Deep Q networks-based optimization of emergency resource scheduling for urban public health events

Xianli Zhao[1] · Guixin Wang[2]

Abstract

In today's severe situation of the global new crown virus raging, there are still efficiency problems in emergency resource scheduling, and there are still deficiencies in rescue standards. For the happiness and well-being of people's lives, adhering to the principle of a community with a shared future for mankind, the emergency resource scheduling system for urban public health emergencies needs to be improved and perfected. This paper mainly studies the optimization model of urban emergency resource scheduling, which uses the deep reinforcement learning algorithm to build the emergency resource distribution system framework, and uses the Deep Q Network path planning algorithm to optimize the system, to achieve the purpose of optimizing and upgrading the efficient scheduling of emergency resources in the city. Finally, through simulation experiments, it is concluded that the deep learning algorithm studied is helpful to the emergency resource scheduling optimization system. However, with the gradual development of deep learning, some of its disadvantages are becoming increasingly obvious. An obvious flaw is that building a deep learning-based model generally requires a lot of CPU computing resources, making the cost too high.

Keywords Deep learning · Emergency resource scheduling · Deep reinforcement learning algorithm · Path planning · Deep Q network algorithm

1 Introduction

Since the outbreak and spread of the global epidemic, due to the rapid spread of the new coronavirus, the long incubation period, and the variability, urban public health emergencies have presented a complex and changeable situation. To deal with the complex situation of urban emergencies, the problem of emergency resource scheduling for public health events needs to be solved urgently. As a technology that has been widely developed in various industries today, deep learning has many small applications in real life. Its existing and mature algorithm mechanism is of great help to the study of urban public health emergency resource scheduling optimization. Today's development and applications of deep learning include Microsoft's Skype real-time voice translation + translation function, Moodstocks' smartphone image recognition function, autonomous driving technology, Baidu's facial tracking technology, brain tumor detection technology, and fluid models created with convolutional networks, etc. However, in the face of urban public health emergencies, emergency resource scheduling, to maximize the benefits in terms of time and space, not only from the four perspectives of path planning, emergency supply points, demand points, and demand priority to build a model, it is necessary to select a highly feasible optimization algorithm related to deep learning, to achieve resource allocation in the shortest time with the fastest

✉ Guixin Wang
wgx@wzbc.edu.cn

Xianli Zhao
zxli820@163.com

1 Research Center for Transformation and Development of Resource Exhausted Cities, Hubei Normal University, Huangshi 435002, Hubei, China

2 School of Information Engineering, Wenzhou Business College University, Wenzhou 325035, Zhejiang, China

transportation speed to ensure that demand points can obtain emergency supply points. This paper conforms to the severe epidemic development situation of the current era and studies the optimization plan of emergency resource scheduling for urban public health emergencies. It uses the deep reinforcement learning algorithm to construct the emergency resource scheduling optimization system model, to achieve the most efficient distribution of emergency resources in emergencies.

The innovation of this paper is reflected as follows: (1) In the urban emergency resource scheduling optimization system based on deep learning, this paper combines the advantages of deep reinforcement learning algorithm with deep learning in the emergency resource distribution unit path planning scheme and abandons the traditional path planning algorithm. A balance between path quality and speed is achieved. (2) In the modeling of the system algorithm, the flexible use of discrete knowledge to reduce the dimensionality of the three-dimensional space model is more conducive to the effective use of the deep reinforcement learning path algorithm. The research on the optimization of emergency resource scheduling in this paper plays a key role in the rescue measures of urban public health emergencies and has made important contributions to the prevention and control of such emergencies.

