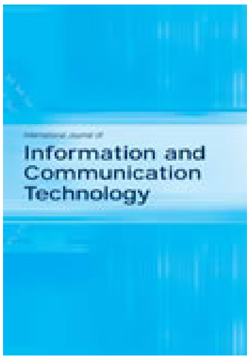


International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

DDQN-GA: a hybrid algorithm for intelligent inspection path optimisation and process modelling in power systems

Qianqiao Zhao, Xuanyu Song, Junyu Liu, Changyu Li, Yan Guan, Wenjie Fan

DOI: 10.1504/IJICT.2025.10071013

Article History:

Received:	01 November 2024
Last revised:	08 February 2025
Accepted:	12 February 2025
Published online:	27 May 2025

DDQN-GA: a hybrid algorithm for intelligent inspection path optimisation and process modelling in power systems

Qianqiao Zhao*, Xuanyu Song, Junyu Liu, Changyu Li, Yan Guan and Wenjie Fan

State Grid Liaoning Electric Power Co. Ltd. Marketing Service Center,
Shenyang 110000, China
Email: zhaoqq0802@outlook.com
Email: SongXY@outlook.com
Email: Liujy123@outlook.com
Email: LICHANGY@outlook.com
Email: guanyan552@outlook.com
Email: fanwenj@outlook.com
*Corresponding author

Abstract: As the electricity trading market expands and becomes more complex, ensuring user safety and efficient equipment operation has become a critical challenge for the power industry. Inspection path planning and process modelling, as core technologies in intelligent inspection within the smart manufacturing system, have become essential tools for addressing this challenge. In response to the low efficiency of power system inspections, this paper proposes an intelligent inspection path optimisation and process modelling method (DDQN-GA) based on a combination of double deep Q-Network (DDQN) and genetic algorithm (GA). First, the proposed method employs the DDQN algorithm to intelligently allocate power-trading users and inspection teams, allowing each team to be optimally scheduled based on real-time system status and demand. Subsequently, GA is used to optimise the internal paths of each inspection team, effectively exploring and optimising complex path combinations to minimise overall inspection costs and achieve the optimal inspection plan. Experimental results demonstrate that this method significantly reduces total inspection costs and shortens computation time. Compared with three traditional algorithms, the DDQN-GA approach considerably improves computational efficiency, especially in handling large-scale inspection teams and user allocations.

Keywords: power trading; user inspection; path planning; double deep Q-network; genetic algorithm; GA.

Reference to this paper should be made as follows: Zhao, Q., Song, X., Liu, J., Li, C., Guan, Y. and Fan, W. (2025) 'DDQN-GA: a hybrid algorithm for intelligent inspection path optimisation and process modelling in power systems', *Int. J. Information and Communication Technology*, Vol. 26, No. 14, pp.1–19.

Biographical notes: Qianqiao Zhao holds a Master's degree, and works in Major Customer Service in State Grid Liaoning Electric Power Co., Ltd., to serve the new power installation work for major industrial power users in Liaoning Province.

Xuanyu Song graduated from Northeast Electric Power University with a Master's degree, engaged in the calculation of electricity charges of power generation enterprises and research on the split mode of electricity generated by power generation enterprises under the spot mode.

Junyu Liu graduated from Liaoning University of Engineering and Technology with a Master's degree in Electrical Engineering. He is currently a Senior Engineer and his research focuses on power market operation and electricity metering.

Changyu Li graduated from North China Electric Power University with a Master's in Project Management. He is a Senior Engineer and his research focuses on power market operation and power spot market trading.

Yan Guan is currently serving as a Senior Marketing Specialist at the Marketing Service Center of State Grid Liaoning Electric Power Co., Ltd., with a Deputy Senior Technical Title and Master's in Electrical Engineering.

Wenjie Fan graduated from North China ElectricPower University with a Master's in Electrical Engineering and Automation in 2019. His research direction was power quality in new energy power systems.

1 Introduction

As the electricity trading market continues to expand and grow more complex, ensuring the safety and reliability of power users has become one of the key challenges faced by the power industry (Wang et al., 2024; Liu and Tan, 2017). To address this challenge, maintaining the stable and efficient operation of electrical equipment has become a core task. In this context, inspection path planning and process modelling are not only crucial for ensuring the normal operation of the power system but are also central components of intelligent inspection in the smart manufacturing framework (Yang et al., 2024; Li et al., 2024).

Through scientifically planned inspection routes, inspectors can regularly check user equipment, identify and address potential issues promptly, reduce the failure rate of equipment, and enhance the overall reliability of the power system. At the same time, inspection process modelling helps power companies analyse inspection efficiency and optimise resource allocation. The combination of these two techniques significantly improves the stability and reliability of power supply, providing strong support for the safe operation of the electricity market.

In traditional inspections, path planning often relies on human experience and pre-established routes. However, in modern, intelligent power inspections, path planning is typically driven by advanced algorithms such as reinforcement learning, genetic algorithms (GAs), and heuristic algorithms, which can automatically generate optimal inspection routes. These intelligent algorithms dynamically adjust inspection strategies based on real-time changes in the power system and inspection requirements, maximising inspection efficiency. By integrating path planning and process modelling, power systems can conduct large-scale intelligent inspections of electrical equipment more efficiently while ensuring reliability and safety.

To address the issue of substation inspection scheduling under multi-resource constraints, an improved genetic algorithm (SAGA) was proposed by Xie et al. (2016) so the problem of local optimal solutions in traditional GAs can be effectively solved. An intelligent substation inspection method based on digital twin technology was introduced by Xie et al. (2023), and the improvement in the efficiency and safety of inspection operations has also been verified through simulation experiments. The design and development of an electric power inspection and monitoring system and data preprocessing was explored by Guo et al. (2024), with the aim of improving Cerational efficiency of power systems. The method of studying intelligent inspection path planning for hydropower stations through GA was used by Qin and Fei (2024), it improved the timeliness of inspection path planning and achieved faster convergence speed and stable optimal solutions.

