Planning of location and path for urban emergency rescue by an approach with hybridization of clustering and ant colony algorithm

Bing Yang\textsuperscript{a}, Lunwen Wu\textsuperscript{a}, Jian Xiong\textsuperscript{a,*}, Yuxin Zhang\textsuperscript{a}, Lidong Chen\textsuperscript{b,*}

\textsuperscript{a}School of Business Administration, Southwestern University of Finance and Economics, Chengdu 611130, China
\textsuperscript{b}Beijing Institute for Advanced Study, National University of Defense Technology, Beijing 100000, China

Abstract

In urban emergency rescue, planning of rescue stations and paths are two important actions. Setting up suitable rescue stations can improve the rescue efficiency of rescue teams and improve rescue responses. Optimal path planning can provide rescue teams with effective rescue routes. This paper addresses the problem of identification of rescue station locations and planning of rescue paths. Firstly, an urban emergency rescue scenario is proposed. In the proposed scenario, the priority of each evacuee is quantified as a weight value which is the main factor considered in the optimization objective. Then a complete urban emergency rescue scheme is proposed, including road network processing, calculation of road network weights, planning of rescue station locations and planning of rescue paths. In the rescue station location planning phase, based on the location of evacuees and the structure of the road network, this paper uses clustering to provide candidate rescue stations for subsequent path planning. In the rescue path planning phase, an improved ant colony optimization algorithm is developed to solve the problem. The proposed approach is termed planning algorithm with clustering and improved ant colony algorithm (PA-
C-IACO). The PA-C-IACO redefines the degree of heuristic and pheromone concentration increments for transfer between intersections in the ant colony algorithm, and incorporates a reward mechanism during the pheromone update process. Finally, this paper uses six datasets of different sizes for experimental analysis. The results show that the proposed PA-C-IACO has better solution quality relative to existing methods and exhibits good robustness and feasibility. 

*Keywords:* Urban Emergency Rescue; Planning of Location and Path; Clustering; Ant Colony Algorithm.

1. Introduction

Urban emergency rescue refers to the emergency response of rescue services within the city limits when a natural or man-made disaster strikes. Fig 1. gives the number of natural disaster events worldwide from 1980 to 2019 and the number of these natural disasters is on the rise. The increase in disasters in recent years has increased the need for urban emergency response management. Government organizations do not accomplish rapid and effective emergency response, evacuees will suffer from life threatening, psychological harm and even resentment towards the government.

![Figure 1: Natural disaster events worldwide](https://ssrn.com/abstract=4041695)
Urban emergency rescue is a complex and comprehensive task, in which the planning of rescue station locations and rescue paths should be addressed in an integrated manner rather than individually. Research in recent years has begun to combine these two problems and has proposed models\[1,2\] to address the planning of rescue station locations and rescue paths in natural disasters. However, they do not address the planning of rescue station locations and rescue paths as a whole. In this paper, this type of optimization problem is combined with urban emergency rescue scenarios to plan rescue stations and rescue paths simultaneously.

Firstly, we construct the city-wide road network structure from the road network latitude and longitude information, which mainly includes the extraction of intersections and the calculation of intersection reachability. Then, we combine the location of evacuees and road network structure information to quantify the priority of evacuees into weight values and calculate the weights of each road section. Then, we use the clustering approach and the acquired knowledge to complete the planning of rescue station locations and provide candidate rescue stations for the subsequent path planning. Finally, an improved ant colony optimization algorithm is used for the planning of the rescue paths.

The main difficulties of this work are as follows: 1. obtain the urban road network structure and intersection accessibility from the raw latitude and longitude information. 2. obtain the priority of evacuees and the specific location in the road network based on the information of evacuees. 3. calculate the road network weights, including the number of evacuees and the weight value for each road section. 4. the setting of rescue stations should provide support for the efficiency and responsiveness of rescue. 5. plan multiple rescue paths for each rescue station to perform multiple rescues and maximizing the total weight value of the rescue.

Specifically, the setting of candidate rescue stations provides a high-quality set of starting points for path planning and provides regional coverage for the whole city. Current emergency rescue studies\[3,4,5,6\] generally focus on the merit of routes and do not consider the priority of evacuee as the goal of model.
optimization. The proposed method converts the priority of evacuee into a weighting approach and uses it as the optimization objective of the model. In addition, we borrow the idea of ant colony algorithm and propose a planning algorithm applied to urban emergency rescue. PA-C-IACO is not only a good fit for solving the actual problem, but also provides a reference for the improvement of the ant colony algorithm.

This paper abstracts, simplifies, determines variables and parameters for the hybrid planning problem of rescue station setup and rescue path in urban emergency rescue, and applies to the actual scenario of emergency rescue to establish the relationship between variables and parameters. Finally, the computer is borrowed to solve the emergency rescue problem, and the resulting solution is explained, tested, and evaluated. This paper is instructive for the solution of practical problems. The main contributions of this paper are as follows.

(1) This paper proposes an optimization scenario that considers both the planning of rescue station locations and the planning of rescue paths, quantifies the evacuee priority into weights to serve as the solution objective. It also refines the urban emergency rescue into four parts: road network processing, calculation of road network weights, planning of rescue station locations and planning of rescue paths, and proposes a complete urban emergency rescue scheme.

(2) This paper proposes the use of clustering and an improved ant colony optimization algorithm to plan the location of rescue stations and rescue paths to improve rescue efficiency.

(3) This paper borrows the idea of ant colony algorithm and reconstructs the degree of heuristic and pheromone increment for intersection transfer. And a reward mechanism for the pheromone updating process is proposed. The reward mechanism can accelerate the convergence of the model and prevent it from falling into local optimum.

(4) In the simulation experiments, the proposed method is more suitable for
the actual scenario requirements and has optimal results on data sets of different sizes.

2. Literature Review

2.1. Planning of location

In order to provide timely and effective relief to the evacuees, rescue station location is the key in urban emergency rescue, and existing studies on rescue station location are generally modeled with the purpose of optimizing the path. Ng et al.\cite{7} proposed a hybrid bilevel model, in which the first layer aims to determine the location of rescue stations and the second layer calculates the optimal path for crowd evacuation. Li et al.\cite{1,8} extended the hybrid bilevel model to a stochastic environment and proposed a scenario-based rescue station location model. At the same time, considering different rescue criteria or purposes, scholars often develop multi-objective models to solve the problem of rescue station location. Jianming et al.\cite{9} first proposed a multi-objective decision system for rescue station location, and then developed a model. Zhao et al.\cite{10} clearly defined the problem of rescue station location as a multi-objective optimization problem, and proposed a decision support tool. The main contribution of the above work lies in the combination of multi-objective optimization algorithms and geographic information systems to solve the problem of rescue station location.

Under the premise of optimizing the path for rescue station location, the impact of different paths on rescue work should also be considered. When faced with an unreasonable rescue station setup, the workload of rescuers will increase, resulting in a negative rescue state\cite{11}. At the same time, when in a high-pressure situation, rescuers are at risk of increased casualties\cite{12}. Therefore, a reasonable and reliable rescue plan can reduce losses for emergency accidents\cite{13}. Feng et al.\cite{14} selects the location of rescue stations from the aspects of emergency rescue response capability and work efficiency of rescue teams, and proposes a multi-objective optimization model to balance the work-
load and timeliness of rescuers. Hadi et al. [15] proposed a model of rescue station location through a practical case study, which can coordinate rescue vehicles and rescuers to allow rescue activities to be carried out efficiently and effectively. From the perspective of rescue decision-making, we should consider both the workload of rescuers and the optimization task of the path to ensure the efficiency and safety of rescue work [16]. Although these studies are modeled with the purpose of optimizing paths, they have not been applied to the task of path planning.

