Computer Fraud and Security
ISSN (online): 1873-7056

# Optimization of Rescue Team Scheduling under High Altitude Earthquakes Based on Reinforcement Learning

**Jun Liu[1,2], Xinhao Li[1], Sha Chen[3*], Ying Wang[2], Chunyan Han[2]**

1 College of Information Technology, Shanghai Ocean University, Shanghai 201306, China;

2 National Earthquake Response Support Service, Beijing 100049, China;

3 National Disaster Reduction Center of China, Ministry of Emergency Management of China, Beijing 100124, China

* Correspondence: chensha@ndrcc.org.cn

**Abstract:** In the early stages of high-altitude earthquake disasters, efficient rescue team scheduling is critical to minimize casualties and optimize resource utilization. This study proposes HRLPPO, a hierarchical reinforcement learning framework combining Proximal Policy Optimization (PPO) with task stratification, to address the dynamic allocation of rescue teams under complex constraints. The framework divides rescue tasks into a high-level strategy for selecting teams and disaster sites, and a low-level strategy for determining personnel dispatch quantities. Key innovations include integrating high-altitude compatibility constraints, minimizing dispatch costs via distance-aware reward functions, and enabling rapid decisions through pre-trained policies.A custom reinforcement learning environment was designed to simulate real-world scenarios, incorporating rescue team capabilities, site demands, and geographical constraints. Experiments using data from the 2022 Luding earthquake in Sichuan demonstrated HRLPPO's superiority over traditional methods (e.g., Genetic Algorithm, Ant Colony Optimization). Results showed 18.5% lower dispatch costs, 95% faster decision times (0.43s vs. 487.43s for 10-team scenarios), and 99.79% rescue satisfaction rates under both sufficient and insufficient high-altitude team conditions. The model's robustness was further validated in large-scale scenarios (40 teams, 20 sites), achieving 81.68% overall satisfaction despite resource shortages.This work provides a novel decision-making tool for emergency management, enhancing rescue efficiency in high-altitude regions. Future efforts will integrate GIS platforms for real-time disaster response.

**Keywords:** Emergency Rescue; Hierarchical Reinforcement Learning (HRL); Proximal Policy Optimization (PPO); Resource Allocation; Earthquake Response

## 1. Introduction

After natural disasters such as earthquakes, it is crucial to quickly and accurately assess post-disaster rescue needs in the affected areas and to ensure a rapid response and effective allocation of rescue teams. Proper allocation of rescue teams can quickly organize a strong rescue force, operate efficiently, and quickly rescue trapped people, reducing casualties. At the same time, allocation can ensure the rational use of rescue resources, avoid waste and redundant investment, and provide the necessary material and human support for rescue operations. Rapidly assessing post-earthquake rescue needs in the affected areas and emergency rescue needs for the allocation and planning of rescue team deployment is an important reference for government departments and emergency management departments to initiate emergency responses and deploy rescue forces. Accurate assessments help to carry out rescue operations to the greatest extent, which is an important scientific issue in the field of post-earthquake emergency rescue in China at this stage.

With the increasing emphasis on resource allocation and route planning issues in various fields, many scholars have conducted in-depth research using various algorithms and achieved many results. Fan et al. [1] proposed a reinforcement learning-based resource allocation mechanism for multi-vehicle communication-assisted perception systems, demonstrating the potential for optimizing resource allocation in dynamic environments. Zhong et al. [2] used deep reinforcement learning to achieve low-carbon optimization of user-side shared energy storage and distribution networks, further proving the application value of deep learning technology in resource optimization problems. Middelhuisa et al. [3] proposed a learning-based resource allocation method to solve resource

allocation problems in business processes, and verified the effectiveness of their method through simulation experiments. Wang et al. [4] addressed the problem of rescue resource allocation and scheduling in emergency logistics during storm surges, using deep reinforcement learning methods, demonstrating the practicality of optimizing resource allocation in emergencies. Chu and Zhong [5] explored resource optimization issues in disaster response in their research on the allocation methods of medical rescue teams after earthquakes. Kool et al. [6] demonstrated solving CVRP and TSP distribution route planning problems through attention mechanisms and learning in their research. Ma Zhenpeng et al. [7] studied vehicle route optimization algorithms for urban logistics distribution, emphasizing the importance of optimizing route planning in the logistics field. Zhao Zilong et al. [8] considered the limited nature of resources in their research on emergency rescue scheduling optimization under major forest fires, which is a typical example of resource allocation in emergencies. Zou Shanshan et al. [9] provided practical methods for optimizing resource allocation in disaster response in their research on emergency logistics vehicle route optimization under urban flood disasters. Jia Tingting et al. [10] conducted research on emergency rescue and scheduling optimization for subways under extreme rain conditions, demonstrating the importance of resource allocation in public transportation systems. Sun Yan et al. [11] studied the optimization of emergency material multimodal transport route planning in an interval fuzzy environment, providing solutions for resource allocation in uncertain environments. Lou Zibo et al. [12] studied path planning systems through improved Q-Learning algorithms, providing new solutions for dynamic route planning problems. Tang Hongwei et al. [13] used ant colony algorithms to study the path planning problem of material transport carts, providing a new perspective for logistics path optimization. Lv Chao et al. [14] adopted a hierarchical deep reinforcement learning method in their research on UAV hybrid path planning, providing an innovative solution for UAV path planning problems. Kong Lin et al. [15] used an improved ant colony algorithm in their research on emergency rescue route planning for ambulances, providing an effective strategy for resource allocation in emergency medical services. Deng Daojing et al. [16] used parallel GA-PSO algorithms in their research on collaborative task planning for multiple UAVs, providing a new planning method for collaborative work of UAVs.

In summary, against this background, this paper focuses on the important issue of earthquake rescue team scheduling, fully integrating the special attributes of high-altitude rescue teams and the specific locations of earthquake points to construct a model for rescue team scheduling. By adopting a task-stratified deep reinforcement learning method, the rescue tasks are divided into two levels: high-level strategy and low-level strategy. The high-level strategy is responsible for the selection of rescue teams and earthquake points, while the low-level strategy mainly controls the number of rescuers dispatched to the rescue teams and earthquake points. Based on the distinction of rescue reserves into sufficient and insufficient high-altitude rescue reserves, the model is deeply analyzed, and the model is converged using a reward function. The model's effectiveness is verified through example rescue decision analysis, aiming to provide valuable reference for improving the level of earthquake rescue in high-altitude areas.