However, traditional inspection path planning methods often suffer from poor accuracy and timeliness in inspection path planning. It results in slow convergence speed and difficulty in dealing with complex and changing practical situations. At the same time, the optimisation of inspection paths also faces many challenges, such as uneven allocation of inspection tasks and unreasonable path planning. These result in low resource utilisation efficiency and high costs.

Reinforcement learning (Shakya et al., 2023) is a machine learning method for solving sequential decision problems. In reinforcement learning, decisions are made by the agent after learning through interaction with the environment. So that rewards can be accumulated to the maximum extent possible. Double deep Q-network (DDQN) (Zhu et al., 2023; Luo et al., 2022; Yin et al., 2024) is a kind of reinforcement learning algorithm. It separates the steps of selecting actions and evaluating actions to reduce overestimation in Q-learning, thereby improving the stability and performance of the strategy.

A patrol path planning method (DDQN-GA) based on the combination of DDQN and GA (Wollmann et al., 2023; Torres et al. 2024; Liu et al. 2022; Du and Li, 2024) is proposed in this paper. This method uses the DDQN algorithm to intelligently allocate power trading users and inspection teams. So as to select appropriate power trading users based on the current state and assign them to the corresponding inspection teams. Then, the GA algorithm was applied to optimise the inspection path within each team, thereby minimising the total cost and finally achieving the overall optimal inspection plan. This study aims to improve inspection efficiency, reduce costs, and optimise resource utilisation, providing robust support for intelligent inspection in power systems. It not only enhances the operational efficiency of the system but also offers a novel solution for the management and maintenance of large-scale electrical equipment.

2 Problem description

2.1 Subsection

m number of users at risk in power trading

n number of participating inspection teams

μ user set, $\mathcal{U} = \{\mathcal{U}_1, ..., \mathcal{U}_m\}$

T inspection team assemble, $\mathcal{T} = \{\mathcal{T}_1, ..., \mathcal{T}_n\}$

i user number for power trading risk, indicate the i^{th} user, $i \in \{1, \dots, m\}$

j number of inspection teams, indicate the j^{th} team, $j \in \{1, \dots, n\}$

x_i represent the horizontal axis of the i^{th} user's position

y_i represent the vertical axis of the i^{th} user's position

dt_j represent the path time coefficient of the j^{th} inspection team

dc_j represent the path cost coefficient of the j^{th} inspection team

wl_j represent the workload of the j^{th} inspection team

S_i represent the start time of the inspection for the i^{th} user

C_i represent the end time of the inspection for the i^{th} user

pt_i represent the standard waiting time of the inspection for the i^{th} user

D_i represent the latest completion inspection for the i^{th} user

op_i represent the overdue penalty for the i^{th} user

rb_i represent the advance completion reward for the i^{th} user

$x_{i,j}$ binary variable, whether team $\mathcal{T}_j$ assigned to power user $\mathcal{U}_i$ for inspection; if so, output 1; otherwise, output 0.

$y_{j,i,i'}$ binary variable, power trading users $\mathcal{U}_i$ and $\mathcal{U}_{i'}$ are both inspected by team $\mathcal{T}_j$; if user $\mathcal{U}_i$ start inspection before user $\mathcal{U}_{i'}$ output 1, otherwise output 0.

2.2 Description of inspection path planning issues

In a period of time, there are m power trading users with different risk levels waiting for the n team with different ability levels to do inspections.

The ability level of the inspection team is divided into three levels, represented by I, II, and III. Teams with different abilities have different path time coefficients and path cost coefficients. Power trading users are categorised based on their risk levels into three distinct groups: high, medium, and low. These risk levels reflect the potential financial impact or instability of the users' trading activities. High-risk users (denoted as H) are typically those involved in large transactions or exposed to market volatility, making their inspections more complex and requiring the highest level of expertise. As such, these users can only be inspected by level I teams, which have the necessary skill set to handle the most intricate and high-stakes inspections. Medium-risk users (denoted as M) are involved in moderately risky activities, with some financial exposure but not to the extent of high-risk users. These users can be inspected by level II teams, which are capable of handling both medium and low-risk users. However, level II teams are not equipped to handle high-risk users. Low-risk users (denoted as L) are those with stable, low-exposure trading patterns. They are the least complex to inspect, and both level II and level III teams can perform these inspections. However, level III teams are limited to inspecting only low-risk users due to their more basic capabilities. In summary, level I team can inspect all risk level power trading users, the level II team can inspect M and L risk level power trading users, and level III team can only inspect L risk level power trading users.

Each power trading user $\mathcal{U}_i$ have different standard waiting worktime pt_i, the latest inspection time D_i, overdue penalty op_i, early completion reward rb_i.

The assumption for this question is as follows:

1 Assuming that each power trading user's inspection task is independent of each other. It means that completing one user's inspection task does not affect the other users' tasks.

2 Assuming that the risk level of power trading users and the attributes of the inspection task do not change during the inspection process.

3 Before starting, all of the power trading users and inspection team are ready.

4 Assume that each team can perform only one inspection task at any time. It means perform multiple tasks at the same time will not be considered.

5 Assuming that the inspection task of each power trading user is continuous. It means that once the inspection starts, the entire task must be completed without interruption.

6 Assuming that each power trading user can only be assigned to one team for inspection. Multiple teams inspect one user at the same time are not be allowed.