2 Related work

The research and investigation on deep learning and resource scheduling in various industries in recent years show that many scholars have achieved rich research results. Polvara R studies a deep reinforcement learning-based end-to-end regulation technique for landing drones with visual markers located on the USV deck. The proposed solution consists of a hierarchical deep Q-network (DQN) used as a high-level navigation strategy to address the two phases of flight: marker detection and descent maneuver. And the robustness of the proposed algorithm to different disturbances acting on the ship is proved by simulation studies, and the obtained performance is comparable to the state-of-the-art methods based on template matching [1]. Another team of scholars has developed a protein residue contact prediction system based on deep learning and massive statistical features of multiple sequence alignments [2]. Ojugo created a predictive and intelligent decision support model for the diabetes pandemic using deep reinforcement learning algorithms [3]. Park implemented a recognition system based on the deep learning algorithm to estimate real intent based on the user's gaze direction [4]. Wang and Srikantha propose a deep learning-based construction model for a novel stealth black-box attack that performs non-intrusive load monitoring on data used in smart meters. Electric utilities and third-party entities such as smart home management solution providers gain important insights into these datasets through machine learning (ML) models. They are then used to perform active or passive power demand management to promote economical and sustainable electricity use [5]. Chen LL uses deep learning methods to simulate the moisture content of high-speed railway subgrade materials. Based on the moisture content of two subgrade materials in a winter-spring cycle in recent years, his research proposes a long short-term memory (LSTM) model, and the reliability and practicability of the LSTM model are proved by comparing the model and its detection data through experiments. The model provides a new method for long-term moisture prediction of high-speed railway subgrade materials in cold regions; simulating and predicting moisture transport plays an important role in analyzing the thermal and hydraulic conditions of subgrades in cold regions [6]. Matthew applies deep learning to the layering of hidden variables, constructs a nonlinear high-dimensional predictor, and develops and trains it for spatiotemporal modeling based on deep learning architectures. He trained the architectural depth through stochastic gradient descent and dropout with parameter regularization to minimize out-of-sample prediction mean squared error. Finally, based on deep learning algorithms, spatiotemporal modeling with dynamic traffic flow and high-frequency trading functions is realized [7]. Lau G proposed an online path planning algorithm for unmanned vehicles responsible for automated border patrols. Based on the goal guarding problem of Isaacs, extended to the scenario where there are multiple fugitives, a fast exploration method based on random tree (RRT) path planning is proposed [8]. Puente-Castro A studied the problem of path planning applied to groups of UAVs. The use of swarms can speed up flight times, thereby reducing operating costs, and when combined with artificial intelligence (AI) algorithms, a single system or operator can control all aircraft. At the same time, the optimal path for each aircraft can be calculated, outlining the trend of using artificial intelligence algorithms to solve path planning problems in UAV swarms [9]. To solve the consistency problem in local path planning, Chen Q extended the traditional definition of topological graph and introduced a new concept of local cognitive graph (LCM). Based on LCM, the consistency of local path planning can be guaranteed by keeping the relationship between the selected route and key obstacles as consistent as possible in successive planning cycles. He devised an iterative decomposition method to generate LCMs. To evaluate the candidate routes in LCM, a vehicle-road evaluation method based on model predictive control (MPC) with vehicle dynamics is combined. He chooses the

best route based on MPC simulation results and criteria such as time spent and route width. The final path followed by the vehicle is also realized by simulation results, verifying the feasibility of the proposed method [10]. Obermeyer studied a sampling-based approach to path planning for visual reconnaissance UAVs, the path planning problem of a single fixed-wing aircraft using one or more electro-optical cameras for reconnaissance missions [11]. Guo studied an improved optimization algorithm based on the artificial potential field method and extended the application of this algorithm to three-dimensional space. It overcame the limited problem of UAV in three-dimensional space and realized the 3D route planning model of UAV flight more optimally [12]. Jungtae proposed a new single-query path planning algorithm that works well in high-dimensional configuration spaces [13]. Chen X studied a path planning algorithm for UAVs to avoid static obstacles or dynamic obstacles at any time [14]. However, these scholars did not elaborate on the combination of deep learning and resource scheduling, but only analyzed its significance unilaterally.

3 Design of optimization model for urban emergency resource scheduling

Problem description: When a city encounters a public health emergency, the things to consider are as follows: A distribution center for emergency resources is established around the disaster-stricken area, which is convenient to deal with the shortage of various resources required by the disaster-stricken area, and the resources stored in the warehouse of the distribution center can be urgently distributed to the surrounding areas of disaster-stricken demand in a timely manner [15, 16]. After a public health emergency occurs, it is necessary to take advantage of the powerful advantages of current network information resources, actively search for and integrate available logistics and transportation resources, and reasonably arrange vehicles to deliver emergency supplies to various demand points according to the priority of the demand gap.

3.1 Basic assumptions

(1) The demand point or the transportation hub in the area close to the demand point should be selected as the emergency resource distribution center.

(2) The route between the supply point and the distribution center is feasible, and the traffic flow is stable. Regardless of the chain reaction of other major natural disasters, the urban climate conditions and the development situation of public health emergencies are relatively stable.

(3) Each emergency resource distribution center does not need to supply each other, but it can distribute emergency supplies to multiple different rescue demand points at the same time.

(4) All materials in the emergency resource distribution center can be used for the rescue and distribution of demand points to ensure that the gap of rescue resources required by all demand points is filled.

(5) The scheduling rule of the emergency resource distribution center is to prioritize the demand point with the largest rescue demand gap and the highest emergency degree, and so on. When the demand gap and urgency are the same or not very different, they can be delivered according to the rescue route.