## 2. Emergency Rescue Model for Multiple Rescue Teams and Multiple Affected Points

### *2.1 Problem Description*

The multi-rescue team and multi-disaster point emergency rescue problem involves determining the most reasonable rescue strategy for each earthquake-affected point, ensuring that the rescue needs of each affected point are met while enabling each rescue team to quickly reach the affected points and carry out efficient rescue operations.

**Table 1.** Problem Description Symbol Meanings.

| Parameter Type | Parameter | Meaning |
|---|---|---|
| Basic Attributes | $\boldsymbol{T}$ | The set of all rescue teams |
| | $\boldsymbol{T_i}$ | Rescue team |
| | $\boldsymbol{S}$ | The set of all earthquake disaster sites |
| | $\boldsymbol{S_j}$ | earthquake disaster site |

| | | |
|---|---|---|
| Parameter | $l_{t_i}$ | Location coordinates of rescue team |
| | $c_i$ | Rescue team available rescue capacity (number of people) |
| | $h_{t_i}$ | Is it a high-altitude rescue team |
| | $l_{s_j}$ | Location coordinates of the disaster site |
| | $d_j$ | Demand for rescue teams in disaster site |
| | $h_{s_j}$ | Is it located in a high-altitude area |
| Decision variables | $x_{i_j}$ | Number of rescue personnel dispatched by the team and disaster stricken areas |

As described in Table 1 for the meaning of the model letter notations, the set of rescue teams is denoted as $T = \{t_1, t_2, \cdots, t_m\}$, where $m$ represents the number of teams. The set of earthquake-stricken points is $S = \{s_1, s_2, \cdots, s_n\}$, where $n$ represents the number of earthquake-stricken points. For all rescue teams $t_i \in T$: - Location coordinates: $l_{(t_i)} = \left(x_{(t_i)}, y_{(t_i)}\right) \in R^2$ - Rescue capacity: $c_i \in N$ (the set of natural numbers) - High-altitude support: $h_{(t_i)} \in \{0,1\}$

For all earthquake-stricken points $s_j \in S$: - Location coordinates: $l_{(s_j)} = \left(x_{(s_j)}, y_{(s_j)}\right) \in R^2$ - Quantity of rescue demand: $d_j \in N$ - Whether it is an earthquake-stricken point in a high-altitude area: $h_{(s_j)} \in \{0,1\}$

The emergency rescue model is as follows:

$$D = \sum_{i=1}^{N}\sum_{j=1}^{M} \text{dist}\left(l_{(t_i)}, l_{(s_j)}\right) \cdot x_{ij} \quad (1)$$

$$\forall t_i \in T, \sum_{j=1}^{n} x_{ij} \leq c_i \quad (2)$$

$$\forall s_j \in S, \sum_{i=1}^{m} x_{ij} \geq d_j \quad (3)$$

$$\forall t_i \in T, \forall s_j \in S, h_{(s_j)} = 1 \wedge h_{(t_i)} = 0 \Rightarrow x_{ij} = 0 \quad (4)$$

$$\forall t_i \in T, \forall s_j \in S; x_{ij} \in N_0 \quad (5)$$

Equation (1) is the objective function that minimizes the sum of the rescue distance and the number of rescued people to minimize the rescue cost. It represents the sum of the rescue distance of each team and the number of rescued people. Equations (2) - (5) are constraint conditions. Among them, Equation (2) is the capacity constraint of the rescue team. The number of people dispatched for each rescue mission shall not exceed the remaining total dispatchable capacity of the team itself. Equation (3) indicates that the earthquake-stricken points need to have their rescue demand met. Equation (4) is the high-altitude constraint. The earthquake-stricken points in high-altitude areas can only be rescued by teams with high-altitude rescue capabilities. Otherwise, the rescue mission will not be executed, that is, the number of dispatched rescuers is 0. Equation (5) represents the constraint that the number of rescuers dispatched by the rescue teams is an integer.

*2.2 Markov modeling for emergency rescue dispatching problems.*

This article adopts the deep reinforcement learning approach to solve the problem of rescue team dispatch and allocation in rescue operations. Firstly, the mathematical formula problem description designed for the emergency

rescue team dispatch issue is transformed into the MDP (Markov Decision Process) framework. The MDP framework is a mathematical space used to solve complex decision-making problems, deal with delayed rewards and so on. The core components of the MDP framework generally include the state space, action space, transition probability, reward function and discount factor.

2.2.1 State space

For the problem of dispatching rescue teams to earthquake-stricken points, in the location coordinate matrix of rescue teams, for each rescue team $T_i$, its location coordinates are $l_{(t_i)} = (x_{(t_i)}, y_{(t_i)})$. The location coordinates of all rescue teams form a matrix, denoted as $LOC_t = \begin{bmatrix} l_{(t_1)} \\ l_{(t_2)} \\ \vdots \\ l_{(t_m)} \end{bmatrix} \in R^{m\times 2}$, where $m$ is the number of rescue teams. This matrix reflects the distribution of rescue teams in the geographical space and is one of the important pieces of information in the decision-making process, because it determines the time and spatial distance required for rescue teams to reach the earthquake-stricken points. Among them, for the rescue team $T_i$, $loc_{(T_i)} = (x_{(T_i)}, y_{(T_i)})$; similarly, for the earthquake-stricken point $S_j$, the location matrix is $loc_{(S_j)} = \left(x_{(S_j)}, y_{(S_j)}\right)$. By combining the location information of the teams and the earthquake-stricken points, the location matrix of the overall state space is:

$$LOC_t = \begin{bmatrix} loc_{(T_1)} \\ loc_{(T_2)} \\ \vdots \\ loc_{(T_m)} \\ loc_{(S_1)} \\ loc_{(S_2)} \\ \vdots \\ loc_{(S_n)} \end{bmatrix} \in R^{(m+n)\times 2} \quad (6)$$

where $m$ is the number of rescue teams and $n$ is the number of earthquake-stricken points.