7 Assuming no unexpected situations or safety issues along the way and all paths are safe and reliable.

The purpose of this paper is the lowest *cost* of inspection, as shown in formula (1).

$$target = \min(\cos t) \tag{1}$$

$$cost = cost_1 + cost_2 + cost_3 \tag{2}$$

$$cost_1 = \sum_{j=0}^{n} dt_j \times dc_j \times \rho_1 \times d_j \tag{3}$$

$$cost_2 = \sum_{i=0}^{m} \max\left((C_i - D_i), 0\right) \times op_i \tag{4}$$

$$cost_3 = \sum_{i=0}^{m} \min\left((S_i - D_i), 0\right) \times rb_i \tag{5}$$

Among them, cost consists of three parts: path cost, overdue penalty, and early reward, as shown in formula (2). $cost_1$ is path cost, as shown in formula (3), d_j represents the distance of the team $\mathcal{T}_j$ along the inspection path, ρ_1 represents the unit distance of path cost; $cost_2$ is the overdue penalty, if the completion time exceeds D_i, there will be an overdue penalty, as shown in formula (4), $cost_3$ is the advance reward, if the completion time do not exceed D_i, there will be a reward, as shown in formula (5).

The mathematical model constraints of this problem are defined as follows:

$$\sum_{j=1}^{n} x_{i,j} = 1, \quad \forall i \tag{6}$$

$$C_i \geq 0, S_i \geq 0, \quad \forall i \tag{7}$$

$$C_i = S_i + pt_i, \quad \forall i \tag{8}$$

$$y_{j,i,i'} + y_{j,i'i} \leq 1, \quad \forall j, i, i' \tag{9}$$

$$S_i \geq C_{i'} - y_{j,i',i} \times U, \quad \forall j, i \neq i' \tag{10}$$

$$S_{i'} \geq C_i - \left(1 - y_{j,i,i'}\right) \times U, \quad \forall j, i \neq i' \tag{11}$$

$$x_{i,j} \in \{0, 1\}, \quad \forall i, j \tag{12}$$

$$y_{j,i,i'} \in \{0, 1\}, \quad \forall i, i', j \tag{13}$$

Unique allocation constraint: formula (6) indicates that each power trading user can only be assigned to one team for inspection.

Inspection time constraints: formula (7) indicates that the start time and end time of inspection should not less than 0; formula (8) indicates the relationship between the start time and end time of user $\mathcal{U}_i$ inspection.

Inspection task constraints: as shown in formula (9), formula (10), and formula (11). U is a sufficiently large positive integer representing the inspection order of the same team, and each team can only inspect one user at the same time. When $y_{j,i,i'} = 1$, user $\mathcal{U}_i$ starts the inspection before user $\mathcal{U}_{i'}$. If the start time $S_{i'}$ of user $\mathcal{U}_{i'}$, is later than the end time of user $\mathcal{U}_i$, output $S_i' \geq C_i$; otherwise, output $S_i \geq C_{i'}$.

Ability matching constraint: The ability level of the team must match the risk level of the power trading user. If team $\mathcal{T}_j$ is a Level **I** team, then for the $\forall i$, the risk level of power trading users $\mathcal{U}_i$ should be **H** or **M** or **L**; if team $\mathcal{T}_j$ is a level **II** team, then for the $\forall i$, the risk level of power trading users $\mathcal{U}_i$ should be **M** or **L**; if team $\mathcal{T}_j$ is a level **III** team, then for the $\forall i$, the risk level of power trading users $\mathcal{U}_i$ should be **L**.

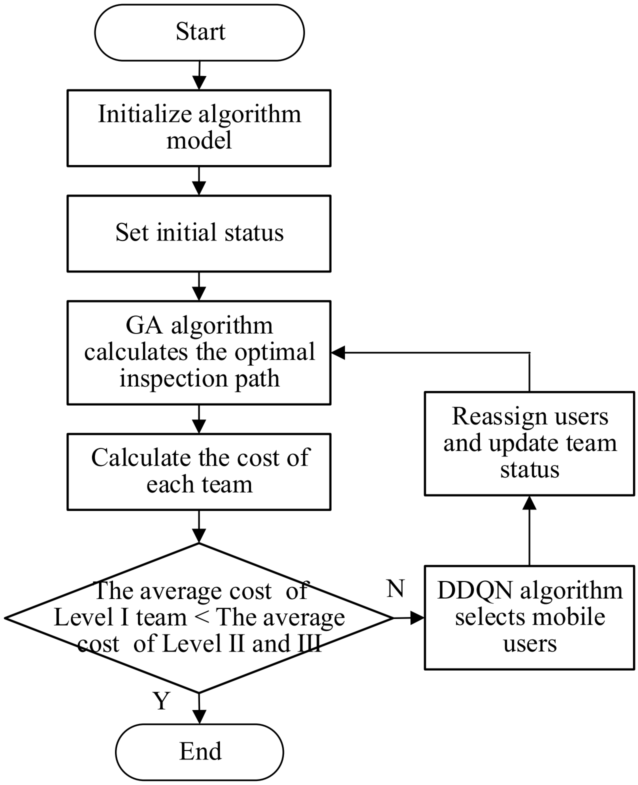
Binary variable constraints are shown in formulas (12) and (13).

3 Algorithm description

3.1 Algorithm flow

The general algorithm flow is shown as Figure 1. Firstly, initialise the parameters of DDQN and GA, and set the initial state according to the initial allocation model. The initial allocation model according to the ability level of the teams and risk level of the users processing the initial allocation; power trading users with risk levels of **H**/**M**/**L** will be assigned to level **I**/**II**/**III** teams for inspection. The initial allocation model provides a reasonable starting point for the subsequent optimisation process and this will make the subsequent optimisation process more efficient.