3.2 Handling of uncertainties

(1) How to deal with uncertain emergency resource demand? Usually, the emergency resource demand required by the demand point can be obtained through model evaluation or direct declaration. At this time, the demand is often described in language with estimation, such as "about n kilograms" or "between n kilograms and m kilograms." We can use the thinking mode of Wu, Y, T to process these data fuzzy [17] and use the fuzzy number $\widetilde{A} = (r_{1i}, r_{2i}, r_{3i})$ in the trigonometric function model to represent the emergency resource demand of demand point i. The trigonometric function is shown in Fig. 1, where r_{1i} and r_{3i} represent the left and right boundaries of the fuzzy number, and r_{2i} represents the preference amount, that is, the actual demand and actual declaration amount of emergency resources. This function expression is

$$f_{\widetilde{A}i}(r_i) = \begin{cases} 0 & , r_i < r_{1i} \\ \dfrac{r_i - r_{1i}}{r_{2i} - r_{1i}} & , r_{1i} \le r_i < r_{2i} \\ \dfrac{r_i - r_{2i}}{r_{3i} - r_{2i}} & , r_{2i} \le r_i < r_{3i} \\ 0 & , r_i \ge r_{3i} \end{cases} \tag{1}$$
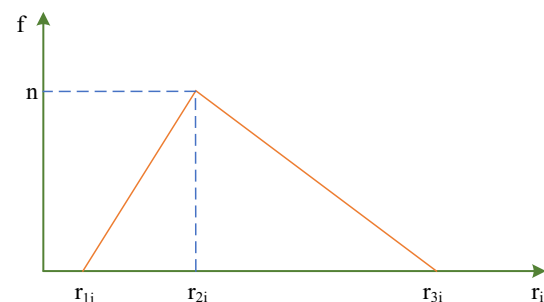

Fig. 1 Trigonometric fuzzy function of demand

(2) The processing method of the vehicle speed of transporting logistics. According to Khanchehzar-rin's speed-dependent function model [18], the change of vehicle speed is a continuous and uniform change, rather than a jumping change. This driving speed dependence function is shown in Fig. 2 , the abscissa represents the time of 24 h a day, and the ordinate represents the speed of the logistics vehicle. According to common sense, the speed of the vehicle will reach two peaks in the morning and evening, and the speed will increase temporarily in the period of noon. Knowing the morning and evening peak driving speeds of the transportation route, the integral method can theoretically be used to calculate the time spent in logistics transportation.

The transportation cost of logistics vehicles may not be considered in the emergency of urban public health emergencies [19, 20]. Therefore, the construction of the resource scheduling optimization model should take the shortest possible total time of logistics and transportation and maximize the resource demand of each demand point in the city for the distribution of emergency materials. According to this main objective, the objective function of emergency resource scheduling can be described as:

$$\min \mu_1 = \sum_{k=1}^{K} \sum_{i=0}^{N} \sum_{j=0}^{N} \left(t_{ij} x_{xjk} \right) + \sum_{i=1}^{N} T_i \quad (2)$$

$$\min \mu_2 = 1 - \frac{1}{N} \sum_{i=1}^{N} f_{\tilde{A}i}(r_i) \quad (3)$$

The constraints of this functional model are as follows:

$$\sum_{i=1}^{N} r_i y_{ik} \leq Q, \quad \forall k = 1, 2, \ldots, K \quad (4)$$

$$LT_i^u \leq ET_j^{u+1}, \quad i = 1, 2, \ldots, N; \quad j = 1, 2, \ldots, N; \\ u = 1, 2, 3, 4 \quad (5)$$

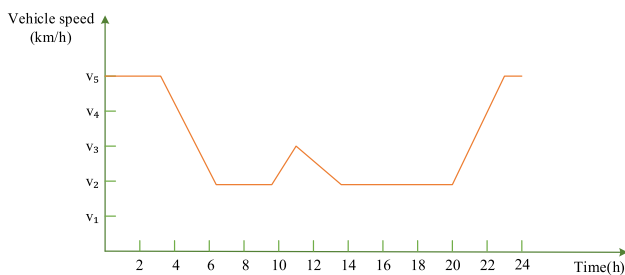$$\sum_{k=1}^{K} y_{ik} = 1, \quad i = 1, 2, \ldots N \quad (6)$$

Fig. 2 Continuous driving speed-dependent function

$$\sum_{i=0}^{N} x_{ijk} - \sum_{j=0}^{N} x_{jik} = 0, \quad \forall k = 1, 2, \ldots, K \quad (7)$$

$$\sum_{i=0}^{N} x_{ijk} = y_{ik}, i = 1, 2, \ldots N, \quad \forall k = 1, 2, \ldots, K \quad (8)$$

$$\sum_{i=0}^{N} x_{ijk} = y_{jk}, \quad j = 1, 2, \ldots N, \quad \forall k = 1, 2, \ldots, K \quad (9)$$

$$\sum_{i \in B} \sum_{j \in B} x_{ijk} \leq |B| - 1, \\ B \subseteq \{0, 1, \ldots, N\}, 2 \leq |B| \leq N - 1; \quad \forall k \quad (10)$$

$$r_{1i} \leq r_i \leq r_{3i} \quad (11)$$

$$y_{ik}, x_{ijk} \in \{0, 1\} \quad (12)$$

Explanation of the parameters of this function model:

N is the number of emergency demand points in the city, where i, j represent the number of demand points, and when $i, j = 0$, it represents the emergency material distribution center;

Q is the maximum number of logistics vehicles;

T_i is the delivery time of urban emergency supplies at emergency demand point i;

t_{ij} is the time required for the logistics vehicle to travel between the emergency demand points i and j, which can be calculated according to the distance and speed of the vehicle;

u is the urgency of emergency demand, $u = 1$ means general, $u = 2$ means urgent, $u = 3$ means urgent, $u = 4$ means urgent;

ET_i^u is the same emergency rescue degree of emergency demand point, that is, the earliest emergency rescue time when the u value in the function model is the same;

LT_i^u is the same rescue emergency degree of emergency demand points, that is, the latest emergency rescue time when the u value in the function model is the same;

x_{ijk}, y_{ik}, r_i are a decision variable, and r_i represents the total amount of emergency rescue resources obtained in the actual situation of urban emergency demand point i, that is, the actual unloading volume of logistics vehicles at emergency demand point i.