The current available capacity matrix of rescue teams can be expressed as $C_t = \left[C^r_{(T_1)}, C^r_{(T_2)}, \cdots, C^r_{(T_m)}\right]^T \in R^m$ (7)

$$C^r_{(T_i)} = C_{(T_i)} - \sum_{j=1}^{n}\sum_{k=0}^{t-1} q_{ijk} \quad (8)$$

where $q_{ijk}$ represents the number of rescuers assigned from team $T_i$ to earthquake-stricken point $S_j$ at time step $k$.

$$D^r_{(S_j)} = D_{(S_j)} - \sum_{i=1}^{m}\sum_{k=0}^{t-1} q_{ijk} \quad (9)$$

The remaining demand matrix of earthquake-stricken points can be expressed as $D_t = \left[D^r_{(S_1)}, D^r_{(S_2)}, \cdots, D^r_{(S_n)}\right]^T \in R^n$, where $D^r_{(S_j)}$ represents the rescue demand of earthquake-stricken point $S_j$.

The allocation matrix $Q_t = \left[q^t_{ij}\right] \in R^{m\times n}$, where $q^t_{ij}$ represents the number of rescuers assigned from team $T_i$ to earthquake-stricken point $S_j$ at time step $t$.

In the problem of dispatching rescue teams to earthquake-stricken points for rescue, the overall state space can be expressed as $S$:

$$S = \{s_t = (LOC_t, C_t, D_t, Q_t, t, i_t)\} \quad (10)$$

2.2.2 Action space

The action space $A$ contains the indices of all decision-making teams, the indices of earthquake-stricken points and the number of rescuers for support. In the problem of dispatching rescue teams to earthquake-stricken points for rescue, an action $a = (i, j, q)$ is designed, where: $i \in \{1,2,\cdots,m\}$, which represents the index of the selected rescue team and corresponds to team $T_i$ in the rescue team set $T$. $j \in \{1,2,\cdots,n\}$, which represents the index of the selected earthquake-stricken point and corresponds to earthquake-stricken point $S_j$ in the earthquake-stricken point set $S$. $q$ represents the number of resources allocated to earthquake-stricken point $S_j$, and satisfies the constraint:

$$0 \le q \le \min\left\{C_{(T_i)} - \sum_{k=0}^{t-1} q_{ik}, D_{(S_j)} - \sum_{k=0}^{t-1} q_{kj}\right\} \quad (11)$$

The action space $A$ can be expressed as:

$$A = \left\{a = (i,j,q) \,\middle|\, i \in \{1,2,\cdots,m\}, j \in \{1,2,\cdots,n\}, 0 \le q \le \min\left\{C_{(T_i)} - \sum_{k=0}^{t-1} q_{ik}, D_{(S_j)} - \sum_{k=0}^{t-1} q_{kj}\right\}\right\} \quad (12)$$

2.2.3 Transition probability

In a deterministic environment, for a given current state $s_t$ and action $a$, the transition probability of transitioning to the next state $s_{t+1}$ is: $P(s_{t+1} \mid s_t, a)$.

2.2.4 Reward

The design of the reward function aims to balance the efficiency and effectiveness of rescue operations, while taking into account the costs and time urgency of rescue operations. The reward function is designed as follows:

For the rescue effectiveness part: $R_{eff}(a) = q$.

For the distance penalty part: Define the distance function $\text{dist}(T_i, S_j) = \sqrt{\left(x_{(T_i)} - x_{(S_j)}\right)^2 + \left(y_{(T_i)} - y_{(S_j)}\right)^2}$, then the distance penalty is $R_{dist}(a) = d \cdot \text{dist}(T_i, S_j) \cdot q$, where $d$ is the distance penalty factor.

For the time urgency part: $R_{time}(t) = \tau \cdot t \cdot \sum_{j=1}^{n} D_{(S_j)}$, where $\tau$ is the time urgency factor.

For the completion reward part: $R_{comp}(a) = \rho \cdot \delta\left(D^r_{(S_j)} = 0 \wedge D^0_{(S_j)} > 0\right)$, where $\rho$ is the completion reward value, and $\delta$ is an indicator function which takes the value of $1$ when the condition is met and $0$ otherwise, and $D^0_{(S_j)}$ is the initial demand of earthquake-stricken point $S_j$.

For the unmet demand penalty part: $R_{unsat}(a) = \theta \cdot \delta\left(D^r_{(S_j)} > 0\right)$, where $\theta$ is the unmet demand penalty value.

For the high-altitude mismatch penalty part: $R_h(a) = \xi \cdot \delta\left(H_{(S_j)} = 1 \wedge H_{(T_i)} = 0\right)$, where $\xi$ is the high-altitude mismatch penalty value.

The total reward function is:

$$R(a, s_t) = R_{eff}(a) - R_{dist}(a) - R_{time}(t) + R_{comp}(a) - R_{unsat}(a) - R_h(a) \quad (13)$$

**3. Task based hierarchical reinforcement learning for scheduling high-altitude earthquake rescue teams**

This paper uses hierarchical reinforcement learning combined with the Proximal Policy Optimization (PPO) algorithm to solve the aforementioned Markov Decision Process (MDP) problem. It combines the ideas of hierarchical reinforcement learning and the stability of the PPO algorithm, and meanwhile introduces an experience replay buffer to improve sample efficiency. The main part of the Actor-Critic Network of this algorithm adopts a hierarchical strategy. The high-level strategy is responsible for selecting subtasks (target rescue points), and the low-level strategy mainly selects specific rescue actions (the number of dispatched rescuers for specific rescue

actions) based on the output of the high-level strategy. Both the High-level and Low-level contain their respective actor networks and critic networks.

In the High-level Actor, it is represented by the function $H(\theta_H)$, where $\theta_H$ is the parameter of the high-level strategy. It selects a high-level action according to the current environmental state $s$, that is, it determines the earthquake-stricken points to focus on. Mathematically, the high-level strategy can be regarded as a probability distribution $\pi_{\theta_H}(a_H|s)$, where $a_H$ is a high-level action, such as the index of the selected earthquake-stricken point. In the High-level Critic, it is represented by the function $V_H(s)$, which is an estimate of the state value. Its goal is to evaluate the expected long-term cumulative reward of following the high-level strategy under state $s$, and it can be expressed as:

$$V_H^{\pi_{\theta_H}}(s) = E_{\pi_{\theta_H}}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t, a_H)|s_0 = s\right] \quad (14)$$

where $\gamma$ is the discount factor, and $R(s_t, a_H)$ is the reward obtained by taking the high-level action $a_H$ under state $s_t$.