2.2. Planning of path

In urban emergency rescue, optimal path planning can provide rescue routes for rescue teams effectively. Many scholars have conducted research on path planning, and optimization models have been developed for different scenarios with different objectives and constraints. Li et al. [17] aims at minimizing the total travel cost for route planning, and finds that the flexible selection of stations can further reduce the total travel cost. Davoodi et al. [18] proposed two multi-objective path planning models to find safe paths by minimizing energy consumption. At the same time, considering the changing characteristics of traffic conditions and human flow, scholars have used a dynamic network flow model in the path planning problem. Lim et al. [19] proposed a capacity-constrained network flow optimization method to determine the optimal evacuation path planning and increase the total number of evacuees in the short-term evacuation plan. Pillac et al. [20] proposed a conflict-based model, which decomposed the evacuation planning problem into a main problem and a sub-problem to solve the evacuation plan and evacuation path respectively. In addition, there has been a gradual increase in research related to the use of swarm intelligence algorithms to solve path planning problems. Liu et al. [3] proposed an improved artificial bee colony algorithm (IABC) based on grouping strategy and exit selection strategy, which reduces the exit bottleneck problem of crowd evacuation. Li et al. [4] proposed the multi-depot green vehicle routing problem with multiple objectives (MDGVRP), and applied an improved ant colony optimization
algorithm (IACO) to solve the problem. Based on an improved ant colony algorithm, Zhu et al. [21] proposed a robot navigation algorithm in a dynamic unknown environment to find a locally optimal path and make dynamic adjustments.

Although path planning provides rescue routes, the location of rescue stations in urban emergency rescue can affect rescue efficiency. Therefore, we should combine the two problems of rescue station location and path planning and optimize them as a whole.

### 2.3. Planning of location and path

The path planning and the selection of rescue stations during the rescue process are crucial for the implementation of urban emergency rescue. When choosing the location of rescue stations, the transportation needs of all people should be considered [1]. Also, the number and location of rescue stations should be concerned when planning rescue routes [22]. Gama et al. [23] proposed a multi-period shelter location-allocation model to reduce the distance for evacuees to the rescue station. Bayram et al. [22] assigned evacuees to the shortest and closest paths, thus minimizing the total evacuation time. While considering time, rescue plans should also take into account the risk to evacuees. Goerigk et al. [24] proposed a comprehensive optimization model, which uses genetic algorithm of NSGA-II type to generate high-quality solutions for shelter location and personnel transportation. Coutinho-Rodrigues et al. [25] applied the multi-objective method to path planning and rescue station location, and pointed out the influence of rescue station location on path planning. Bayram et al. [2] proposed an algorithm based on Benders decomposition to locate rescue stations and plan paths. Kilci et al. [26] proposed a linear programming model to select the location of the rescue station and assign the path of rescuers.

The above studies proposed some models to deal with the problems of shelter location and evacuation planning under natural disasters, but they did not optimize these two problems as a whole. Meanwhile, current emergency rescue studies generally focus on the objectives of minimizing evacuation time [22].
minimizing road network distance\cite{23}, and minimizing rescue cost\cite{4}, and they do not consider the priority of evacuee as the objective of model optimization. This paper proposes a planning algorithm using clustering and improved ant colony optimization algorithm (PA-C-IACO) to solve the urban emergency rescue problem. PA-C-IACO not only improves the ant colony algorithm, but also quantifies the priority of evacuee as an optimization goal for the first time, ensuring the rationality of rescue. We extended the problem of rescue station location and path planning to dynamic scenarios, taking into account the dynamics of evacuees’ location, road structure and other issues. In this paper, the evacuees are transported to the rescue station as optimally as possible, which improves the rescue efficiency and rescue security.

3. Problem Formulation

3.1. Problem description

When an emergency occurs in a city and the evacuees are distributed in various areas within the city, rescue vehicles need to be deployed to rescue the scattered evacuees in various locations. Figure 2 shows a scenario in which 3 rescue vehicles are deployed to rescue evacuees from 3 different locations.

![Figure 2: Schematic diagram of urban emergency rescue issues](https://ssrn.com/abstract=4041695)

When rescuing, the categories of evacuees need to be considered, such as the seriously ill and injured, elderly, women, children, young men, etc. Different categories of evacuees have different priorities for their rescue. In the case of
limited rescue resources (such as vehicles and supporting resources), multiple rescues are required. In each rescue process, the rescue vehicle starts from the starting position and carries evacuees in turn on the path it passes through until it reaches the capacity limit of the rescue vehicle and delivers them to the resettlement site. In order to effectively use rescue resources and improve rescue efficiency and effectiveness, the starting position and rescue path of each rescue vehicle needs to be determined in each rescue process. In other words, the rescue station location and the path of rescue vehicles need to be planned collaboratively.

3.2. Mathematical modeling

Known emergency rescue base information: latitude and longitude of key points of the road network, total number of evacuees, categories of evacuees, location of evacuees, road and bridge access status affected by dynamic factors, number of rescue stations, and number of rescue vehicles. To facilitate the description, the symbols used in the modeling process are summarized as shown in Table 1.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l )</td>
<td>total number of evacuees</td>
</tr>
<tr>
<td>( t )</td>
<td>total number of road lines</td>
</tr>
<tr>
<td>( R )</td>
<td>key points of road network</td>
</tr>
<tr>
<td>( L )</td>
<td>location of evacuees</td>
</tr>
<tr>
<td>( K )</td>
<td>number of rescue stations</td>
</tr>
<tr>
<td>( H )</td>
<td>number of rescues per rescue vehicle</td>
</tr>
<tr>
<td>( O_k )</td>
<td>rescue station ( k )</td>
</tr>
<tr>
<td>( R_h(O_k) )</td>
<td>the path of rescue vehicle ( k ) departing for the ( h\text{th} ) time with ( O_k ) as the starting point</td>
</tr>
<tr>
<td>( S^h_k )</td>
<td>the set of intersections contained in the rescue path ( R_h(O_k) )</td>
</tr>
<tr>
<td>( I^h_k )</td>
<td>the set of evacuees contained in the rescue path ( R_h(O_k) )</td>
</tr>
</tbody>
</table>
3.2.1. Road network data

The road network of the rescue area consists of \( t \) roadlines, \( R \) is the set of all key points in the road network, with \( R = \{(lat^1_1, lon^1_1), (lat^2_1, lon^2_1), \ldots, (lat^r_i, lon^r_i)\}; \ldots; (lat^1_t, lon^1_t), (lat^2_t, lon^2_t), \ldots, (lat^r_t, lon^r_t)\}\}. Where \((lat^j_i, lon^j_i), i = 1, 2, \ldots, t; j = 1, 2, \ldots, r_i\) is the longitude and latitude of the \( j^{th} \) key point on the \( i^{th} \) path. The number of key points of the \( i^{th} \) path is denoted as \( r_i \). The traffic road network can be denoted as \( G = (V, E) \), where \( V = 1, 2, \ldots, n \) is the set of intersections in the city, and \( E \) is the set of edges. Obviously, \( V \) is a true subset of \( R \).

3.2.2. Evacuees data

Evacuees consisted of five different categories, namely children, young adults, middle age, old age, and seriously illness. The corresponding weights (priorities) for each category of evacuees are shown in Table 2.

<table>
<thead>
<tr>
<th>category</th>
<th>weight</th>
<th>symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>children</td>
<td>3</td>
<td>( Q^c )</td>
</tr>
<tr>
<td>young adults</td>
<td>1</td>
<td>( Q^y )</td>
</tr>
<tr>
<td>middle age</td>
<td>2</td>
<td>( Q^m )</td>
</tr>
<tr>
<td>old age</td>
<td>4</td>
<td>( Q^o )</td>
</tr>
<tr>
<td>serious illness</td>
<td>5</td>
<td>( Q^s )</td>
</tr>
</tbody>
</table>

In addition, we denote the location information of evacuees as \( L = \{(lat_1, lon_1), (lat_2, lon_2), \ldots, (lat_l, lon_l)\}\}, and \( l = |L| \). It is worth noting that the number of rescue stations in the rescue scenario is \( K \). Each rescue station has only one rescue vehicle, and the maximum capacity of all rescue vehicles is \( D \). The maximum number of rescues at each rescue station is \( H \), that is, the number of departures and returns of each rescue vehicle is \( H \). Then the total number of rescues of all rescue vehicles to the city is \( K \cdot H \).
3.2.3. Assumptions

(1) The rescue station is used as the starting point of the rescue, the rescue vehicle can drive on the road network and must rescue the evacuee it encounters.

(2) The number of Evacuees reaches the maximum capacity of the vehicle and returns to the rescue station.