3.3 Explanation of constraints

Function expression (2) calculates the shortest total time spent by emergency resources dispatching logistics vehicles; function expression (3) calculates the minimum remaining amount after satisfying the demand of urban emergency resources to the greatest extent; expression (4) indicates that in a logistics distribution, the sum of the unloading amount of the k-th vehicle at each emergency point cannot exceed its maximum carrying capacity. Expression (5) represents the

time limit for the distribution of rescue resources between emergency points with different emergency degrees in the city. The emergency point with high urgency is given priority to arrange logistics distribution, that is, the latest emergency rescue time of the emergency point with high emergency degree should be earlier than the earliest emergency rescue time of the emergency point with low emergency degree. Expression (6) means that each emergency point must have a rescue logistics vehicle to provide rescue services, and the planned driving path of the vehicle must reach all emergency points in the city. Other special circumstances can be solved by additional logistics. Expression (7) is a conservation formula for the number of logistics vehicles, which indicates that the number of vehicles entering and leaving an emergency point should be the same. Equations (8) and (9) indicate that if a vehicle k is assigned to a certain distribution route, there must be a vehicle route starting from the distribution center and returning to the distribution center. Equation (10) is to solve the problem of eliminating undesired bifurcations in route planning and to avoid distribution errors caused by the formation of circular routes that do not pass through the emergency distribution center. The function of formula (11) is that, under the reliability level α of the rescue emergency service, the supply of emergency materials meets the specified interval of demand for materials at the emergency point i, that is, the reliability of emergency rescue services $\text{Pos}\{r_{1i} \leq r_i \leq r_{3i}, i = 1, 2, ..., N\} \geq a$. When $\alpha = 100\%$, it is converted to the current expression. In formula (12), $y_{ik} = 1$ indicates that the demand of emergency demand point i is completed by vehicle k, $y_{ik} = 0$ indicates other conditions, $x_{ijk} = 1$ indicates that the vehicle travels from emergency point i to emergency point j, and $x_{ijk} = 0$ indicates other conditions.

4 DQN algorithm design for emergency resource distribution path

In the face of urban public health emergency resource scheduling, this paper should maximize the benefits in terms of time and space. This requires not only the establishment of models from the four perspectives of route planning, emergency supply points, emergency demand points, and demand priority, but also the selection of highly feasible optimization algorithms related to deep learning to achieve resource allocation in the shortest time and the fastest transportation speed to ensure that demand points can obtain emergency supply points.

Deep Reinforcement Learning (DRL) is a product derived and evolved from Deep Learning. It not only has the perception ability of deep learning, but also combines the decision-making ability of reinforcement learning. To improve the effective application of deep reinforcement learning algorithms, it is necessary to map the emergency resource distribution environment of all demand points in the city. Theoretically, the urban distribution environment can be imagined as a closed space, where all logistics vehicles distribute emergency supplies to their assigned demand point units. In this way, the simulation environment of the deep learning algorithm can be defined according to this limited space abstraction, which is more conducive to maintaining the global state of the entire system. When the vehicle performs the next action in the distribution unit of the space, the action decision can be made according to the current information state. In the distribution route planning, we use the Deep Q Network (DQN) algorithm in deep reinforcement learning. This algorithm can use the powerful learning function of the neural network model to convert high-dimensional data into low-dimensional data and then use the table method to perform action learning according to the state until the learning goal of obtaining the maximum reward is achieved.

In the path planning scheme of emergency resource logistics distribution, we can subjectively reduce the three-dimensional space of the environment and then discretize it into a two-dimensional plane grid. And it uses the grid method to construct a grid map on a two-dimensional plane [21], as shown in Fig. 3. In the figure, the unit space where P is located represents the distribution center; the unit space where K is located represents the emergency demand point, that is, the target location of P; the other orange unit spaces represent obstacles. On the map, each grid cell is a moving cell space. Then, the movement of the delivery unit can only move from the grid unit to the grid unit, but not across the grid units. The constructed raster map is a binary map. When the value is 1, it means that there is no obstacle in the space environment, and movement is possible; when the value is 0, it means that the space has obstacles and
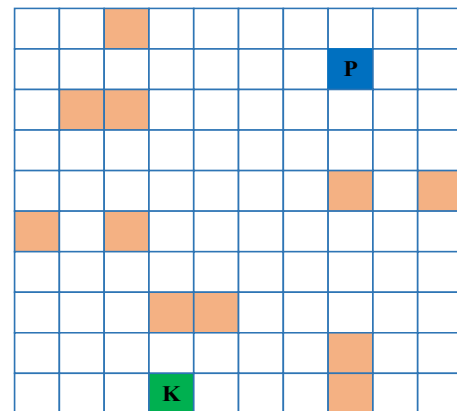