The rollout buffer is mainly used to store the trajectory data generated during the interaction between the agent and the environment, and these data will be used for subsequent policy optimization. Its storage form is usually tuples $(s, s_t, a_H, r, s')$, where $s$ is the current state, $s_t$ is the intermediate state transition from state $s$ to the next state $s'$, $a$ is the action taken (for the high-level strategy, $a$ is a high-level action such as selecting earthquake-stricken points and can be represented as a probability distribution, where $\theta_H$ is the parameter of the high-level strategy; for the low-level strategy, it is a low-level action of determining rescue teams and allocating the number of rescuers and can be represented as a probability distribution $\pi_{\theta_H}(a_H|s)$, where $\theta_H$ is the parameter of the low-level strategy), $r$ is the reward obtained, which reflects the feedback given by the environment after taking action $a$ under state $s$, and $s'$ is the next environmental state.

The goal is to maximize the expectation of the long-term cumulative reward, and its objective function is:

$$J_H(\theta_H) = E_{s\sim\rho^{\pi_{\theta_H}}}\left[V_H^{\pi_{\theta_H}}(s)\right] \quad (15)$$

where $\rho^{\pi_{\theta_H}}$ is the state distribution under the policy $\pi_{\theta_H}$.

During the optimization process, data is sampled from the rollout buffer to estimate the policy gradient. The policy gradient method is used for optimization. A batch of samples sampled from the rollout can be represented as $\{(s_i, s_{t,i}, a_{H,i}, r_i, s_i')\}_{i=1}^{B}$, where $B$ is the batch size. The advantage function can be estimated as $A_{t,i} = \sum_{l=0}^{\infty}(\gamma\lambda)^l \delta_{t+l,i}$, where $\delta_{t,i} = r_i + \gamma V_{oH}(s_i') - V_{oH}(s_i)$.

The policy gradient update formula is:

$$\nabla_{\theta_H} J_H(\theta_H) \approx \frac{1}{B}\sum_{i=1}^{B}\nabla_{\theta_H}\log\pi_{\theta_H}(a_{H,i}|s_i)A_{t,i} \quad (16)$$

where $Q_H^{\pi_{\theta_H}}(s, a_H)$ is the action-value function after taking the high-level action $a_H$ under state $s$.

In the Low-level Actor, it is represented by the function $L(\theta_L)$, where $\theta_L$ is the parameter of the low-level strategy. Given the earthquake-stricken points selected by the high-level strategy and the current environmental state, it selects specific rescue actions, such as determining rescue teams and the amount of allocated resources. It can be represented as a probability distribution $\pi_{\theta_L}(a_L|s, a_H)$, where $a_L$ is a low-level action and $a_H$ is a high-level action. The Low-level Critic is represented by $V_L(s, a_H)$, which evaluates the value of state $s$ under the given high-level action $a_H$, that is, it measures the expected long-term cumulative reward of following the low-level strategy in this situation.

The goal is also to maximize the expectation of the long-term cumulative reward, and the objective function is

$$J_L(\theta_L) = \mathbb{E}_{s\sim\rho^{\pi_{\theta_H},\pi_{\theta_L}}} \sum_{a_H\in\mathcal{A}_H} \pi_{\theta_H} \sum_{a_L\in\mathcal{A}_L} \pi_{\theta_L} Q_L^{\pi_{\theta_H},\pi_{\theta_L}}(17)$$

where $\rho^{\pi_{\theta_H},\pi_{\theta_L}}$ is the state distribution under the combination of high-level and low-level strategies, and $Q_L^{\pi_{\theta_H},\pi_{\theta_L}}(s, a_H, a_L)$ is the action-value function after taking the high-level action $a_H$ and the low-level action $a_L$ under state $s$. Data is sampled from the rollout buffer to estimate the policy gradient, and the policy gradient is approximately updated as:

$$\nabla_{\theta_H} J_H(\theta_H) \approx \frac{1}{B}\sum_{i=1}^{B} \nabla_{\theta_H} \log \pi_{\theta_H}(a_{H,i}|s_i) A_{t,i} \quad (18)$$

The hierarchical PPO agent is used to complete the training, and the trained hierarchical policy network is used for rescue decision-making. The agent can complete the decision-making in milliseconds. The framework of the hierarchical PPO algorithm process is shown in Figure 1.