Figure 1 The general algorithm flow



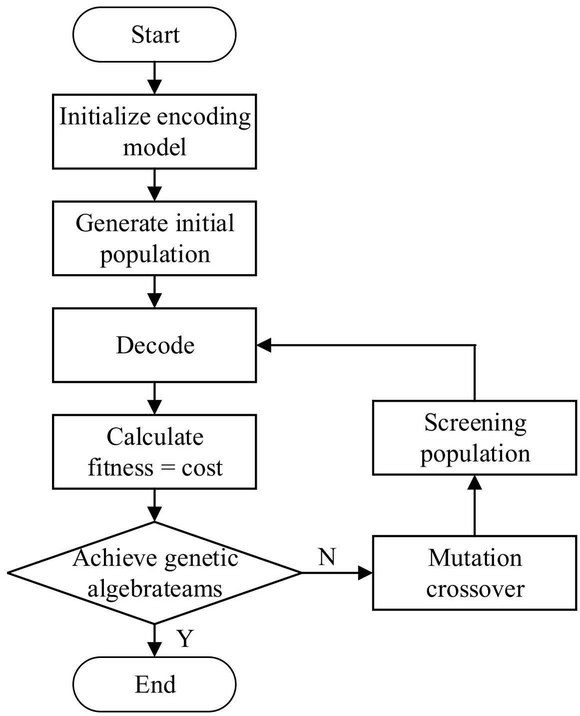
Secondly, the GA algorithm iteratively calculates the best inspection path for each team based on the initial state, and calculates the path cost, overdue penalty, and early reward for each team in the initial state. After the calculation is completed, if the average cost of the level **I** team is greater than the average cost of the levels **II** and **III** teams, the calculation will be stopped, otherwise, the DDQN user selection section will be started.

Finally, DDQN selects and reassigns mobile users based on the state space, updates the team state, and iteratively calculates the optimal inspection path and inspection cost for each team using GA algorithm.

3.2 GA algorithm for solving the optimal inspection path

After determining the task allocation for each team based on the current team state, GA is used to optimise the inspection path for each team. GA is an optimisation method based on natural selection and genetic mechanisms, particularly suitable for solving combinatorial optimisation problems. In this section, GA is used to plan the optimal user inspection path within the team. So, the minimum cost of the team can be found. The GA algorithm flow is shown as Figure 2.

Figure 2 The GA algorithm flow



3.2.1 Encoding and decoding model

Number the user locations that require maintenance tasks and form a oriented path code. The inspection path is 2-3-1-4. Decoding is the inverse process of encoding, which transforms the optimal chromosome into an inspection path and calculates the total cost of the plan. They are shown in formulas (2) to (5). The steps are as follows:

Step 1 Read the chromosome to get the inspection path, the user's location coordinates, the task time and the latest inspection time.

Step 2 Read the first chromosome to get the first inspection point as the starting point, calculate the user's task time, and update the user's inspection start and end time; calculate the costs related to inspection time based on formula (4-5), such as overdue penalties and early rewards.

Step 3 Read the last chromosome to get the next inspection point, calculate the Euclidean distance between this point and the previous point. Then calculate the time required for the team to reach the new user point (multiplied by the path time coefficient) and update the user inspection start time. At last, calculate the user's work time and update the end time of user's inspection. The total cost of inspection can be calculated by formula (2)–(5).

Step 4 Judge if the reading of chromosome is complete. If not, execute Step 3. Otherwise, ending the decode.

3.2.2 Initialise the population

Population initialisation is a key issue in GAs, and the quality of the initial population has a significant impact on search speed and effectiveness. This part will adopt a random search strategy and the steps are as follows:

Step 1 Number the user's locations that require inspection tasks and place them into the candidate pool.

Step 2 Randomly select the serial number from the candidate pool and weed out it from the candidate pool.

Step 3 Judge if the candidate pool is empty. If not, execute step 2. Otherwise, combine the serial number in order to form chromosomes.

3.2.3 Initialise the population

Partially-matched crossover (PMX) is a crossover strategy in GAs. It selects two-point Crossover between two parental chromosomes, swaps the gene fragments in the middle, and establishes a mapping relationship between the genes in the exchanged segments to correct the genes in the non-exchanged segments, so the uniqueness of genes in each offspring chromosome can be ensured.

Mutation adopts an improved multi-point crossover mutation method, randomly selecting genes in the parent generation, reordering the selected genes and placing them back in their original positions but keeping the genes in other positions unchanged at the same time.

3.2.4 Improve elite retention strategy

In order to improve the quality of the subsequent generation population, an improved elite retention strategy has been introduced. After mutation and crossover, half of the optimal solutions in the population are retained. The other half are selected through roulette wheel selection based on fitness. Therefore, size of the initial population can be kept unchanged. This strategy aims to preserve the diversity of excellent individuals while giving other individuals the opportunity to evolve. The overall evolution and improvement of the population can be promoted. By balancing the relationship between preserving the optimal solution and maintaining population diversity, to accelerate the convergence speed and global search ability of the algorithm can be expected.

3.3 DDQN algorithm select the moved users

Markov decision process (MDP) is an important concept in reinforcement learning. A formal framework is provided for describing decision problems in a random environment. Define the inspection planning-problem as a finite MDP which consisting of five elements: state space S, action space A, state transition function $P(s' \mid s, a)$, reward function $R(s, a)$, and discount factor γ. Among them, the state transition function describes the probability of transitioning from state s to the next state s' after performing an action a. In this problem, the state transition function is certainty, and based on the currently selected user, the reassigned state will be updated to the next deterministic state. The discount factor γ is used to balance the importance of immediate rewards and future rewards, typically within the range of [0, 1].