Fig. 3 The raster map

cannot be moved. In the real world, the space is always continuous, and discrete space units cannot accurately and completely correspond to the real space, so it is relatively simple and convenient to use a two-dimensional grid in this way. However, in the modeling of path planning in this paper, this method can retain the most critical environmental core features, which greatly improves the correlation between spatial units and motion decisions. It can be well used for the decision-making behavior of the distribution unit, thus effectively realizing the feasibility of the algorithm in the virtual simulation experiment.

In the DRL algorithm, the reward function plays an important role in its application, which is responsible for guiding the neural network to determine the decision-related factors in the state information. It is used for the next action after selection and extraction, and directly affects the convergence speed and final performance of the algorithm. In this paper, we assume four basic situations for each step of the action decision when the delivery unit moves forward in the path space. According to these four situations, the design of the reward function is shown in Table 1 below, including approaching the destination point, moving away from the destination point, causing collision, and reaching the destination point. And it designs the return values corresponding to the four state conditions, which as shown in the table, for the learning and accumulation of the algorithm until it obtains the maximum reward feedback, and then, the optimal path can be determined.

The steps of multidelivery unit path planning based on deep reinforcement learning algorithm are as follows:

(1) Initialize all parameter data in the space environment;
(2) The distribution units start moving from different starting points to explore the state information of the space environment;
(3) According to the deep reinforcement learning algorithm, the next action is decided. If the state information is extracted to explore, the next action will be randomly generated; if the state information is extracted to use experience, the delivery unit needs to perform reinforcement learning and accumulate

experience according to the environmental state, and then decide the next action;
(4) The environmental state will be updated to a new state following the action of the distribution unit, and the enhanced state signal will be obtained by using the return value of the reward function and loaded into the Deep Q Network (DQN) neural network;
(5) Judging whether the delivery unit has reached the destination point, if it reaches the destination point, the learning process is completed; otherwise, the time step $t = t + 1$ is updated, and the three steps (2), (3) and (4) are repeated;

The entire reinforcement learning behavior will not end until the set index parameter value is reached.

5 Simulation results of emergency resource scheduling system

This paper designs a system architecture for urban emergency resource scheduling optimization based on deep learning. Deep learning has the powerful function of processing a large amount of high-dimensional data and supervision. In the process of the increasingly mature development of neural networks, deep reinforcement learning algorithms with decision-making functions have evolved. In this paper, a deep reinforcement learning algorithm is used to construct the distribution path of urban rescue vehicles. The purpose of virtual simulation experiment design is mainly divided into the following three aspects:

5.1 Experiment 1

In the virtual simulation system, based on the system functions of real-time update of space environment status and effective control of distribution units, taking advantage of its functional advantages of action decision-making, state update, and feedback of optimal path results in the system space, the DQN path planning algorithm is simulated and tested to prove the practicability of the urban emergency resource distribution system constructed in this paper. In the simulated virtual environment, the DQN path planning algorithm is tested, which is divided into a reinforcement learning phase and testing phase.

The experimental environment of the reinforcement learning stage is carried out in the virtual simulation space, and the TensorFlow framework is used in the virtual machine to realize the system programming based on the DQN path planning algorithm [22–24]. When reinforcement learning starts, the delivery unit starts to move from the starting point to the destination point, and the deep

Table 1 Return function

State	Reward
Close to the destination	Return 1
Away from the destination	Return − 1
Collision	Return − 1000
Reach the destination	Return 1000

learning algorithm in the simulation space will update the state of the entire space environment at each step. Then it decides whether to choose "explore" (explore) or "accumulate action experience" (use experience) according to the algorithm strategy. If the spatial environment state parameter data are passed to the neural network to extract the environmental features, then the reward value R can be obtained according to the environmental feature information, and the feedback is passed to the Q value, and the update formula is used for iterative calculation. After each completion, it increases the Q-learning frequency by one. According to whether the frequency reaches 3000, it decides whether to end the study.

During the testing phase, the delivery unit progresses from the origin to the destination, updating the state of the environment at each step. The algorithm model directly decides the next action based on the environmental feature information. After several iterations, the delivery unit chooses the optimal path to the destination.

In the reinforcement learning stage of experiment 1, the convergence process of the reinforcement learning experiment of the DQN algorithm was tested by changing the number of distribution units and the scale of the distribution environment. Its convergence data are shown in Table 2.