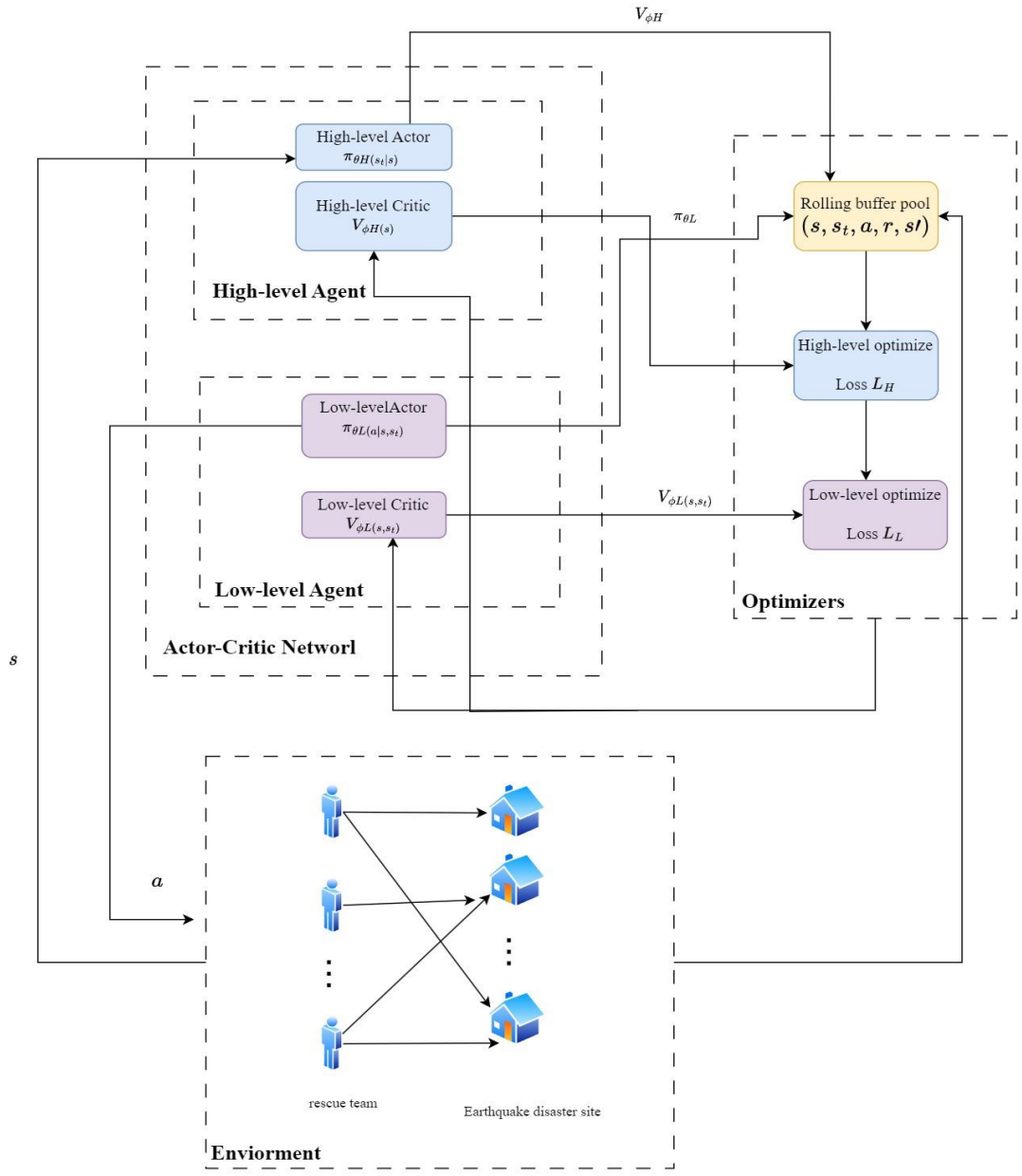



Figure 1. HRLPPO algorithm process framework

## 4. Experiment and Analysis

In order to verify the above-mentioned problem of rescue team dispatch and allocation in rescue operations, tests were conducted on different scales of rescue teams and earthquake-stricken points randomly generated from the combination of the data of rescue teams and the rescue team demands of earthquake-stricken points in the actual case of the Luding earthquake in Sichuan in 2022.

### *4.1 Experimental setup*

The experiment uses Reinforcement Learning (RL) to train the model earthquake-stricken points of different scales. The training is based on Python 3.9, PyTorch 2.1.2, stable_baselines3 2.3.2 and Gymnasium 0.29.1. The settings of the hyperparameters related to the model training algorithm are shown in Table 2.

### *4.2Analysis of Model Training Effect*

During the model training period, Figure 2 shows the reward curves of the cumulative rewards per episode for the traditional Proximal Policy Optimization (PPO) algorithm and the hierarchical task-based HRLPPO (proposed in this paper) algorithm. The comparison of the curves reveals that in the first 200,000 time steps, the agents are in the exploration phase, and as they learn how to navigate the policy space, the rewards fluctuate significantly. After approximately 200,000 time steps, the rewards of the agents start to stabilize, indicating that they have learned effective strategies for the given tasks.

The light green curve represents the hierarchical task-based HRLPPO algorithm proposed in this paper. This algorithm disassembles tasks into high-level and low-level strategies and adopts different solution strategies for hierarchical subtasks, enabling it to adapt to the environment more quickly. The light orange curve in the figure represents the training process of the traditional PPO algorithm. During the training process, the light green curve converges and reaches the highest reward value more quickly than the light orange curve at almost the same training time steps. In the training process, under the same convergence conditions, the light green curve has a higher cumulative reward value. After reaching the convergence conditions, the exploration noise and the randomness of the environment itself may lead to fluctuations in the reward curve. After reaching a stable state, the hierarchical task-based HRLPPO algorithm proposed in this paper has more stable fluctuations than the traditional PPO algorithm.

**Table 2**. Algorithm-related hyperparameter settings

| Hyperparameter | Value |
|---|---|
| Total Training Steps | $5 * 10^5$ |
| Hidden Layer Dimension | 256 |
| Discount Factor | 0.96 |
| Clipping Range | 0.3 |
| Replay Buffer Size | 2048 |
| Actor Learning Rate | $3 \times 10^-2$ |
| Critic Learning Rate | $3 \times 10^-2$ |
| Network Architecture | MlpPolicy |
| High-Level PolicyActor | Linear→ReLU→Linear→ReLU→Linear→Tanh |
| High-Level Policy Critic | Linear→Tanh→Linear→Tanh→Linear |
| Low-Level Policy Actor | Linear→ReLU→Linear→ReLU→Linear→Tanh |
| Low-Level Policy Critic | Linear→Tanh→Linear→Tanh→Linear |