3.3.1 State space

The state space S describes all possible states of the environment at each time t. The state $s(t)$ contains the state information of the current power trading user and the inspection team. The state features $s_1(t)$ – $s_4(t)$ are for the inspection team, and the state features $s_5(t)$, $s_6(t)$ are for the inspected user.

State feature 1 At time t, formula (14) represents the average of the number of users that the inspection team needs to inspect. $un_j(t)$ represents the number of users that the team $\mathcal{T}_j$ needs to inspect at time t, and $un\_nor_j(t)$ in formula (15) represents the normalised result of $un_j(t)$. Normalisation can improve the training speed, stability, and generalisation ability of the model, and avoid imbalanced feature ratio.

State feature 2 At time t, formula (16) represents the standard deviation of the $un_{ave}(t)$ for all inspection teams.

State feature 3 At time t, $st_j(t)$ in formula (17) represents the sum of the standard working hours of the number of users that the inspection team $\mathcal{T}_j$ needs to inspect. The $st\_nor_j(t)$ in formula (18) represents the normalisation result of $st_j(t)$. Formula (19) represents the average of $st\_nor_j(t)$, and it seem as state feature 3.

State feature 4 At time t, formula (20) represents the standard deviation of $st\_nor_j(t)$ for all inspection teams.

State feature 5 At time t, formula (21) represents the average distance of user coordinates (USC). $\mathcal{T}_j(t)$ represents the number set that the team $\mathcal{T}_j$ needs to inspect at time t.

State feature 6 At time t, formula (24) represents the standard deviation of $d\_nor_i(t)$ for all users.

$$s_1(t) = un_{ave}(t) = \frac{\sum_j^n unnor_j(t)}{n} \tag{14}$$

$$un\_nor_j(t) = \frac{un_j(t) - \min_j un_j(t)}{\max_j un_j(t) - \min_j un_j(t)} \tag{15}$$

$$s_2(t) = un_{std}(t) = \sqrt{\frac{\sum_j^n \left(un_{nor_j(t)} - un_{ave}(t)\right)^2}{n}} \tag{16}$$

$$st_j(t) = \sum_{i=0}^{m} pt_i \times x_{i,j} \tag{17}$$

$$st\_nor_j(t) = \frac{st_j(t) - \min_j st_j(t)}{\max_j st_j(t) - \min_j st_j(t)} \tag{18}$$

$$s_3(t) = st_{ave}(t) = \frac{\sum_j^n st\_nor_j(t)}{n} \tag{19}$$

$$s_4(t) = st_{std}(t) = \sqrt{\frac{\sum_j^n \left(st\_nor_j(t) - st_{ave}(t)\right)^2}{n}} \tag{20}$$

$$s_5(t) = dis_{ave}(t) = \frac{\sum_i^m d\_nor_i(t)}{m} \tag{21}$$

$$d_i(t) = \sum_{i' \in T_j(t), i \neq i'} \left\| (x_i - x_{i'}), (y_i - y_i') \right\| \tag{22}$$

$$d\_nor_i(t) = \frac{d_i(t) - \min_i d_i(t)}{\max_i d_i(t) - \min_i d_i(t)} \tag{23}$$

$$s_6(t) = dis_{std}(t) = \sqrt{\frac{\sum_i^m \left(d\_nor_j(t) - dis_{ave}(t)\right)^2}{m}} \tag{24}$$

3.3.2 Action space

Action space A describes all possible actions that can be executed in each state, as shown in Table 1. In this problem, the action is to select mobile power trading users and reassign them to higher-level teams for inspection. The mobile power trading user levels are **M** and **L**. The intelligent agent selects the relevant action from the action space based on the current environment. It means to select a user for reassignment. If the selected user level is **L**, it will be assigned to the level **II** team for inspection; if the selected user level is **M**, it will be assigned to the level **III** team for inspection.

Table 1 Action space

Num.	*Notion*	*Description*
1	SP (shortest pt_i)	Select the user with the shortest standard work time
2	LP (longest pt_i)	Select the user with the longest standard work time
3	SD (shortest $d_i(t)$)	Select the user with the shortest average deviation distance
4	LD (longest $d_i(t)$)	Select the user with the biggest average deviation distance

3.3.3 Reward function

The reward function $R(s, a)$ is an important part of guiding the behaviour of intelligent agents. It defines the feedback that the intelligent agent after taking action a in the current state s. The learning effectiveness and final performance of the intelligent agent will be directly affected due to the design of reward function. Reward function as shown in formula (25).

$$r(t) = cost(t) - cost_{optimal} \tag{25}$$

$cost(t)$ represents the total cost at time t. $cost_{optimal}$ represents the historical optimal total cost, also means the minimum value of the total cost for this iteration.