According to the experimental data in Table 2, the reinforcement learning convergence curve is drawn. Figure 4 shows the system model composed of distribution units of different scales and the convergence trend diagram during the convergence process of the reinforcement learning experiment. Among them, the abscissa represents the number of delivery units, and the ordinate represents the convergence time of the system algorithm. Curves 1, 2, and 3 represent the trend of the convergence function under different delivery unit tasks, that is, the trend of the convergence trend of delivering 3, 30, and 300 units, respectively.
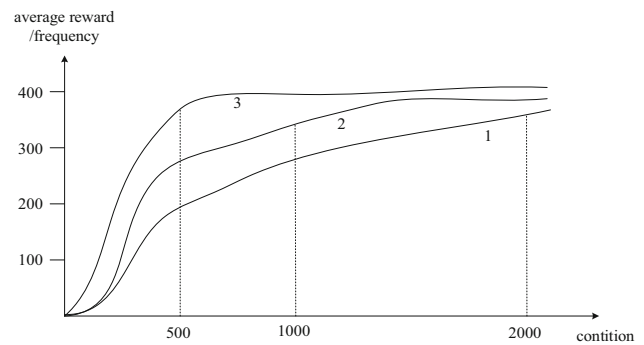

Fig. 4 Training convergence function trend plot

In the reinforcement learning convergence trend graph, experiment, numbers 1, 2, 3, and 4 correspond to curve 1 with the least number of delivery units and the slowest convergence rate. With the increase in the number of delivery units in experiments No. 5, 6, 7, 8 and experiments No. 9, 10, 11, and 12, the convergence speed of the whole system is also improved. This is because the DQN algorithm is a process of gradual learning by constantly exploring new situations. When the scale of the environment is the same, the fewer the number of delivery units, the lower the probability of discovering new situations through random exploration, making it difficult for the system to learn more about the environment state knowledge; as the number of distribution units increases, the probability of the distribution units encountering each other is greater, and more environmental states can be learned more quickly, resulting in a significant increase in the convergence speed. After testing the convergence speed of a pair of system models, the feasibility of the urban emergency resource distribution system based on deep learning is proved.

Table 2 Convergence data sheet	Experiment number	Number of delivery units	Delivery environment size	Convergence iterations
	1	3	10*10	512
	2	3	20*20	868
	3	3	30*30	1790
	4	3	40*40	2408
	5	30	10*10	398
	6	30	20*20	686
	7	30	30*30	1204
	8	30	40*40	2081
	9	300	10*10	116
	10	300	20*20	299
	11	300	30*30	909
	12	300	40*40	1109

5.2 Experiment 2

On the basis of the system framework, the same virtual simulation space environment is guaranteed, and different path planning algorithms are used and compared, and the sensitivity of the DQN path planning algorithm in the urban emergency resource distribution system is proved through performance analysis. In this paper, the basis of using the grid method is to construct a map model of the virtual simulation space, and the influence of other interference factors is not considered in the experiment. Three different path planning algorithm models are designed, including Genetic Algorithm (GA) [25], Ant Algorithm (ACO) [26], and Deep Q Network Algorithm (DQN). In the same virtual simulation environment, by changing the number of distribution units, the amount of distribution tasks completed in unit time R is calculated, and the value of R is used to compare the performance of the algorithm. It is also possible to assign the same distribution task, count the number of completed path planning of the corresponding algorithm model, and compare the efficiency of different path planning algorithms, and the amount of distribution tasks completed per unit time R reflects the efficiency of the entire system. Therefore, the amount of delivery tasks R completed per unit time reflects the efficiency of the entire system. In the experiment, we assign the same delivery task, count the number of completed path planning of the corresponding algorithm model, and compare the efficiency of different path planning algorithms. The calculation method is as follows:

Assuming that the system takes t as the cycle, for each distribution unit r_i, count the number of distribution tasks n_i completed in the cycle,

Then, all distribution units complete the sub-task $\sum_i r_i$ in the time t, so there are $R = \sum_i r_i / t$.

Figure 5 shows the efficiency test results of three different path planning algorithms when the same distribution task is assigned under the same simulation system environment (scale is 30*30). The experimental comparison
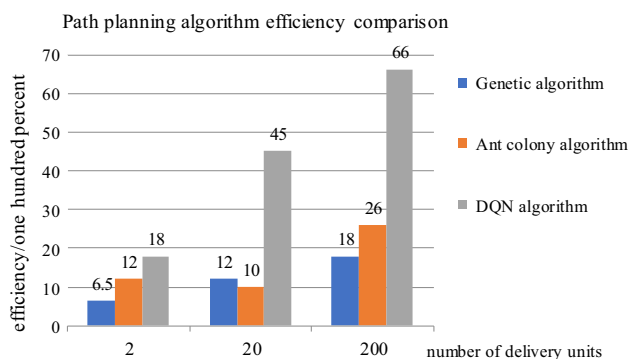
shows that, with the increase in the number of distribution units, the system of the DQN algorithm is more sensitive, which proves that the distribution unit path planning ability of the DQN algorithm system is stronger and more efficient.