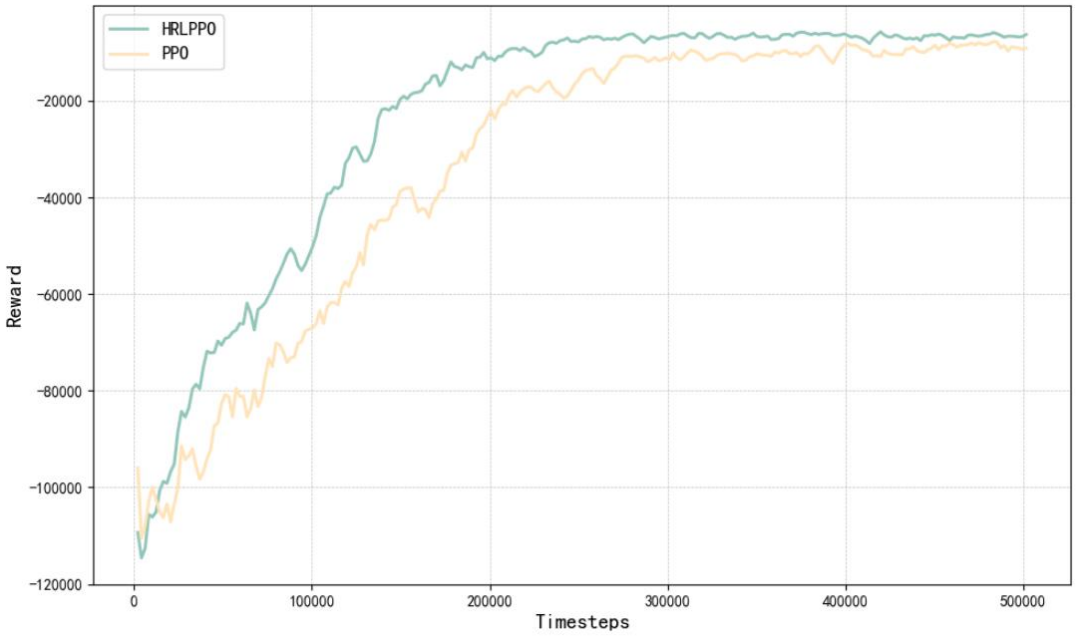



**Figure 2.** The cumulative reward changes with the training timesteps.

*4.3 Analysis of Model Decision-making Effect*

To ensure the rationality of the experiment and verify the performance of the hierarchical task-based HRLPPO algorithm proposed in this paper in the problem of rescue team dispatch and allocation, its decision-making effect is compared with that of the Genetic Algorithm (GA) in Reference [7], the Ant Colony Optimization (ACO) algorithm in Reference [15], and the GA-PSO algorithm in Reference [16]. Based on the data of rescue teams in the actual case of the Luding earthquake in Sichuan in 2022, different-scale experimental example environments with the number of rescue teams-earthquake-stricken points randomly generated as 10-5, 20-10, and 40-20 are created. In these examples, the rescue allocation situations under the circumstances of sufficient reserve of rescue teams and the shortage of reserve of rescue teams are compared between the solutions obtained by the above three algorithms and the rescue decision-making of the hierarchical task-based HRLPPO algorithm proposed in this paper.

**Table 3** Rescue Teams and Earthquake Points with High Altitude Area Data

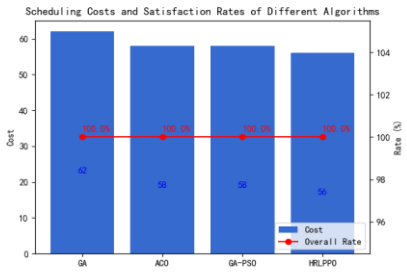
| Number of Rescue Teams | Number of Earthquake Points | Number of People Rescued by Rescue Teams | Number of Rescue Demands at Earthquake Points | Number of People Rescued in High Altitude Areas | Number of Rescue Demands at High Altitude Earthquake Points |
|---|---|---|---|---|---|
| 10 | 5 | 673 | 218 | 311 | 183 |
| 20 | 10 | 1272 | 451 | 585 | 412 |
| 40 | 15 | 2365 | 604 | 865 | 432 |

**Table 4** Scheduling Costs and Decision Times of GA and ACO

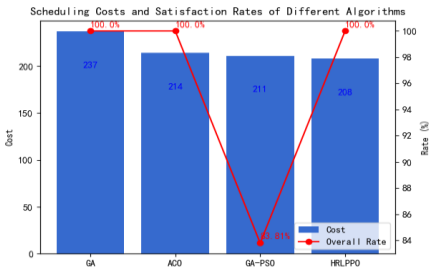
| Number of Rescue Teams | Scheduling Cost (GA) | Decision Time (GA)/s | Scheduling Cost (ACO) | Decision Time (ACO)/s |
|---|---|---|---|---|
| 10 | 61.7 | 9.91 | 58.18 | 3.67 |
| 20 | 237.31 | 18.90 | 213.55 | 9.01 |
| 40 | 501.36 | 54.66 | 485.88 | 11.04 |

**Table 5** Scheduling Costs and Decision Times of GA-PSO and HRLPPO

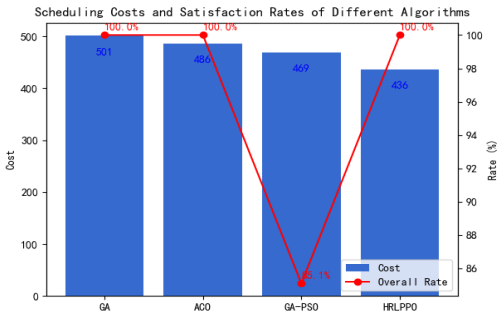
| Number of Rescue Teams | Scheduling Cost (GA-PSO) | Decision Time (GA-PSO)/s | Scheduling Cost (HRLPPO) | Decision Time (HRLPPO)/s |
|---|---|---|---|---|
| 10 | 58.18 | 487.43 | 56.21 | 0.43 |
| 20 | 157.35 | 226.63 | 208.2 | 0.68 |
| 40 | 469.31 | 776.99 | 436.20 | 0.93 |




**(a)** Satisfaction Rates of Different Algorithms under a Rescue Team Scale of 10




**(b)** Satisfaction Rates of Different Algorithms under a Rescue Team Scale of 20




**(c)** Satisfaction Rates of Different Algorithms under a Rescue Team Scale of 40

**Figure 3.** Scheduling cost and satisfaction rate of different algorithms

The experiment was conducted by distinguishing between the conditions of sufficient rescue teams (when the reserve quantity of high-altitude rescue teams is greater than the high-altitude earthquake rescue demand, which is subsequently referred to as sufficient rescue reserve) and the conditions of shortage of rescue teams (when the reserve quantity of high-altitude rescue teams is less than the high-altitude earthquake rescue demand, which is subsequently referred to as insufficient rescue reserve).