3.3.4 Algorithm flow

- Initialisation: Initialise the intelligent agent, including strategy network Q, target network Q', and experience recover pool. The structure of the strategy network Q and the target network Q' are the same, both of which are 6 × 32 × 16 × 4. But the network parameters are different, strategy network Q is θ, the other one is θ'. The network structure of intelligent agent is shown in Figure 3.

$$a(t) = \begin{cases} \arg\max_a Q\left(s(t), a; \omega\right); & \text{if } rand() \le \epsilon \\ rand(a); & \text{else} \end{cases} \tag{26}$$

$$L(\omega) = \mathbb{E}\left[\left(y - Q(s, a; \theta)\right)^2\right] \tag{27}$$

$$y = r + \gamma Q'\left(s', \arg\max_a Q(s', a; \theta); \theta'\right) \tag{28}$$

$$\theta' \leftarrow \theta \tag{29}$$

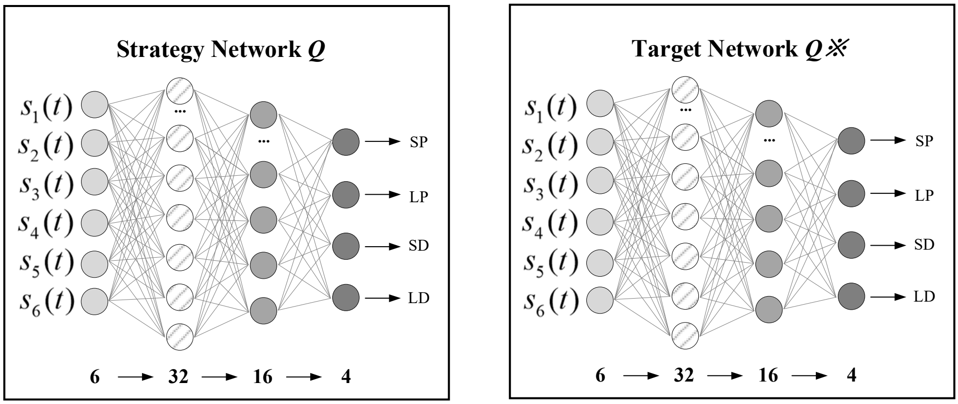
- Action selection: Update the current status, and the intelligence agent selects actions based on the epsilon-greedy strategy. The epsilon-greedy strategy is used to balance the relationship between exploration and exploitation. Parameter ϵ is used to control the manner the intelligence agent selects actions, helping the agent to explore new possible actions and utilise the current optimal strategy. The action selection is shown in formula (26).

  $rand()$ is a function that returns a random number between 0 and 1. $Q(s(t), a; \theta)$ represents the Q for selecting action a at the current time s (t), which calculated from the neural network parameter θ. $\text{argmax}_a Q(s(t), a; \theta)$ represents selecting the action of the biggest Q.

- Execute and update: Execute action $a(t)$ and calculate the current reward $r(t)$ based on the reward function. Update the next status $s(t + 1)$ and place $(s(t), a(t), r(t), s(t + 1))$ as a whole (s, a, r, s') into the experience recover pool.

- Update network parameters: Every specified period learning frequency, randomly select a batch of experiences (s, a, r, s') called ‘batch’ from the experience recover pool for training. Use mean square error as the Loss-function, and update the parameters of the strategy network, as shown in formulas (27) and (28). Update the target network parameters with strategy network parameters every fixed number of steps, as shown in formula (29).

Figure 3 Network structure



4 Simulation analysis

4.1 Parameter settings

In order to verify the effectiveness of the inspection path planning method based on the combination of DDQN and GA proposed in this article, this section conducts simulation analysis on the algorithm. The experiment was conducted in the Python 3.7 environment, using PyTorch to build the strategy and target network model. The relevant parameter settings are shown in Table 2.

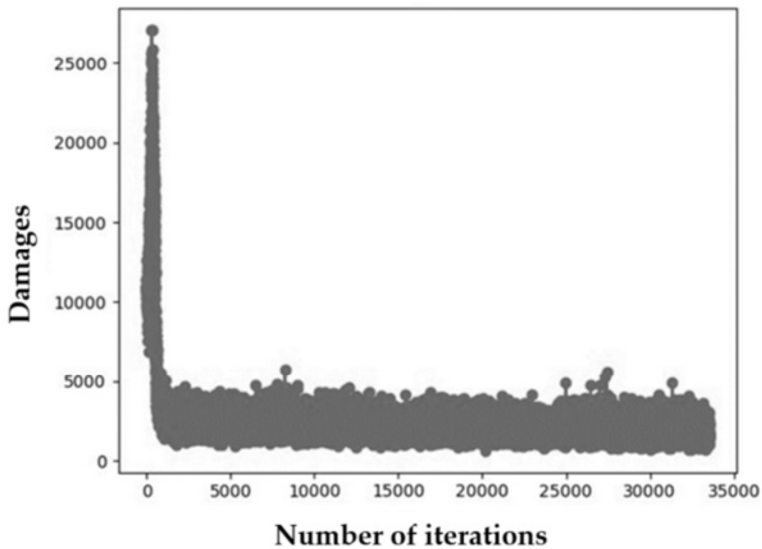
Table 2 Parameter setting

Parameter	*Value*
GA iteration	200
GA population size	50
Batch	32
ε	0.90
γ	0.85
Learning frequency	10
Size of experience pool	10,000
Target network update frequency	100

4.2 Training process

This section aims to train intelligent agent to optimise strategies of user grouping and minimise total costs. During the training process of the intelligent agent, the intelligent agent continuously iterates and adjusts the scheme to obtain the optimal result and the loss value gradually decreases. It is shown in Figure 4. The horizontal axis in the figure represents the training frequency, and the vertical axis represents the loss value. With the training rounds increase, the agent gradually learns better selection strategies. It makes the loss value continuously decrease and eventually stabilise.

Figure 4 Training process-loss



From Figure 4, it can be seen that the loss value decreases rapidly in the early stages. It means the intelligent agent quickly learned some basic selection strategies during the early exploration process. With the training progresses, the rate of decrease in loss values gradually slows down and eventually stabilises. It indicates the intelligent agent has found a better grouping strategy and the update of Q is gradually converging.