(3) Each evacuee is on the road network road.

(4) During the rescue process, evacuees maintain their current position.

3.2.4. Modeling

With limited rescue resources, whether the key individuals can be rescued in priority depends on the total weight of the rescued evacuees. The total weight of rescue is the sum of the weights of the rescue vehicles picking up a fixed number of evacuees $D$ in turn along the rescue path after starting from the rescue station, provided that the emergency rescue rules are followed.

The planning of location and path is translated into solving the maximization problem:

$$\max W = \sum_{i=1}^{K} \sum_{j=1}^{H} f(R_j(O_i)),$$  \hspace{1cm} (1)

where $W$ is the total weight gained by the rescue vehicle when rescuing evacuees in the rescue path, $K$ is the number of rescue stations, $H$ is the number of rescue departures from each rescue point and the number of evacuees per rescue is equal to the capacity $D$ of the rescue vehicle, $R_j(O_i)$ denotes the rescue path of rescue vehicle $i$ from rescue station $O_i$ for the $j^{th}$ departure, and $f$ is the cost function on $R_j(O_i)$. $R_h(O_k)$ refers specifically the rescue path of rescue vehicle from rescue station $O_k$ and is denoted as:

$$R_h(O_k) = S^h_k = \left\{ p_1^{h,k}, p_2^{h,k}, \ldots, p_{|S^h_k|}^{h,k} \right\},$$  \hspace{1cm} (2)

where $p_1 = O_k$, $S^h_k$ denotes the set of all intersections contained in the rescue path $R_h(O_k)$, and $S^h_k$ is any subset of the set $V$ of intersections, $|S^h_k|$ is the number of intersections contained in the set $S^h_k$. The intersections in $S^h_k$
form a complete path, and there may be one or more intersections in $S_h^k$ between evacuees, the set consisting of these evacuees is represented using $I_h^k$ as follows:

$$I_h^k = \{ e_{h,k}^1, e_{h,k}^2, \ldots, e_{h,k}^{D'} \} , \quad (3)$$

where $\{ e_{h,k}^1, e_{h,k}^2, \ldots, e_{h,k}^{D'} \}$ is the number of each evacuee in $I_h^k$ and $D'$ is the total number of evacuees that can be covered by the rescue path $R_h(O_k)$. Since the complete path necessarily contains a certain number of intersections, and the number of evacuees between intersections is not fixed, $|I_h^k| = D' \geq D$. Any case has $D' \geq D$, so the set of evacuees rescued by rescue vehicle $h$ should be $I_{h}^{k'} = \{ e_{h,k}^1, e_{h,k}^2, \ldots, e_{h,k}^{D'} \}$. Based on the above knowledge, the computation procedure of the cost function $f$ for computing the path weights is defined as:

$$f (R_h (O_k)) = f (S_h^k) = f (I_h^{k'}) = \sum_{d=1}^{D} w \left( e_{d}^{h,k} \right) , \quad (4)$$

where $w(*)$ is a priority (weight) conversion function. It is because each person in $I_{h}^{k'}$ has a different category and a conversion function is needed to quantify the priority. This conversion function $w(*)$ is calculated as follows:

$$w \left( e_{d}^{h,k} \right) = \begin{cases} 
3, & \text{if the category of } e_{d}^{h,k} \text{ is } Q^c \\
1, & \text{if the category of } e_{d}^{h,k} \text{ is } Q^y \\
2, & \text{if the category of } e_{d}^{h,k} \text{ is } Q^m \\
4, & \text{if the category of } e_{d}^{h,k} \text{ is } Q^o \\
5, & \text{if the category of } e_{d}^{h,k} \text{ is } Q^s 
\end{cases} \quad (5)$$

Ultimately, the objective function of the problem is:

$$\max W = \sum_{i=1}^{K} \sum_{j=1}^{H} f \left( S_i^j \right) = \sum_{i=1}^{K} \sum_{j=1}^{H} f \left( I_i^{j'} \right) = \sum_{i=1}^{K} \sum_{j=1}^{H} \sum_{d=1}^{D} w \left( e_{d}^{j,i} \right) \quad (6)$$

Next, define the variables:

$$X_{ijkh} = \begin{cases} 
1, & \text{if } R_h (O_k) \text{ includes moving from intersection } i \text{ to intersection } j \\
0, & \text{other} 
\end{cases} \quad (7)$$
\[
Y_{ij} = \begin{cases} 
2, & \text{everyone between intersection } i \text{ and } j \text{ is picked up} \\
1, & \text{evacuees between intersections } i \text{ and } j \text{ are partly picked up} \\
0, & \text{other}
\end{cases} \tag{8}
\]

The mathematical model of this emergency rescue problem can be expressed as:

\[
\max W = \sum_{k=1}^{K} \sum_{h=1}^{H} W_{k,h} \tag{9}
\]

\[
\begin{align*}
\sum_{k=1}^{K} \sum_{h=1}^{H} |I^h_k| & = D, \tag{10.a} \\
\sum_{i=1}^{n} \sum_{j=1}^{n} X_{ijkh} & \leq Y_{ijkh}, \forall h, \forall k, \tag{10.b} \\
\sum_{i=1}^{n} X_{ijh} & \geq 1, \forall h, \forall k, \tag{10.c} \\
X_{ijkh} \in \{0, 1\}, Y_{ij} \in \{0, 1, 2\}, i \in V, j \in V, n = |V|, \forall h, \forall k. \tag{10.d}
\end{align*}
\]

where constraint (10.a) is the vehicle capacity limit and the maximum number of evacuees D per pickup, constraint (10.b) ensures that the vehicle must rescue evacuees when it encounters them, and constraint (10.c) ensures that each person can only be rescued once.

4. Proposed method

Emergency rescue program planning mainly considers determining the location of rescue stations and planning rescue routes given the location of evacuees and road network information as well as rescue vehicle constraints. Figure 3 illustrates the general framework of the emergency response planning approach proposed in this paper. Firstly, the raw data of road network and evacuee locations are processed to obtain the road network structure information and the location information of evacuees in the road network. Secondly, according to the evacuee location information, the evacuation area is clustered and used as the main basis for the rescue point setting. Finally, the improved genetic algorithm is used to plan the rescue route for each rescue vehicle. The proposed planning
algorithm using clustering and improved ant colony optimization algorithm is referred as PA-C-IACO.

![Diagram of emergency response planning](image)

Figure 3: The general framework of emergency response planning

### 4.1. Road network processing

The road network processing needs to obtain the urban road network structure and reachability between all intersections from the raw latitude and longitude information. Figure 4 gives the processing flow of a road network with 5 roadlines and 6 intersections. The process is summarized as follows:

1. Extract the key points in all roadlines.
2. If 1 key point exists in different roadlines, then the key point is an intersection.
3. Sequentially traverse the intersections in each roadline, with the previous intersection having reachability to the next one. Following the above process, all the intersections and the reachability matrix between them can be calculated. In this process, the latitude and longitude of all intersections and the latitude and longitude of key points between all intersections are recorded.

In the following algorithm, the key points of each roadline have been ordered in a forward and backward order, and if the intersections in \( V \) are reachable, then both intersections must belong to the same roadline. Algorithm 1 first extracts all the intersections in the road network (Line 1 to 6), initializes the reachability matrix between intersections \( A \) (Line 7) and then calculates the
reachability between each intersection (Line 8 to 11), and finally obtains all intersections in the road network $V$ and reachability matrix $A$ (Line 12).

4.2. Calculation of road network weights

The location information of evacuees determines the weight and number of evacuees on each road section, and also directly influences the planning of rescues. In the traffic road network $G = (V, E)$, the weights between the vertices are obtained from the priority calculation of evacuees and are denoted as $W_{ij}(W_{ij} \geq 0; W_{ii} = \infty; i, j \in V)$. The number of evacuees between intersection $i$ and intersection $j$ is denoted as $N_{ij}(N_{ij} \geq 0; N_{ii} = \infty; i, j \in V)$. Obviously, $l = |L| = \sum_i \sum_j N_{ij}$. If $W_{ij} = 0$, it means that intersection $i$ and intersection $j$ are reachable and there are no evacuees on the roadway between them. If $W_{ij} > 0$, it means that intersection $i$ and intersection $j$ are reachable and there are evacuees on the roadway between them. If $W_{ij} = \infty$, it means that intersection $i$ and intersection $j$ are not reachable and rescue vehicles cannot reach intersection $j$ from intersection $i$. When $W_{ij} \geq 0$, then $N_{ij} \geq 0$. When $W_{ij} = \infty$, then $W_{ij} = \infty$.