Under the condition of sufficient rescue reserve, a comparison of the dispatch cost and the decision-making time consumed by different methods was carried out. The specific experimental results are shown in Table 3-5. Under a small scale (with a rescue team size of 10), the costs of several algorithms do not differ much. After being pre-trained for the corresponding scale, the algorithm proposed in this paper can make decisions within a short time. Among other algorithms, the Ant Colony Optimization (ACO) algorithm also has a relatively short time. The hierarchical task-based deep reinforcement learning HRLPPO algorithm proposed in this paper outperforms other algorithms. As the size of the rescue teams and the number of earthquake-stricken points increase, the Genetic Algorithm (GA), the ACO algorithm, and the GA-PSO algorithm all show a significant increase in decision-making time. For the method proposed in this paper, after pre-training, the overall increase in time cost for different scale increases is not significant. The time cost of the ACO algorithm also increases from 9.91 seconds to 11.04 seconds, and the GA-PSO algorithm, which has the worst time efficiency, increases from 487.43 seconds to 776.99 seconds. Figure 3 shows the satisfaction rates of different methods for rescue demands. In terms of dispatch cost, although the GA-PSO algorithm is occasionally not the highest in cost, it fails to fully meet the rescue demands when the rescue team size is 20 or 40, and its rescue dispatch time is the slowest. The method proposed in this paper can fully meet the rescue satisfaction rate in all three situations, and its time efficiency is also better than that of other methods.

In the early stage of earthquake rescue, the available rescue teams may be less than the rescue demand, and the rescue needs to be completed under the condition of limited reserve of rescue teams. Table 6 shows the example data of different scales under the shortage of high-altitude rescue reserve. Under the shortage of high-altitude rescue, the GA, ACO, GA-PSO algorithms used in the case of sufficient high-altitude rescue teams and the hierarchical task-based reinforcement learning method proposed in this paper after pre-training were used to compare the decision-making effects of the examples. Table 7 shows the experimental results of the dispatch cost and decision-making time of different methods under different scales of rescue teams. Table 6 shows the experimental results of the rescue satisfaction rates of different methods under different scales of rescue teams under the shortage of high-altitude rescue.

In the context of the shortage of high-altitude rescue, by comparing the experimental results of rescue teams of different scales, it can be found that the hierarchical task-based deep reinforcement learning HRLPPO method proposed in this paper shows significant advantages in dispatch cost, decision-making time, and rescue satisfaction rate in the examples after pre-training. For GA and ACO, when dealing with large-scale rescue teams, the cost and decision-making time are relatively high. Especially for ACO, the dispatch cost can be as high as 1063.54 and the decision-making time can be as long as 34.43 seconds when dealing with large-scale rescue. The GA-PSO algorithm performs moderately in small-scale rescue, but its decision-making time increases sharply to 421.91 seconds in large-scale rescue, with extremely low efficiency. Compared with these methods, HRLPPO maintains a relatively low dispatch cost and a fast decision-making time under all scales. Moreover, in terms of the rescue satisfaction rate, whether it is high-altitude or non-high-altitude rescue, it reaches a satisfactory level. Especially in large-scale rescue teams, the high-altitude rescue satisfaction rate of HRLPPO is 80.75%, the non-high-altitude rescue satisfaction rate is 86.67%, and the overall rescue satisfaction rate is 81.68%. The experimental result data are all better than those of other methods. HRLPPO shows obvious advantages in terms of high efficiency and high rescue satisfaction rate. After pre-training, the model has significant advantages in example decision-making for large-scale and urgent high-altitude rescue operations, and is more suitable for the allocation and decision-making of initial rescue operations under the condition of interrupted rescue reserve.

**Table6.** Rescue reserve data under high altitude rescue shortage

| Number of Rescue Teams | Number of Earthquake Points | Number of People Rescued by Rescue Teams | Number of Rescue Demands at Earthquake Points | Number of People Rescued in High Altitude Areas | Number of Rescue Demands at High Altitude Earthquake Points |
|---|---|---|---|---|---|
| 10 | 5 | 346 | 249 | 148 | 207 |
| 20 | 10 | 955 | 475 | 399 | 400 |
| 40 | 15 | 2024 | 642 | 433 | 470 |

**Table 7.** Experimental results of different scale costs under high altitude rescue shortage

| Method | Scheduling Cost | | | Decision Time | | |
|---|---|---|---|---|---|---|
| Numbers of Rescue Teams | 10 | 20 | 40 | 10 | 20 | 40 |
| GA | 280.16 | 548.24 | 934.66 | 22.14 | 48.96 | 53.8 |
| ACO | 401 | 627.72 | 1063.54 | 0.54 | 4.08 | 34.43 |
| AG-PSO | 49.23 | 343.51 | 605.80 | 27.10 | 180.34 | 421.91 |
| HRLPPO | 275.16 | 591.12 | 1024.18 | 0.40 | 0.62 | 0.89 |

**Table 8.** The experimental results of different scale rescue satisfaction rate under high altitude rescue shortage (Team Size of 10)

| Method | Satisfaction Rate for a Team Size of 10 | | |
|---|---|---|---|
| | High Altitude Rescue Satisfaction Rate | Non-High Altitude Rescue Satisfaction Rate | Overall Rescue Satisfaction Rate |
| GA | 71.5 | 100 | 76.31 |
| ACO | 71.5 | 100 | 76.31 |
| GA-PSO | 9.18 | 100 | 24.5 |
| HRLPPO | 71.5 | 100 | 76.31 |

**Table 9.** The experimental results of different scale rescue satisfaction rate under high altitude rescue shortage (Team Size of 20)

| Method | Satisfaction Rate for a Team Size of 20 | | |
|---|---|---|---|
| | High Altitude Rescue Satisfaction Rate | Non-High Altitude Rescue Satisfaction Rate | Overall Rescue Satisfaction Rate |
| GA | 91.5 | 100 | 92.84 |
| ACO | 82.75 | 100 | 85.47 |
| GA-PSO | 48.25 | 100 | 56.42 |
| HRLPPO | 99.75 | 100 | 99.79 |

**Table 10.** The experimental results of different scale rescue satisfaction rate under high altitude rescue shortage (Team Size of 40)

| Method | Satisfaction Rate for a Team Size of 40 | | |
|---|---|---|---|
| | High Altitude Rescue Satisfaction Rate | Non-High Altitude Rescue Satisfaction Rate | Overall Rescue Satisfaction Rate |
| GA | 75.5 | 86.67 | 77.26 |
| ACO | 77.25 | 86.67 | 78.74 |
| GA-PSO | 43 | 86.67 | 49.89 |
| HRLPPO | 80.75 | 86.67 | 81.68 |