4.3 Algorithm comparison

This section will compare the proposed DDQN-GA algorithm with existing inspection path planning algorithms to evaluate their performance and advantages. Comparative algorithms include greedy algorithm, GA and K-means combined with a single rule (K-means-single rule). The greedy algorithm selects the nearest user each time, and the termination iteration of GA is 1,000 times. K-means refers to clustering users based on their coordinates, aiming to minimise the inspection range within the same team. The single rule refers to selecting the user with the smallest latest inspection time each time.

In order to comprehensively evaluate the performance of the four algorithms, as shown in Table 3, each experiment was compared using the three algorithms to calculate their total cost and computation time.

The experimental results are shown in Table 3. Table 3 shows the comparison of total costs and computing time under different numbers of teams and users.

Table 3 Comparison of total costs and computing time under different team numbers and user numbers

Number of teams	*Number of users*	*DDQN-GA*		*Greedy algorithm*		*GA*		*K-means-single rule*	
		Total cost	*Computing time (s)*	*Total cost*	*Computing time (s)*	*Total cost*	*Computing time (s)*	*Total cost*	*Computing time (s)*
5	15	110.47	0.20	115.68	3.11	110.47	7.47	116.54	0.1
	25	142.39	0.24	146.73	3.87	143.95	13.28	151.16	0.1
	35	214.29	0.33	217.98	4.57	215.75	20.18	231.01	0.1
10	30	180.56	0.52	184.93	4.32	182.57	18.61	208.71	0.1
	40	214.29	0.68	227.98	4.95	225.75	29.34	225.72	0.1
	50	204.63	0.79	237.12	5.13	235.86	37.12	252.41	0.1
15	45	249.61	0.73	261.14	4.75	260.27	32.08	271.44	0.1
	55	276.62	0.78	283.33	5.35	289.76	45.20	301.91	0.1
	65	317.47	0.86	344.12	6.11	360.48	62.50	414.87	0.1
20	60	292.82	0.92	353.57	5.64	236.93	59.60	371.48	0.1
	70	364.56	0.96	413.93	6.96	388.57	71.50	463.87	0.1
	80	438.29	1.05	515.98	7.33	481.75	105.8	563.98	0.1
25	75	389.18	1.32	475.67	8.47	433.21	88.7	511.98	0.1
	85	438.29	1.38	535.98	9.55	471.75	123.2	559.78	0.1
	95	484.37	1.43	599.89	10.60	537.64	165.9	624.71	0.1

Table 4 Team information

Number of Inspection team	*Level of team capability*	*Coefficient of distance time*	*Coefficient of distance cost*
1	I	0.12	0.18
2	II	0.15	0.14
3	III	0.18	0.10

Table 5 User information

Number of users	*X coordinate*	*y coordinate*	*Standard waiting time for work*	*Latest inspection time*	*Overdue penalty, early reward*	*Level of risk*
0	49	10	18	36	1.5	L
1	73	29	11	55	1.7	H
2	11	87	5	15	1.3	L
3	45	63	17	85	1.9	L
4	92	8	8	24	1.5	H
5	31	50	14	42	1.8	M
6	66	99	19	38	1.2	M
7	18	41	6	18	1.6	H
8	84	75	4	8	1.4	L
9	6	53	9	27	1.1	M
10	97	20	15	45	1.0	H
11	37	84	16	32	1.3	L
12	15	52	5	15	1.6	L
13	68	29	9	27	1.8	L
14	91	73	12	18	1.1	L
15	74	23	7	17	1.2	L

From Table 3, it can be concluded that the DDQN-GA algorithm outperforms the greedy algorithm, GA and K-means-Single Rule under different team and user numbers. Show lower overall costs. Especially in large-scale cases, the advantages of DDQN-GA algorithm are more obvious, especially in terms of global search capability. This is of great significance for resource optimisation and cost control in practical applications.

From Table 3, DDQN-GA has a longer computing time compared to K-means-single rule; however, it demonstrates a significant advantage in terms of the total cost. The computation time of the DDQN-GA algorithm is not excessively long, and its work time is notably shorter than that of the greedy algorithm and traditional GA. Especially when the number of teams and users increases, the growth of work time of DDQN-GA algorithm is relatively flat. This indicates that the DDQN-GA algorithm has high computational efficiency while ensuring optimisation effectiveness. Therefore, the DDQN-GA algorithm can satisfy the real-time requirements in practical applications.

4.4 Instance test

This section will verify the effectiveness of the DDQN-GA inspection path planning algorithm proposed in this paper through a specific example.

There are three inspection teams and 16 users in the system. The overdue penalty is equal to the advance completion reward. The specific information is shown in Tables 4 and 5.

Table 6 Result of instance test

Number of users	*K-means-single rule*		*DDQN-GA*	
	Actual completion time	*The difference from the latest inspection time*	*Actual completion time*	*The difference from the latest inspection time*
0	51.86	15.86	68.66	32.66
1	86.82	31.82	38.63	-16.37
2	14.23	-0.77	41.91	26.91
3	95.54	10.54	72.55	-12.45
4	17.81	-6.19	8	-16
5	76.25	34.25	14	-28
6	40.5	2.5	96.55	58.55
7	25.81	7.81	51.38	33.38
8	4	-4	17.31	9.31
9	36.85	9.85	31.76	4.76
10	72.74	27.74	24.56	-20.44
11	58.11	26.11	61.83	29.83
12	5	-10	21.41	6.41
13	30.64	3.64	35.07	8.07
14	17.09	-0.91	12	-6
15	7	-10	41.91	24.91

The existing solution method uses K-means-single rule, with the hyperparameter group set to *n*, and the results are shown in Table 6. The total cost is calculated by the difference from the latest inspection time. The final total cost obtained is 278.86, while the total cost calculated using DDQN-GA is 218.56, representing an improvement of 21.6%.