The calculation of road network weights is mainly to match the location of evacuees with the traffic road network, and then calculate $W_{ij}$ and $N_{ij}$. The process is mainly as follows: 1. Determine the specific location of each
Algorithm 1 Road network processing

Input: key points of road network $R = \{(\text{lat}_1, \text{lon}_1), (\text{lat}_2, \text{lon}_2), \ldots, (\text{lat}_r, \text{lon}_r)\}$

Output: all intersections in the road network $V$, reachability matrix between intersections $A$

1: for $i = 1, 2, r$ do:
2:     for $j = 1, 2, r$ do:
3:         if $(R_i == R_j) \&\& (R_i$ and $R_j$ do not belong to the same roadline):
4:             $V$.append($R_i$):
5:     end for
6: end for
7: initialize $A = \text{zeros}(|V|, |V|)$  
   // The size of $A$ is $|V| \cdot |V|$, and the initial value is 0
8: for $i = 1, 2, |V| - 1$ do:
9:     if $V_i$ and $V_{i+1}$ belong to the same roadline:
10:        $A(i, i + 1) = A(i + 1, i) = 1$
11: end for
12: output all intersections $V$, reachability matrix between intersections $A$
evacuee in the road network by the location of evacuees. 2. Calculate the weight \( W_{ij} \) between intersections based on the specific location of evacuees. 3. Calculate the number of evacuees \( N_{ij} \) between intersections based on the specific location of evacuees. 4. Update the accessibility between intersections based on dynamic factors. Algorithm 2 first initializes the weights and the number of evacuees between intersections to 0 (Line 1), and then determines the reachability between intersections (Lines 2-12). When the intersection is not reachable, both \( W_{ij} \) and \( N_{ij} \) are set to \( \infty \) (Lines 4 to 5). When intersections are reachable, the number of evacuees and weights between intersections are calculated (Lines 6 to 10), where \( w(e_k) \) is the weight of person \( e_k \) (Equation (5), see Table II for details). The final output weight matrix \( W_{ij} \) and number matrix \( N_{ij} \) (Line 13).

4.3. Planning of rescue station locations by clustering

In order to ensure that all rescue operations are effectively carried out throughout the city, rescue stations need to be set up dynamically according to the location of the evacuees. Therefore, this paper proposes a clustering-based method for planning the location of rescue stations. For a given distribution location of evacuees, the weight \( W_{ij} \) between each intersection is first calculated. Then, the distance between intersections is calculated, and the urban road network is divided into \( K \) clusters, that is, the traffic road network is partitioned into \( K \) regions. Then, we extract the \( \theta \) intersections with the highest weight \( W_{ij} \) in each cluster, and use these \( \theta \) intersections as the candidate rescue stations of the current cluster. We note \( R^\theta_K \) as the candidate rescue stations in each cluster, where \( \theta \) is a constant. Finally, the optimal intersection is selected as the final rescue station by the path planning algorithm. Such a clustering process lets the intersections within a cluster be as tightly connected as possible, while letting the intersections and evacuees between the clusters all be as far apart as possible. If \( K \) clusters are denoted as: \( C_1, C_2, \ldots, C_K \), then our goal is to
Algorithm 2 Calculation of road network weights

**Input:** all intersections in the road network $V$, reachability matrix between intersections $A$, location of evacuees $L = \{(lat_1, lon_1), (lat_2, lon_2), \ldots, (lat_l, lon_l)\}$

**Output:** weight matrix $W_{ij}$ between intersections, number matrix $N_{ij}$ between intersections

1: initialize $W_{ij}, N_{ij} = \text{zeros}(|V|, |V|)$

2: for $i = 1, 2, |V|$ do:

3:     for $j = 1, 2, |V|$ do:

4:        if $A_{ij} = 0$: // intersection $i$ and $j$ are not reachable

5:            $W_{ij} = \infty, N_{ij} = \infty$

6:        else: // intersection $i$ and $j$ are reachable

7:            for $k = 1, 2, l$ do:

8:                if $e_k$ is located between intersection $i$ and $j$:

9:                    $N_{ij} = N_{ij} + 1$

10:                   $W_{ij} = W_{ij} + 1$

11:            end for

12:     end for

13: end for

14: output weight matrix $W_{ij}$ and number matrix $N_{ij}$
minimize the squared error $\mathcal{E}$:

$$
\mathcal{E} = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2_2
$$

where $\mu_i$ is the center-of-mass of cluster $C_i$, is expressed formally as:

$$
\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x
$$

The choice of $K$ value is closely related to the setting of the number of rescue stations, and the number of $K$ is set to be equal to the number of rescue stations in this paper. Algorithm 3 gives this clustering process. First, $K$ samples are randomly selected from $L$ as the initial $K$ centroids (Line 1). Then the cluster partition $C$ is initialized (Lines 3 to 5), the distance between each sample and the center-of-mass of each category is calculated (Line 7), and each sample is assigned to the corresponding category with the shortest center-of-mass distance (Lines 8 to 9). Finally, the new center of mass is recalculated for all sample points in each cluster (Lines 11 to 13), and the final result is output if all $K$ center of mass vectors remain unchanged (Lines 14 to 17). Finally, the new center-of-mass is recalculated based on all sample points in each cluster (Lines 11 to 13), and the final result is output if all $K$ center-of-mass vectors remain unchanged (Lines 14 to 17).

4.4. Planning of rescue paths by improved ant colony optimization algorithm

4.4.1. Framework of ACO

For the traditional ant colony algorithm, the main purpose is to prevent falling into local optimum by introducing pheromones, and it can speed up the convergence of the algorithm. In the pathfinding process, the pathfinding process of each ant follows the roulette wheel method.

$$
P_{ij}^k(t) = \begin{cases} 
\frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in J_k(i)} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta}, & j \in J_k(i) \\
0, & j \notin J_k(i)
\end{cases}
$$

19
Algorithm 3 Calculation of road network weights

**Input:** location of evacuees $L = \{x_1, x_2, x_3, \ldots, x_l\}$; number of clusters $K$; Maximum number of iterations $M$

**Output:** cluster partition $C = C_1, C_2, \ldots, C_K$

1: initialize $K$ centroid vectors: $\mu_1, \mu_2, \ldots, \mu_K$
2: for $n = 1, 2, M$ do:
3: for $t = 1, 2, K$ do:
4: initialize $C_t = \emptyset$
5: end for
6: for $i = 1, 2, l$ do:
7: Calculate the distance between the sample $x_i$ and each centroid vector $\mu_j (j = 1, 2, \ldots, K)$: $d_{ij} = ||x_i - \mu_j||_2^2$
8: mark $x_i$ as the category $\lambda_i$ corresponding to the smallest $d_{ij}$
9: update $C_{\lambda_i} = C_{\lambda_i} \cup \{x_i\}$
10: end for
11: for $j = 1, 2, K$ do:
12: recalculate new centroids: $\mu_j = \frac{1}{|C_j|} \sum_{x \in C_j} x$
13: end for
14: if all $K$ centroid vectors have not changed:
15: break
16: end for
17: output cluster partition $C = C_1, C_2, \ldots, C_K$
where \( i \) is the code of the city where the ant is currently located, \( j \) is the code of the city that the ant can reach, \( \tau(t) \) is the pheromone concentration from \( i \) to \( j \) at time \( t \), and \( \eta \) is the reciprocal of the distance from \( i \) to \( j \), indicating the visibility between the two points \( i \) and \( j \). \( J_k(i) \) is the set of locations that can be reached by the ant. In order to allow the ant colony to be more flexible, a pheromone factor \( \alpha \) and a heuristic factor \( \beta \) were also introduced to indicate the importance of the pheromone and the likelihood that the ants would refuse to explore new regions, respectively.