5.3 Experiment 3

The performance stress test is carried out in the virtual simulation system [27]. In order to increase the load pressure of the whole system, the number of distribution units can be continuously increased to verify the computing power and load limit of the system. The experiment in this article uses four virtual machines (VMware), and the test starts with the first one VMware. As the number of distribution units increases, the system load gradually increases while maintaining the operation and computation of each distribution unit, and VMware's resource surplus is gradually reduced. When the threshold value is reached, VMware is expanded to trigger the elastic scaling of system resources. The experimental system implements the "change to 1" strategy. When expanding VMware, only one unit is expanded at a time; when recycling, only 1 unit is recycled at a time. Under normal circumstances, the CPU is the core computing resource of the system, and the IO network is the network resource. With the increase in the system load in the experiment, these two resources will limit the experiment to the bottleneck stage. Therefore, to simplify the system performance indicators and consider the impact of the two on performance, this paper uses weighting to obtain the target resource and then defines the system performance pressure measurement indicator as the unit resource remaining performance. The specific method is as follows:

$$resource = \frac{CPU}{2} + \frac{IO}{2} \tag{13}$$

$$performance = \frac{resource}{VM\,ware} \times 100\% \tag{14}$$

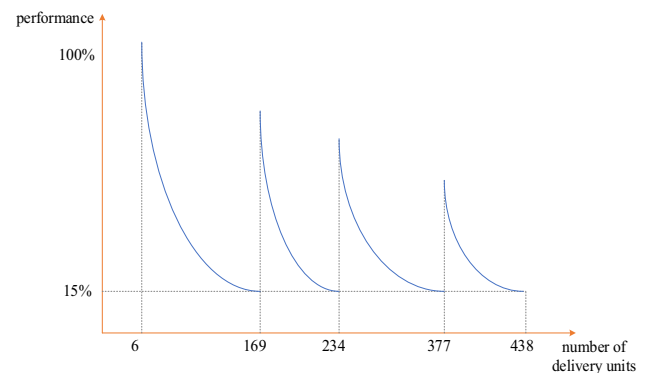Fig. 5 Path planning algorithm efficiency comparison

Fig. 6 System performance test

Figure 6 is the performance stress test result after the system is tested and calculated. The abscissa is the number of distribution units, and the ordinate is the resource remaining rate per unit of VMware. By dynamically tracking the performance of this indicator, you can clearly observe how the system elastically expands according to the load pressure. Among them, the four curves represent the performance curve of 1 VMware, the performance curve of 2 VMware, the performance curve of 3 VMware, and the performance curve of 4 VMware. The experimental results show that the maximum concurrency of a single VMware is 169, the maximum concurrency is 234 when it is dynamically increased to 2, the maximum is 377 when it is dynamically increased to 3, and the maximum is 438 when it is dynamically increased to 4. In theory, assuming that the performance of one VMware can support Q distribution units, then N VMwares with the same performance can support N * Q distribution units. However, the above experimental results are based on dynamic expansion, there is some performance loss, and the CPU and IO resources cannot be perfectly coordinated. Therefore, with the expansion of VMware, the entire system will reach the elastic expansion boundary faster, which can be optimized through better resource scheduling algorithms and elastic expansion strategies to approximate the theoretical effect. Ultimately, this experiment confirms that the system of urban emergency resource scheduling optimization based on deep learning has good application value. Finally, the load test of the system in experiment three makes us feel the load capacity of the system more intuitively, which can be well applied to the actual situation in emergency rescue. Therefore, this experiment confirms that the system of urban emergency resource scheduling optimization based on deep learning has good application value.

5.4 Discussion

The three experiments designed in this paper are, respectively, Experiment 1. The simulation modeling experiment of the deep reinforcement learning DQN algorithm is carried out in the system to verify the feasibility of the system. Experiment 2 compares the genetic algorithm, ant colony algorithm, and DQN algorithm to solve the path planning problem of multiintelligent distribution units in the system simulation experiment. It is found that the deep reinforcement learning DQN algorithm has a significant effect in the simulation experiment and has obvious advantages in path planning. In the third experiment, by simulating the distribution tasks in the actual environment, in the virtual simulation environment, the number of distribution units is continuously increased to increase the system load for stress testing. The experiment verifies that the urban emergency resource scheduling optimization system based on deep learning has superior performance.

6 Conclusion

For the research on the optimal scheduling of emergency resources for urban public emergencies, this paper optimizes and upgrades the urban emergency resource scheduling scheme mainly from the aspect of path planning for the distribution of emergency resources. And through research to build an emergency resource distribution model, the optimal algorithm, that is, a deep reinforcement learning algorithm, is selected for path planning. Finally, compared with other commonly used path planning algorithms, the system simulation experiment proves that the urban emergency resource scheduling optimization scheme based on deep learning is effective and feasible.

Declarations

Conflict of interest The author(s) declared no potential conflicts of interest with respect to the research, author- ship, and/or publication of this article.

