIS), Denpasar, Indonesia, 2019, pp. 40-44.
[19] J. Lee, H. Park, Y. Kim, C. G. Park and J. H. Lee, Multi-Level Indoor Path Planning and Clearance-Based Path Optimization for Search and Rescue Operations, *IEEE Access*, **11** (2023), pp. 40930-40943.
[20] X. Yang, R. Zhang, Y. Li and F. Pan, Passenger Evacuation Path Planning in Subway Station Under Multiple Fires Based on Multiobjective Robust Optimization, *IEEE Transactions on Intelligent Transportation Systems*, **23**(11) (2022), pp. 21915-21931.
[21] Z. Han, W. Weng, Q. Zhao, X. Ma, Q. Liu and Q. Huang, Investigation on an Integrated Evacuation Route Planning Method Based on Real-Time Data Acquisition for High-Rise Building Fire, *IEEE Transactions on Intelligent Transportation Systems*, **14**(2) (2013), pp. 782-795.
[22] F. H. Panahi, F. H. Panahi and T. Ohtsuki, An Intelligent Path Planning Mechanism for Firefighting in Wireless Sensor and Actor Networks, *IEEE Internet of Things Journal*, **10**(11) (2023), pp. 9646-9661.
[23] H. Jiang, Mobile Fire Evacuation System for Large Public Buildings Based on Artificial Intelligence and IoT, *IEEE Access*, **7** (2019), pp. 64101-64109.
[24] NFPA 704, SMOKE ALARM INFORMATION, [Online]. Available: https://www.nfpa.org/Education-and-Research/Home-Fire-Safety/Smoke-alarms. Accessed: October 2022.
[25] J. Zhu, S. Zhao and R. Zhao, *Path Planning for Autonomous Underwater Vehicle Based on Artificial Potential Field and Modified RRT*, Proceedings of International Conference on Computer, Control and Robotics (ICCCR), Shanghai, China, 2021, pp. 21-25.
[26] P. Ren, S. Chen and H. Fu, *Intelligent Path Planning and Obstacle Avoidance Algorithms for Autonomous Vehicles Based on Enhanced RRT Algorithm*, Proceedings of 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatre, India, 2021, pp. 1868-1871.