At the beginning of the algorithm, all positions contain pheromone concentration \( Q \). In each iteration, there are \( m \) ants searching. After each completed iteration, the pheromone is left behind by this optimal ant, but only \( \tau \times (1 - \rho) \) of pheromone is also retained at each location, and the operator equation is:

\[
\begin{align*}
\tau_{ij}(t + 1) &= \tau_{ij}(t) \times (1 - \rho) + \Delta \tau_{ij} \\
\Delta \tau_{ij} &= \sum_{k=1}^{m} \Delta \tau_{ij}(k)
\end{align*}
\]

where \( \rho(0 < \rho < 1) \) is the pheromone concentration volatilization factor, and \( (1 - \rho) \) denotes the pheromone concentration remaining in the path after volatilization. This process stops after the maximum number of iterations is reached.

### 4.4.2. Proposed improved ACO

Several scholars have made different improvements on the basis of ant colony algorithm for solving robot path planning[27], UAV path planning[28], delivery problem[29] and vehicle routing problem[30]. In this paper, the basic idea of ant colony algorithm is adopted to construct PA-C-IACO. In path planning, the probability of reaching the next intersection from the current intersection for each rescue vehicle follows Equation (13). We define \( p^k_{ij}(t) \) as the probability that at time \( t \) rescue vehicle \( k \) moves from intersection \( i \) to intersection \( j \). \( i \) is the number of the intersection where the rescue vehicle is currently located, \( j \) is the number of the intersection that the rescue vehicle can reach, and \( J_k(i) \) is the set of intersections that rescue vehicle \( k \) is allowed to choose next. In
addition, we redefine for the heuristic factor $\eta_{ij}(t)$:

$$
\eta_{ij}(t) = \frac{W_{ij} + 1}{P}
$$

(15)

where $W_{ij}$ is the weight value obtained by transferring from intersection $i$ to intersection $j$ and $P$ is a constant. In the above equation, $W_{ij} + 1$ is used to prevent the extreme case where $W_{ij}$ is zero and the probability of rescue vehicle transfer is zero. If there are no evacuees between intersection $i$ and intersection $j$, then $W_{ij}$ will be zero, but the rescue vehicle is allowed to pass through intersection $i$ to reach intersection $j$.

The setting of the number of rescue stations $K$ determines the computational cycle of the algorithm, and each computational cycle completes a specified number of iterations, and each iteration randomly selects an intersection from the set of candidate rescue stations $R^\theta_K$ as the starting point of rescue. In each iteration there are $m (m = H)$ ants searching, where $H$ is the number of rescues performed at each rescue point. During the iteration process, after the rescue vehicle moves from intersection $i$ to intersection $j$, both $W_{ij}$ and $N_{ij}$ are set to 0 (indicating that the evacuees are picked up) and the total number of evacuees and the total weight of the current rescue are recorded. Therefore, in the rescue process, the previous rescue affects the planning of the next rescue path.

In the pheromone update process, we incorporate a reward mechanism, i.e., changing the pheromone volatility process if a better solution is encountered during the iterative process. The pheromone update process is redefined as:

$$
\tau_{ij}(t + 1) = \begin{cases} 
\tau_{ij}(t) \ast (1 - \rho) \ast R + \Delta \tau_{ij}, & W_{\text{current}} > W_{\text{best}} \\
\tau_{ij}(t) \ast (1 - \rho) + \Delta \tau_{ij}, & W_{\text{current}} \leq W_{\text{best}} \end{cases}
$$

(16)

where $R$ is the reward coefficient given to the pheromone in the reward mechanism, which is triggered when the total weight $W_{\text{current}}$ of the optimal path under this iteration is greater than the optimal weight $W_{\text{best}}$. The addition of the reward mechanism can reduce the pheromone volatility on the optimal path, guide the ants to search for a better route, and accelerate the convergence.
speed. Unlike the traditional ant colony algorithm, the pheromone concentration increment $\Delta \tau_{ij}$ is derived by equation (17).

$$
\Delta \tau_{ij}(k) = \begin{cases} 
\frac{Q}{W_k}, & W_{ij} \in R_k \\
0, & \text{other}
\end{cases}
$$

(17)

where $W_k$ denotes the total weights available on the rescue path of rescue vehicle $k$, $R_k$ is the path taken by rescue vehicle $k$, and $Q$ is a constant. This pheromone update process is similar to the pheromone update method in [4], since they both aim to reduce the impact of inferior solutions on the next path planning. The optimal path can be found by continuously iterating over the above path search process.

The PA-C-IACO flow chart is shown in Figure 5. The algorithm first initializes the weights between all intersections, pheromones, number of evacuees, clustering results and the number of rescue stations $K$ to be set according to the user parameters, where the number of rescue stations $K$ is equal to the number of categories in the clusters. After completing the initial operation, the intersection with the top $\theta$ weight ranking under the current clustering category is extracted as an alternative rescue station. Then the ants start to act one by one and randomly select a starting point from the alternative rescue stations in each iteration, and each ant selects a route based on the pheromones and weights among intersections. When all the ants finish searching the route, the iteration is completed and the pheromone is updated. When all the ants finish searching the route, this iteration is completed and the pheromone is updated. Repeat the above process to get the location of all rescue stations and the corresponding rescue paths.

5. Experiment and result analysis

5.1. Dataset preparation

To verify the effectiveness of the algorithm in this paper, we randomly generated six scenarios in a city with the number of people 300, 500, 700, 900,
Start

Initialize the weights, pheromones, the number of evacuees between all intersections and the clustering results

Initialize the number of rescue station $K$

Extract the top-$\theta$ intersections in a cluster category as the starting point

Read the weight and the number of evacuees between all intersections

Initialize $m$ ants and randomly select from $\theta$ intersections as the starting point of the ants

$m$ ants obtain the probability of going to the next intersection according to the pheromone and the weight between intersections, and the termination condition is $D$ people. After each ant completes the path finding, immediately update the weight between the intersections and the number of evacuees

Updating pheromone under reward mechanism

Maximum number of iterations reached

Yes

Save the $T$ optimal routes of the current rescue station, save the updated weights and the number of evacuees between all intersections

$K = K - 1$

$K = 0$

No

Save $H$ routes corresponding to each rescue station, and a total of $K \times H$ rescue plans

End

Figure 5: The general framework of emergency response planning
1500 and 2000. The different scenarios can be used to comprehensively verify the model’s resilience to real-time road conditions and its effectiveness for path planning. In this paper, the latitude and longitude information of the city is used as the raw road network information. The processing of the road network information follows Algorithm 1 in Section 4.1, and the data set contains road ID, road length, road key points (latitude and longitude representation), and other relevant information. The network information is stored in a json file containing 370 roads, 833 intersections and 5414 key points. The location information of evacuees is also stored in a json file, which contains key information such as the total number of evacuees, the location of evacuees, etc.

Table 3 gives the number of evacuees in each category for each scenario. Data1 through Data6 are in increasing order of the number of evacuees, and the data involved in this section can be downloaded from here.

<table>
<thead>
<tr>
<th>dataset</th>
<th>children</th>
<th>young adults</th>
<th>middle age</th>
<th>old age</th>
<th>serious illness</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data1</td>
<td>64</td>
<td>55</td>
<td>65</td>
<td>60</td>
<td>56</td>
<td>300</td>
</tr>
<tr>
<td>Data2</td>
<td>94</td>
<td>97</td>
<td>100</td>
<td>103</td>
<td>106</td>
<td>500</td>
</tr>
<tr>
<td>Data3</td>
<td>136</td>
<td>137</td>
<td>143</td>
<td>144</td>
<td>140</td>
<td>700</td>
</tr>
<tr>
<td>Data4</td>
<td>177</td>
<td>182</td>
<td>178</td>
<td>180</td>
<td>183</td>
<td>900</td>
</tr>
<tr>
<td>Data5</td>
<td>309</td>
<td>311</td>
<td>293</td>
<td>291</td>
<td>296</td>
<td>1500</td>
</tr>
<tr>
<td>Data6</td>
<td>400</td>
<td>407</td>
<td>381</td>
<td>416</td>
<td>396</td>
<td>2000</td>
</tr>
</tbody>
</table>