References

1. Polvara R, Sharma S, Wan J, Manning A, Sutton R (2019) Autonomous vehicular landings on the deck of an unmanned surface vehicle using deep reinforcement learning. Robotica 37(11):1867–1882
2. Zhang H, Hao M, Wu H, Ting HF, Tang Y, Xi W et al (2022) Protein residue contact prediction based on deep learning and massive statistical features from multi-sequence alignment. Tsinghua Sci Technol 27(5):843–854
3. Ojugo AA, Ekurume E (2021) Predictive intelligent decision support model in forecasting of the diabetes pandemic using a reinforcement deep learning approach. Int J Educ Manag Eng 11(2):40–48
4. Park HJ, Jang I, Ko KE (2021) Meal intention recognition system based on gaze direction estimation using deep learning for the user of meal assistant robot. J Inst Control 27(5):334–341
5. Wang J, Srikantha P (2021) Stealthy black-box attacks on deep learning non-intrusive load monitoring models. IEEE Trans Smart Grid 12(4):3479–3492

6. Chen LL, Chen J, Wang C (2021) Modeling of moisture content of subgrade materials in high-speed railway using a deep learning method. Adv Mater Sci Eng 2021(4):1–9

7. Dixon MF, Polson NG et al (2019) Deep learning for spatio-temporal modeling: dynamic traffic flows and high frequency trading. Appl Stoch Models Bus Ind 35(3):788–807

8. Lau G, Liu HHT (2013) Real-time path planning algorithm for autonomous border patrol: design, simulation, and experimentation. J Intell Rob Syst 75(3–4):517–539

9. Puente-Castro A, Rivero D, Pazos A (2022) A review of artificial intelligence applied to path planning in UAV swarms. Neural Comput Appl 34(1):153–170

10. Chen Q, Lu Y, Wang Y (2021) From topological map to local cognitive map: a new opportunity of local path planning. Intell Serv Robot 14(2):285–301

11. Obermeyer KJ, Oberlin P, Darbha S (2015) Sampling-based path planning for a visual reconnaissance unmanned air vehicle. J Guid Control Dyn 35(2):619–631

12. Guo Y, Liu X, Zhang W, Yang Y (2020) 3D path planning method for UAV based on improved artificial potential field. Xibei Gongye Daxue Xuebao/J Northwest Polytech Univ 38(5):977–986

13. Jungtae K, Daijin K (2010) Single-query probabilistic roadmap planning algorithm using remembering exploration method. J KIISE Comput Pract Lett 16(4):487–491

14. Chen X, Zhao M, Yin L (2020) Dynamic path planning of the UAV avoiding static and moving obstacles. J Intell Rob Syst 99(3):909–931

15. Farouk A, Alahmadi A, Ghose S, Mashatan A (2020) Blockchain platform for industrial healthcare: vision and future opportunities. Comput Commun 154:223–235

16. Shankar K, Zhang Y, Liu Y, Wu L, Chen CH (2020) Hyperparameter tuning deep learning for diabetic retinopathy fundus image classification. IEEE Access. https://doi.org/10.1109/ACCESS.2020.3005152

17. Wu YT, Lan HY, Liu CJ (2021) On implicit coupled systems of fuzzy fractional delay differential equations with triangular fuzzy functions. AIMS Math 6(4):3741–3760

18. Khanchehzarrin S, Shahmizad M, Mahdavi I, Mahdavi-Amiri N, Ghasemi P (2021) A model for the time dependent vehicle routing problem with time windows under traffic conditions with intelligent travel times. RAIRO Oper Res 55(4):2203–2222

19. Chen Y, Zheng W, Li W, Huang Y (2020) The robustness and sustainability of port logistics systems for emergency supplies from overseas. J Adv Transp. https://doi.org/10.1155/2020/8868533

20. Yin L, Pan W, Kuang J, Zhuang M (2019) Application of boot-strap-DEA with fuzzy computing in performance evaluation of forklift leasing supplier. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2959598

21. Ren L (2021) A novel raster map exchange scheme based on visual cryptography. Adv Multimed 2021(10):1–7

22. Nour M (2021) Real-time detection and motivation of eating activity in elderly people with dementia using pose estimation with TensorFlow and OpenCV. Adv Soc Sci Res J 8(3):28–34

23. Lv Z, Qiao L (2020) Deep belief network and linear perceptron based cognitive computing for collaborative robots. Appl Soft Comput 20:106300

24. Lv Z, Song H (2019) Mobile internet of things under data physical fusion technology. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2019.2954588

25. Peng D, Tan G, Fang K, Chen L, Zhang Y (2021) Multiobjective optimization of an off-road vehicle suspension parameter through a genetic algorithm based on the particle swarm optimization. Math Probl Eng 2021(9):1–14

26. Wan L, Du C (2021) An approach to evaluation of environmental benefits for ecological mining areas based on ant Colony algorithm. Earth Sci Inf 14(2):797–808

27. Hong T, Zhao W, Liu R, Kadoch M (2020) Space-air-ground IoT network and related key technologies. IEEE Wirel Commun 27:96–104