## 5. Conclusions

This paper focuses on the research of the rescue dispatch problem of rescue teams in the early stage of earthquake disasters in high-altitude and plateau areas. It comprehensively constructs a model for the earthquake rescue problem of rescue teams by referring to the rescue attributes of rescue teams, the attributes of earthquake-stricken points, the reserve of rescue teams and the rescue demands of earthquake-stricken points. The article adopts the hierarchical task-based deep reinforcement learning method to stratify the rescue tasks. The high-level strategy is responsible for the selection of rescue teams and earthquake-stricken points, while the low-level strategy is responsible for determining the number of dispatched rescuers from rescue teams to earthquake-stricken points. Based on the actual earthquake situation in Luding in 2022, examples of different scales of rescue teams and earthquake-stricken points were generated. Under the conditions of sufficient reserve of high-altitude rescue teams and shortage of reserve of high-altitude rescue teams, the hierarchical task-based deep reinforcement learning algorithm HRLPPO mentioned in the article was compared with GA (Genetic Algorithm), ACO (Ant Colony Optimization) and GA-PSO (Hybrid Genetic Particle Swarm Optimization) through examples, verifying the effectiveness and rationality of the algorithm.

Due to problems such as the continuous change of rescue demands in actual earthquakes and incomplete data collection, the main problems to be solved in the next step are as follows: establish a perfect earthquake rescue

demand platform to collect more comprehensive geological disaster situations, gather more information about rescue teams and earthquake demands, etc. to complete the training of the pre-trained model of deep reinforcement learning and improve the strategic learning of the model. In the future, combined with technical means such as GIS and Amap, a one-stop rescue allocation and planning platform will be completed to improve the more precise digital rescue platform for subsequent natural disasters.

**Author Contributions:** Conceptualization, J.L. and Y.W.; methodology, X.L.; software, C.H.; validation, J.L., S.C., Y.W. and C.H.; formal analysis, S.C.; investigation, J.L.; resources, Y.W. and S.C.; data curation, J.L. and Y.W.; writing—original draft preparation, J.L. X.L. and S.C.; writing—review and editing, J.L. and X.L.; supervision, S.C. All authors have read and agreed to the published version of the manuscript

**Funding:** This work was funded by the National Key R&D Program of China, grant number(2022YFC3004405).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

**Conflicts of Interest:** The original contributions presented in the study are included in the

article, further inquiries can be directed to the corresponding author

**References**

[1] FAN, Y.; FEI, Z.; HUANG, J.; et al. Reinforcement Learn-ing-Based Resource Allocation for Multiple Vehicles with Communication-Assisted Sensing Mechanism[J]. Elec-tronics, 2024, 13(13), 2442.

[2] ZHONG, L.; YE, T.; YANG, Y.; et al. Deep Reinforcement Learning-Based Joint Low-Carbon Optimization for Us-er-Side Shared Energy Storage–Distribution Networks[J]. Processes, 2024, 12(9), 1791.

[3] MIDDELHUISA, J.; LO BIANCOA, R.; SCHERZERB, E.; et al. Learning Policies for Resource Allocation in Business Processes[J/OL]. arXiv:2304.09970v2, 2024 (accessed on 05 July 2024). https://arxiv.org/pdf/2304.09970v2.pdf.

[4] WANG, Y.; CHEN, X.; WANG, L. Deep Reinforcement Learning-Based Rescue Resource Distribution Scheduling of Storm Surge Inundation Emergency Logistics[J]. IEEE Transactions on Industrial Informatics, 2023, 19(10), 10004-10013.

[5] CHU, X.; ZHONG, Q. Post-earthquake Allocation Ap-proach of Medical Rescue Teams[J]. Natural Hazards, 2015, 79(3), 1763-1782.

[6] Kool, W.; Van Hoof, H.; Welling, M. Attention, Learn to Solve Routing Problems! In Proceedings of the 7th In-ternational Conference on Learning Representations (ICLR 2019), New Orleans, LA, USA, 6-9 May 2019.

[7] Ma, Z.P.; Jiao, H.Y.; Zhang, Z.; et al. Research on vehicle routing optimization algorithm for urban lo-gistics distribution [J/OL]. Journal of System Simulation, 1-10 (accessed on 09 November 2024). https://doi.org/10.16182/j.issn1004731x.joss.24-0639.

[8] Zhao, Z.L.; Wu, P.; Sun, S.L.; et al. Research on emergency rescue scheduling optimization considering limited resources under major forest fires [J/OL]. System engineering theory and practice, 1-21 (accessed on 09 November 2024).

[9] Zou, S.S.; Yang, R.; Yang, W. Emergency logistics vehicle routing optimization under urban flood disaster [J]. Agricultural equipment and vehicle engineering, 2024, 62(09), 154-159 + 177.

[10] Jia, T.T.; Zhao, J.H.; Ke, Z.Q.; et al. Metro emer-gency rescue and scheduling optimization for torrential rain [J]. Transportation technology and economy, 2024, 26(05), 10-17.

[11] Sun, Y.; Sun, G.H.; Li, M.; et al. Route optimization of multimodal transport of emergency supplies under inter-val fuzzy environment [J/OL]. Comprehensive trans-portation, 1-9 (accessed on 09 November 2024). https://doi.org/10.20164/j.cnki.cn11-1197/u.20240921.001.

[12] Lou, Z.B.; Peng, Y.; Xin, K. Research on path planning system based on improved Q-Learning algorithm [J]. Computer and digital engineering, 2024, 52(08), 2312-2316.

[13] Tang, H.W.; Gao, F.K.; Deng, J.X.; et al. Research on path planning of material delivery trolley based on ant colony algorithm [J]. Modern Manufacturing Engineering, 2024, (02), 24-30 + 119.

[14] Lyu, C.; Li, M.C.; Ou, J.J. UAV hybrid path plan-ning based on hierarchical deep reinforcement learning [J/OL]. Journal of Beijing University of Aeronautics and Astronautics, 1-13 (accessed on 09 November 2024). https://doi.org/10.13700/j.bh.1001-5965.2023.0550.

[15] Kong, L.; Zhang, G.F.; Su, Z.P.; et al. Ambulance emergency rescue path planning based on improved ant colony algorithm [J]. Computer Engineering and Appli-cation, 2018, 54(13), 153-159.

[16] Deng, D.J.; Ma, Y.H.; Gong, J.; et al. Multi-UAV cooperative mission planning based on parallel GAPSO algorithm [J]. Electro-optics and control, 2016, 23(11), 18-22.