5.2. Assessment methods and experimental setup

We judge the merit of the path planning by comparing the total weights obtained by the rescue solutions, which are calculated following Equation (6). In the rescue process, the weight value $W_{ij}$ obtained by the rescue vehicle from

1https://drive.google.com/file/d/1zTzlPEn4Goo0uhFzNKGEQZlau51x8wQC/view?usp=sharing
intersection $i$ to intersection $j$ can be calculated by the following formula:

$$W_{ij} = 3Q_{ij}^c + 1Q_{ij}^y + 2Q_{ij}^m + 4Q_{ij}^o + 5Q_{ij}^s$$  \hspace{1cm} (18)$$

where $Q_{ij}^c$, $Q_{ij}^y$, $Q_{ij}^m$, $Q_{ij}^o$, and $Q_{ij}^s$ are respectively the number of children, young people, middle-aged, elderly and seriously ill people between intersection $i$ and intersection $j$ in real-time conditions. The number of people $N_{ij}$ picked up by rescue vehicles arriving at intersection $j$ from intersection $i$ is:

$$N_{ij} = Q_{ij}^c + Q_{ij}^y + Q_{ij}^m + Q_{ij}^o + Q_{ij}^s$$  \hspace{1cm} (19)$$

In the experiments, we used the six personnel location datasets in Table 3. The number of rescue stations was set to 3, the total number of rescue vehicles was set to 3, the number of rescues per rescue vehicle was set to 3, the number of passengers accommodated in each rescue vehicle was set to 25, and the number of iterations was set to 20. The relative importance of the pheromone factor $\alpha$ and the relative importance of the heuristic factor $\beta$ were set to 1, and the pheromone concentration volatility coefficient was set to 0.5. Constant $P$ in Formula (15) is set to 5, constant $R$ in Formula (16) is set to 0.5, and constant $Q$ in Formula (17) is set to 50. All algorithms were computed on the same machine, and the details of the experimental environment are shown in Table 4.

In this paper, Python libraries, mainly including pandas, numpy and sklearn, were used to process the data and build the model.

<table>
<thead>
<tr>
<th>Table 4: Experimental environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
</tr>
<tr>
<td>Memory</td>
</tr>
<tr>
<td>CPU</td>
</tr>
<tr>
<td>Python version</td>
</tr>
<tr>
<td>pandas version</td>
</tr>
<tr>
<td>numpy version</td>
</tr>
<tr>
<td>sklearn version</td>
</tr>
</tbody>
</table>
5.3. Influence of hyperparameters

In the task of urban emergency rescue, the merit of path planning directly affects the efficiency of rescue. In this section, the influence of each hyperparameter on path planning is analyzed by adjusting the value of hyperparameter. The value ranges and specific descriptions of each hyperparameter are given in Table 5. Since the path planning efficiency of the proposed method is affected by the value of each hyperparameter, it is important to determine the optimal value of each hyperparameter to optimize the effect of the model. In this experiment, all hyperparameters are adjusted while keeping other parameters constant, and the optimal values are the values marked in bold in the table.

Table 5: Setting and description of hyperparameters

<table>
<thead>
<tr>
<th>hyperparameters</th>
<th>values</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>${0.6, 0.8, 1, 1.2, 1.4}$</td>
<td>relative importance of pheromone factor</td>
</tr>
<tr>
<td>$\beta$</td>
<td>${0.6, 0.8, 1, 1.2, 1.4}$</td>
<td>relative importance of heuristic factor</td>
</tr>
<tr>
<td>$P$</td>
<td>${1, 3, 5, 7, 9}$</td>
<td>constant in heuristic factor calculation formula</td>
</tr>
<tr>
<td>$Q$</td>
<td>${10, 30, 50, 70, 90}$</td>
<td>constant in pheromone concentration calculation formula</td>
</tr>
<tr>
<td>$R$</td>
<td>${1, 3, 5, 7, 9}$</td>
<td>reward coefficient in reward mechanism</td>
</tr>
</tbody>
</table>

5.3.1. Influence of $\alpha$

The relative importance of the pheromone factor $\alpha$ directly affects the probability of the rescue vehicle choosing the road section. The larger the value of $\alpha$, the higher the probability that a rescue vehicle will choose a previously traveled roadway. The smaller the value of $\alpha$, the smaller the search range of the algorithm and the easier it is to fall into a local optimum. Figure 6 shows how the relative importance of pheromone factors affects the results in different datasets. With the increase of data size, the stability of the model decreases gradually. This is because when the data scale increases, the number of paths available to rescue vehicles will increase sharply, and it is easier to fall into local optimum. When the value of $\alpha$ is 1, the best rescue path can be planned.
5.3.2. Influence of $\beta$

The relative importance of heuristic factors $\beta$ affects the probability of road section selection, and also reflects the importance of road weight on road section selection. The larger the value of $\beta$, the easier it is for the rescue vehicle to choose the next road section with a larger weight, which can accelerate the convergence of the algorithm, but it is easy to be ignored in the road sections with lower weights and get the local relative optimal path. Figure 7 shows the influence of the relative importance of the heuristic factors on the results under different datasets. When the $\beta$ value is small, the sensitivity of the algorithm to the weights decreases, and the increased randomness makes the convergence of the algorithm slower, which in turn affects the final results. When the $\beta$ value is large, the algorithm will give preference to the sections with larger weights, and the calculation of large-scale data will become difficult. This is because in the case of large-scale data (e.g., Data6), preferring larger weights is more likely to fall into local optimum. When the value of $\beta$ is 1, it can better balance the selection of the unknown road sections and the already traveled road sections.

5.3.3. The value of $P$

$P$ is a constant in Formula (15), which reflects the inspiration degree of the rescue vehicle’s transfer from intersection $i$ to intersection $j$. The larger the value of $P$, the smaller the influence of the weights on the selection of road segments.
Conversely, the selection of road sections is more dependent on the weights. As can be seen from Figure 8, when the $P$ value increases, the volatility of the results is greater. This is because when the influence of the weights becomes smaller, the randomness of the algorithm will increase, and it will be easier to fall into local optimum and difficult to converge. When the data scale is large, we suggest selecting a small $P$ to ensure the stability of the model.

5.3.4. The value of $Q$

$Q$ is a constant in Formula (17), which directly affects how much pheromone is released and reflects the importance of the total weight of the path to the pheromone. The larger the value of $Q$, the more pheromone is released by the
ants. Figure 9 shows the effect on path planning as the $Q$ value becomes larger, that is, the effect of the change in pheromone concentration on the results. Appropriate pheromone concentration can guide the further optimization of the algorithm and is suitable for different rescue situations.

![Graphs showing the influence of the value of $Q$ on the results](image)

Figure 9: The influence of the value of $Q$ on the results

5.3.5. The value of $R$

As the reward coefficient in the reward mechanism, $R$ reduces the volatility of pheromones on the better path and guides the algorithm to further optimize on the current better path. If $R = 1$, there is no reward mechanism. If $R > 1$, it reflects the reward degree of the algorithm for the optimal path. It can be seen from Figure 10 that the path planning results will drop when $R$ takes a value of 1 or takes a larger value. When it comes to smaller data sets, the value of $R$ should not be set too large, because too much reward will make the algorithm fall into local optimum. For the Data6, the performance of the algorithm decreases obviously when the value of $R$ exceeds 5. In other datasets, the volatility of the algorithm is relatively flat. This is because when the reward mechanism is faced with complex rescue environment, the excessive reward coefficient can cause the algorithm to fall into a local optimum.
5.4. **Performance analysis of different algorithm components**

In order to distinguish the impact of different algorithm components on performance, we compare different algorithm components on datasets of various magnitudes. As shown in Table 6, PA-C-IACO(-R) is based on PA-C-IACO with the removal of the reward mechanism in the update pheromone process. PA-C-IACO(-C) is based on PA-C-IACO with the removal of the rescue station siting strategy introduced in Section 4.3. In the calculation process of PA-C-IACO(-C), candidate rescue stations are no longer obtained from the results of clustering, but selected randomly from all intersections. All subsequent experiments were validated 30 times on each data set and the mean, minimum, maximum and variance of all results were derived, where the variance reflects the stability of the algorithm.