The user grouping and inspection order using K-means-single rule are as follows:

Team 1 User 8, user 14, user 6.

Team 2 User 12, user 2, user 7, user 9, user 11, user 5, user 3.

Team 3 User 15, user 4, user 13, user 0, user 10, user 1.

The user grouping and inspection order using DDQN-GA are as follows:

Team 1 User 4, user 10, user 1, user 7, user 3, user 6.

Team 2 User 5, user 12, user 9, user 2, user 11.

Team 3 User 14, user 8, user 13, user 15, user 0.

5 Conclusions

As the electricity trading market continues to expand and grow more complex, traditional inspection methods struggle to cope with the ever-changing environment and increasing demands, highlighting the growing importance of intelligent inspection technologies. This paper proposes an inspection path planning method that combines DDQN and GA. In the inspection of the power system, the allocation and path optimisation of long-term power trading risk users can be solved through this method. The DDQN algorithm is used to implement intelligent allocation of users and inspection groups. Use GA algorithm to optimise the path of each team, so the overall optimal inspection plan can be achieved. The simulation experiment results show that the algorithm proposed in this paper exhibits superior performance under various settings. Compared with traditional greedy algorithms or standalone GAs, DDQN-GA algorithm has significant advantages in reducing total cost and shortening work time. Specifically, DDQN-GA effectively reduces the total cost of inspections and significantly improves work efficiency. It performs particularly well when there are a large number of teams and users. The algorithm proposed in this paper has practical application value in improving the efficiency of power system inspection, optimising resource utilisation, and reducing costs.

Currently, path obstacles are not considered, and coordinates are based on straight-line distances. Future work could incorporate obstacle constraints and more complex path planning methods to enhance the algorithm's real-world applicability. Additionally, the performance of the algorithm in handling big datasets warrants further investigation to improve its scalability, robustness, and efficiency across different scenarios. By continuously optimising the algorithm's practicality and adaptability, it will be better equipped to address challenges in complex industrial environments.

Acknowledgements

This work was supported by the State Grid Corporation of China's Science and Technology Project 'Research on the evaluation technology and prevention mechanism of medium – and long term trading market power under a diversified value system' (Grant No. 5222YF240002).

Declarations

All authors declare that they have no conflicts of interest.

References

Du, Z. and Li, H. (2024) 'Research on application of improved quantum optimization algorithm in path planning', *Applied Sciences*, Vol. 14, No. 11, p.4613.

Guo, J., Shen, J., Guo, X. et al. (2024) 'Research on the design and development of electric power inspection and monitoring system and data preprocessing', *Electrical Technology and Economy*, No. 1, pp.53–55+58.

Li, D., Wang, X., Li, L. et al. (2024) 'Automated deep learning system for power UAV inspection image analysis and processing: architecture design and key technologies', *Power Information and Communication Technology*, Vol. 22, No. 4, pp.38–54.

Liu, C., Liu, A., Wang, R. et al. (2022) 'Path planning algorithm for multi-locomotion robot based on multi-objective genetic algorithm with elitist strategy', *Micromachines*, Vol. 13, No. 4, p.616.

Liu, Y. and Tan, J. (2017) 'Present situation and prospect of large consumers participation in power market transaction in Southern Region', *Southern Power System Technology*, Vol. 11, No. 11, pp.68–74.

Luo, X., Du, J., Tian, J. et al. (2022) 'An active distribution network high resilience decision-making method based on deep reinforcement learning', *Southern Power Grid Technology*, Vol. 16, No. 1, pp.67–74.

Qin, H. and Fei, Y. (2024) 'Intelligent inspection path planning method of hydropower station based on improved genetic algorithm', *Sichuan Water Resources*, Vol. 45, No. 1, pp.164–166.

Shakya, A.K., Gopinatha, P. et al. (2023) 'Reinforcement learning algorithms: a brief survey', *Expert Systems with Applications*, Vol. 231, p.120495.

Torres, G.A.S., Calumba, S.P., Fajardo, F. et al. (2024) 'Genetic algorithm-driven optimization for enhanced accessibility in mobile robotics', *10th International Conference on Control, Automation and Robotics (ICCAR)*, IEEE, pp.109–115.

Wang, B., Yang, H. and Zhang, S. (2024) 'Research on user privacy protection scheme in electricity call auction transaction system', *Electric Power Science and Engineering*, Vol. 40, No. 7, pp.42–51.

Wollmann, J., Muschalski, L., Wang, Z. et al. (2023) 'Application of genetic algorithm for the synthesis of path-generating compliant mechanisms', *Smart Materials and Structures*, Vol. 33, No. 1, p.015023.

Xie, J., Zhang, Q., Wu, J. et al. (2023) 'Intelligent substation inspection method based on digital twin technology', *Electrical Switchgear*, Vol. 61, No. 6, pp.87–89+113.

Xie, X., Zhuo, W. and Hu, P. (2016) 'Application of improved SAGA algorithm in substation inspection job scheduling', *Computer and Modernization*, No. 11, pp.109–113.

Yang, N., Xian, Y. and Zhu, X. (2024) 'Research and development of key technology of unmanned aircraft power inspection in the era of 'internet+'', *Internet Weekly*, No. 9, pp.42–44.

Yin, Y., Zhang, L., Shi, X. et al. (2024) 'Improved double deep Q network algorithm based on average Q-value estimation and reward redistribution for robot path planning', *Computers, Materials & Continua*, Vol. 81, No. 2, p.2769.

Zhu, X., Hu, C., Yang, J. et al. (2023) 'Resource allocation algorithm for power sensing network based on DDQN', *China Electric Power*, Vol. 56, No. 11, pp.60–66.