From the experimental results, it can be seen that the path planning performance of PA-C-IACO decreases on each dataset after removing the incentive mechanism and the rescue station siting strategy. In each data set, the result of PA-C-IACO(-C) is slightly better than that of PA-C-IACO(-R), but the stability of PA-C-IACO(-C) is not as good as that of PA-C-IACO(-R). This is because in the absence of the rescue station location strategy, the algorithm will randomly select locations as rescue stations and ignore the road network structure and real-time population location, making it difficult to use the most favorable information to complete the rescue response. When the data scale is smaller,
the wrong rescue station location will lead to the local optimal algorithm. In actual scenarios, the setting of rescue stations can directly affect the efficiency of rescue operations.

Table 6: Performance with different algorithm components

<table>
<thead>
<tr>
<th>methods</th>
<th>Data1</th>
<th>Data2</th>
<th>Data3</th>
<th>Data4</th>
<th>Data5</th>
<th>Data6</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA-C-IACO</td>
<td>mean</td>
<td>739</td>
<td>781</td>
<td>802</td>
<td>832</td>
<td>898</td>
</tr>
<tr>
<td></td>
<td>min</td>
<td>688</td>
<td>735</td>
<td>753</td>
<td>777</td>
<td>843</td>
</tr>
<tr>
<td></td>
<td>max</td>
<td>822</td>
<td>842</td>
<td>838</td>
<td>918</td>
<td>964</td>
</tr>
<tr>
<td></td>
<td>var</td>
<td>1260</td>
<td>659</td>
<td>436</td>
<td>961</td>
<td>964</td>
</tr>
<tr>
<td>PA-C-IACO(-R)</td>
<td>mean</td>
<td>718</td>
<td>744</td>
<td>762</td>
<td>792</td>
<td>851</td>
</tr>
<tr>
<td></td>
<td>min</td>
<td>671</td>
<td>698</td>
<td>719</td>
<td>757</td>
<td>806</td>
</tr>
<tr>
<td></td>
<td>max</td>
<td>773</td>
<td>788</td>
<td>806</td>
<td>875</td>
<td>915</td>
</tr>
<tr>
<td></td>
<td>var</td>
<td>568</td>
<td>569</td>
<td>496</td>
<td>722</td>
<td>776</td>
</tr>
<tr>
<td>PA-C-IACO(-C)</td>
<td>mean</td>
<td>713</td>
<td>765</td>
<td>777</td>
<td>795</td>
<td>869</td>
</tr>
<tr>
<td></td>
<td>min</td>
<td>654</td>
<td>676</td>
<td>723</td>
<td>744</td>
<td>810</td>
</tr>
<tr>
<td></td>
<td>max</td>
<td>800</td>
<td>832</td>
<td>848</td>
<td>845</td>
<td>945</td>
</tr>
<tr>
<td></td>
<td>var</td>
<td>1319</td>
<td>1149</td>
<td>771</td>
<td>552</td>
<td>939</td>
</tr>
</tbody>
</table>

The results of one algorithm are better than those of other methods, but the differences of experimental results may be caused by random factors or sampling errors. Therefore, we conducted significance tests on the 30 experimental results of different methods. If the experimental results of method 1 are better than that of method 2 and there is a significant difference in the 30 times results between them, then we can determine that method 1 is indeed better than method 2. In the subsequent significance test, we set the significance level as 0.05. As shown in Table 7, significance test results of component performance of different algorithms are given, where p-value represents the probability value at the corresponding F value, and F-crit is the F critical value at the corresponding significance level. If F is greater than F-crit, P-value is greater than 0.01 and less than 0.05, which means the difference is significant; if P-value is less than 0.05.
0.01, it means the difference is highly significant. If $F$ is smaller than $F_{\text{crit}}$, then P-value is definitely higher than 0.05, which indicates that there is no difference between the two groups’ results. The results in Table 7 show that the P-value of the test results are all less than 0.05, so the inclusion of the incentive mechanism and the rescue station siting strategy has a significant effect on the performance.

<table>
<thead>
<tr>
<th>significance test</th>
<th>Data1</th>
<th>Data2</th>
<th>Data3</th>
<th>Data4</th>
<th>Data5</th>
<th>Data6</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA-C-IACO, PA-C-IACO(-R)</td>
<td>F</td>
<td>P-value</td>
<td>F-crit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8.11755</td>
<td>0.00798</td>
<td>4.18296</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>27.98049</td>
<td>1.13E-05</td>
<td>4.8815</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>94.49399</td>
<td>1.25E-10</td>
<td>16.91183</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>23.52973</td>
<td>3.84E-05</td>
<td>23.37602</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>34.42697</td>
<td>2.29E-06</td>
<td>13.87162</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>32.65219</td>
<td>3.50E-06</td>
<td>39.70155</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PA-C-IACO, PA-C-IACO(-C)</th>
<th>F</th>
<th>P-value</th>
<th>F-crit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10.04043</td>
<td>0.00359</td>
<td>4.18296</td>
</tr>
<tr>
<td></td>
<td>4.8815</td>
<td>0.03519</td>
<td>4.18296</td>
</tr>
<tr>
<td></td>
<td>16.91183</td>
<td>0.00295</td>
<td>4.18296</td>
</tr>
<tr>
<td></td>
<td>23.37602</td>
<td>0.00004</td>
<td>4.18296</td>
</tr>
<tr>
<td></td>
<td>13.87162</td>
<td>0.00084</td>
<td>4.18296</td>
</tr>
<tr>
<td></td>
<td>39.70155</td>
<td>7.00E-07</td>
<td>4.18296</td>
</tr>
</tbody>
</table>

5.5. Further performance comparison

5.5.1. Compared algorithms

The main methods compared in this paper are the simple heuristic rule (Named as Weighted probability algorithm, WPA), an improved ant colony optimization (IACO) algorithm proposed by[4] and an Improved Artificial Bee Colony (IABC) algorithm proposed by[3]. Under the premise of the overall architecture of this paper, WPA uses a weighted probability method to obtain the probability of intersection i transferring to intersection j. This process can be defined as:

$$p_{ij}(t) = \begin{cases} 
\frac{W_{ij}}{\sum_{s \in J(i)} W_{is}}, & j \in J(i) \\
0, & j \notin J(i)
\end{cases} \quad (20)$$

where $p_{ij}(t)$is the probability of moving from intersection i to intersection j at moment t, i is the number of the intersection where the rescue vehicle is currently located, j is the number of the intersection that the rescue vehicle
can reach. \( \sum_{s \in J(i)} W_{is} \) is the sum of the weights of the rescue vehicle going to all reachable intersections, and \( J(i) \) is the set of cities that the rescue vehicle is allowed to choose in next step. The method utilizes the weight values of all accessible intersections in the current position, and the probabilistic selection process makes the rescue vehicles pass through the sections with higher weight preferentially for rescue.

![Path 1, Path 2](image)

Figure 11: Simple heuristic rules

As shown in Figure 11, WPA will preferentially select the next section with a higher weight value for rescue, so this method will preferentially select Path 2 as the rescue Path. However, the rescue solution of Path 1 is significantly better than Path 2, so WPA tends to fall into local optimality instead of global optimality.

IACO[4] proposed an innovative pheromone updating method to obtain a better solution. This method is similar to formula (17) in this paper, where the total value obtained by each ant is taken into account in the pheromone updating process. This particular pheromone update process reduces the pheromones left by the inferior route and increases the pheromones of the superior route.

IABC[3] proposed a two-layer interaction mechanism of inter-group interaction and intra-group interaction on the basis of Artificial Bee Colony Algorithm. The algorithm first divides the original single species group into multiple groups, and in utilizes a parallel search mechanism and collaborative interaction between groups to enhance the search capability of the algorithm and optimize.
the efficiency.

5.5.2. Comparison of results

As the scenario in this paper is a special urban emergency rescue scenario, there is no existing literature solution for a reasonable solution. We replicated the above method and compared it with the method proposed in this paper. As shown in Table 8, experimental comparisons are made on data sets with different numbers of people, and the proposed method consistently outperforms all other methods. Neither IACO nor IABC gives better results than PA-C-IACO, which also indicates that our method is more suitable for the scenes in the paper in terms of the location of rescue stations and rescue planning. Overall, the method proposed in this paper can well complete the urban emergency rescue task, and the clustering-based rescue station location strategy and reward mechanism can optimize the rescue path planning to a great extent.

It can also be seen from Table 9 that the P-values in the significance test results are all less than 0.05, so the results of PA-C-IACO are fundamentally different from those of the other methods. That is, the better results of PA-C-IACO are not caused by randomness, but PA-C-IACO has better modeling ability.

5.6. Some observational analysis

Figure 12-14 shows the curve of weight value changing with the number of iterations starting from each rescue station for rescue. In the figure, the abscissa is the number of iterations and the ordinate is the weight value. The three curves respectively represent the rescue weights obtained from the three rescue stations. Compared with PA-C-IACO(-R), the results of PA-C-IACO have been completely convergent before the number of iterations reaches 20, except for data set Data4. PA-C-IACO(-R) has not been completely convergent on Data1, Data3 and Data4. The result of PA-C-IACO is obviously better than that of PA-C-IACO(-C), and it can be seen that the convergence effect of PA-C-IACO(-C) is the worst. It can be seen that PA-C-IACO can cope
### Table 8: Comparison of experimental results

<table>
<thead>
<tr>
<th>methods</th>
<th>Data1</th>
<th>Data2</th>
<th>Data3</th>
<th>Data4</th>
<th>Data5</th>
<th>Data6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PA-C-IACO</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mean</td>
<td>739</td>
<td>781</td>
<td>802</td>
<td>832</td>
<td>898</td>
<td>969</td>
</tr>
<tr>
<td>min</td>
<td>688</td>
<td>735</td>
<td>753</td>
<td>777</td>
<td>843</td>
<td>887</td>
</tr>
<tr>
<td>max</td>
<td><strong>822</strong></td>
<td>842</td>
<td>838</td>
<td><strong>918</strong></td>
<td>964</td>
<td><strong>1085</strong></td>
</tr>
<tr>
<td>var</td>
<td>1260</td>
<td>659</td>
<td>436</td>
<td>961</td>
<td>964</td>
<td>1626</td>
</tr>
<tr>
<td><strong>WPA</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mean</td>
<td>705</td>
<td>747</td>
<td>785</td>
<td>800</td>
<td>876</td>
<td>947</td>
</tr>
<tr>
<td>min</td>
<td>641</td>
<td>701</td>
<td>746</td>
<td>730</td>
<td>805</td>
<td>766</td>
</tr>
<tr>
<td>max</td>
<td>754</td>
<td>797</td>
<td>834</td>
<td>864</td>
<td>940</td>
<td>1047</td>
</tr>
<tr>
<td>var</td>
<td><strong>738</strong></td>
<td>444</td>
<td>644</td>
<td>1073</td>
<td>1328</td>
<td>3667</td>
</tr>
<tr>
<td><strong>IACO[4]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mean</td>
<td>704</td>
<td>763</td>
<td>784</td>
<td>808</td>
<td>876</td>
<td>941</td>
</tr>
<tr>
<td>min</td>
<td>654</td>
<td>707</td>
<td>722</td>
<td>736</td>
<td>818</td>
<td>854</td>
</tr>
<tr>
<td>max</td>
<td>784</td>
<td>821</td>
<td>826</td>
<td>920</td>
<td>976</td>
<td>1056</td>
</tr>
<tr>
<td>var</td>
<td>956</td>
<td>861</td>
<td>942</td>
<td>1660</td>
<td>1560</td>
<td>2255</td>
</tr>
<tr>
<td><strong>IABC[3]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mean</td>
<td>717</td>
<td>756</td>
<td>787</td>
<td>812</td>
<td>879</td>
<td>952</td>
</tr>
<tr>
<td>min</td>
<td>636</td>
<td>664</td>
<td>748</td>
<td>750</td>
<td>821</td>
<td>836</td>
</tr>
<tr>
<td>max</td>
<td>795</td>
<td>819</td>
<td>842</td>
<td>890</td>
<td>937</td>
<td>1064</td>
</tr>
<tr>
<td>var</td>
<td>1666</td>
<td>1207</td>
<td>744</td>
<td>1101</td>
<td>939</td>
<td>1980</td>
</tr>
</tbody>
</table>

### Table 9: Significance tests of PA-C-IACO and other methods

<table>
<thead>
<tr>
<th>significance test</th>
<th>Data1</th>
<th>Data2</th>
<th>Data3</th>
<th>Data4</th>
<th>Data5</th>
<th>Data6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PA-C-IACO, WPA</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>15.38993</td>
<td>28.82791</td>
<td>8.98939</td>
<td>12.05749</td>
<td>5.65841</td>
<td>4.53714</td>
</tr>
<tr>
<td>P-value</td>
<td>0.00049</td>
<td>9.10E-06</td>
<td>0.00552</td>
<td>0.00164</td>
<td>0.02417</td>
<td>0.04178</td>
</tr>
<tr>
<td><strong>PA-C-IACO, IACO</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>19.89676</td>
<td>5.69875</td>
<td>7.26274</td>
<td>7.00866</td>
<td>5.56039</td>
<td>6.78094</td>
</tr>
<tr>
<td>P-value</td>
<td>0.00011</td>
<td>0.02372</td>
<td>0.01159</td>
<td>0.01298</td>
<td>0.02533</td>
<td>0.01438</td>
</tr>
<tr>
<td><strong>PA-C-IACO, IABC</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>4.38918</td>
<td>7.30502</td>
<td>5.57186</td>
<td>5.89732</td>
<td>6.33991</td>
<td>7.18765</td>
</tr>
<tr>
<td>P-value</td>
<td>0.04501</td>
<td>0.01137</td>
<td>0.02519</td>
<td>0.0216</td>
<td>0.01758</td>
<td>0.01198</td>
</tr>
</tbody>
</table>

Electronic copy available at: https://ssrn.com/abstract=4041695
with urban emergency rescue of different scales and select a better rescue path. As a special attentional mechanism, reward mechanism can accelerate model convergence and help to find the optimal solution. Since the location of the rescue station is directly related to the path planning, PA-C-IACO(-C) can use the reward mechanism to better plan the path, but the poor rescue station setting greatly reduces the effectiveness of the path planning.

![Figure 12: Variation curve of weight value under PA-C-IACO](https://ssrn.com/abstract=4041695)

![Figure 13: Variation curve of weight value under PA-C-IACO(-R)](https://ssrn.com/abstract=4041695)

Figures 15-17 visualize the location of rescue stations at different rescue scales. The left side of the figure shows a visualization of the rescue station location planning by clustering method, and the right side shows a visualization of the randomly selected rescue station location. In the case of different rescue
scales, the proposed rescue station location scheme can take good care of the whole city and facilitate the reasonable allocation of rescue resources. The randomized site selection will face rescue difficulties, and the location of the rescue station will reduce the rescue efficiency and increase the difficulty of rescue if it is difficult to map to the whole city. When faced with a large number of affected people or a lack of rescue resources, the location of the rescue station determines the utilization of rescue resources.
Figure 16: Visualization of rescue station location based on data2

Figure 17: Visualization of rescue station location based on data3
6. Conclusion

In this paper, a realistic urban emergency rescue scenario is proposed and a complete rescue planning scheme is given. We quantify the priority of each evacuee into weight values, which are then used as the optimization objectives of the model. This allows for maximum priority rescue of the sick, elderly and children which is more in line with humanitarian and rescue requirements. Secondly, we subdivide the rescue planning method into four parts of road network processing, calculation of road network weights, rescue station siting and rescue path planning to deal with multiple problems in a systematic way. Further, we use both the location of evacuees and the road network structure in the rescue station siting stage to obtain a reasonable set of candidate rescue stations and provide a good reference starting point for path planning. In this paper, we propose to use an improved ant colony optimization algorithm to randomly select the starting point from the set of candidate rescue stations in the path planning stage, and iterate to select the optimal path and rescue station location. We extend the traditional ant colony algorithm and redefine the degree of heuristic for intersection transfer and the calculation of pheromone concentration increments. The pheromone update process incorporates a reward mechanism to speed up the convergence of the model and improve the results. Finally, we validate PA-C-IACO on six datasets of different sizes. The results show that our proposed approach outperforms other baselines and is robust.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 71871185.

References


