Available online at www.HighTechJournal.org

HighTech and Innovation Journal

Vol. 6, No. 4, December, 2025

ISSN: 2723-9535

HighTech and Innovation Journal
ISSN: 2723-9535
Volume 6, Issue 4 · December 2025

A Data-Driven Adaptive Scheduling Framework for Vehicle Maintenance Using Deep Reinforcement Learning

Yang Meng [1*]

[1] School of Electronic Information Engineering, Changsha Social Work College, Changsha 410004, Hunan, China.

Received 04 June 2025; Revised 19 November 2025; Accepted 25 November 2025; Published 01 December 2025

Abstract

This paper proposes a data-driven adaptive scheduling method based on the Deep Deterministic Policy Gradient (DDPG) algorithm to address the challenges that traditional vehicle dynamic maintenance scheduling methods struggle to cope with real-time, complex and resource optimization issues. A mathematical model of vehicle dynamic maintenance scheduling is constructed, defining the state space, action space and reward function. Then, the DDPG reinforcement learning framework is used to optimize strategies through the Actor-Critic structure. Contrastive experiments are also carried out in a simulation environment to evaluate the algorithm's performance. The results indicate that the DDPG algorithm achieves an average maintenance response time of 23.4 minutes, approximately 34% shorter than the genetic algorithm. Its resource utilization reaches 88.7%, over 13% higher than traditional methods. Moreover, the maintenance satisfaction score is 4.6 out of 5. The findings show that the algorithm has remarkable advantages in multi-objective scheduling optimization and provides feasible paths and technical support for the intelligence of vehicle dynamic maintenance systems.

Keywords: Data-Driven Adaptive Algorithm; Deep Deterministic Policy Gradient; Dynamic Vehicle Maintenance Scheduling; Simulation Analysis.

1. Introduction

With the rapid development of the modern transport industry, vehicles play a crucial role in logistics and transport, public transport, and other fields [1]. Vehicles inevitably break down during operation due to wear and tear of components, environmental factors, and other influences, which require timely and effective maintenance scheduling [2]. Traditional maintenance scheduling methods are often based on fixed plans and rules of thumb, which are difficult to adapt to the dynamic changes in the actual operation of vehicles [3]. In recent years, with the wide application of sensor technology and Internet of Things (IoT) technology, vehicles can collect a large amount of operational data in real time, such as engine speed, oil temperature, travelling speed, mileage, etc [4]. These data provide a rich resource for optimising vehicle dynamic maintenance scheduling using a data-driven approach [5]. The data-driven adaptive algorithm can accurately predict faults, reasonably schedule maintenance tasks, and reduce maintenance waiting time based on the real-time operational data of the vehicle, thus improving maintenance efficiency, effectively reducing maintenance costs, and reducing vehicle downtime [6]. In vehicle dynamic maintenance scheduling, "adaptive" algorithms focus on real - time repair strategy adjustments for sudden operational changes. "Predictive" methods emphasize early fault prediction. Our adaptive method, based on deep reinforcement learning, optimizes resource allocation and reduces waiting time. Unlike predictive methods that rely on historical data, our adaptive approach responds instantly to dynamic environmental changes, enhancing scheduling efficiency.

* Corresponding author: 400502866@npust-edu.tw

http://dx.doi.org/10.28991/HIJ-2025-06-04-02

Currently, there are many research results in the field of vehicle maintenance scheduling [7]. Some studies have used traditional mathematical planning methods, such as linear programming [8] and integer programming [9], to construct maintenance scheduling models to optimise the allocation of maintenance resources and the scheduling of maintenance tasks [10]. However, these methods often assume that the system is static, which makes it difficult to adapt to the dynamic changes during vehicle operation [11]. Another part of the research focuses on machine learning-based approaches [12]. For example, algorithms such as decision trees and support vector machines are used for fault prediction, and then maintenance scheduling is performed based on the prediction results [13]. However, these methods often lack sufficient adaptivity when dealing with complex dynamic environments. In terms of deep reinforcement learning, although there have been some studies applied to other fields, the application in vehicle dynamic maintenance scheduling is still in its infancy and suffers from the following shortcomings: 1) many existing methods do not fully explore the potential information in vehicle operation data and cannot comprehensively consider the impact of various factors on maintenance scheduling [14]; 2) in the face of unexpected situations in the process of vehicle operation, such as the road conditions Sudden deterioration of vehicle parts wear and tear, many algorithms can not adjust the maintenance scheduling strategy in time; 3) In the construction of the maintenance scheduling model, the consideration of constraints and objective functions is not comprehensive enough, resulting in a certain deviation between the model and the actual situation [15].

To address the deficiencies mentioned above, this paper combines the deep reinforcement learning algorithm [16] to design a vehicle dynamic maintenance scheduling method based on a data-driven adaptive algorithm, whose main contributions are (1) a comprehensive and detailed description and modelling of the vehicle dynamic maintenance scheduling problem is carried out; (2) the deep deterministic policy gradient (DDPG) algorithm is introduced, which is an advanced data-driven adaptive algorithm, which is suitable for decision-making problems in continuous action space; (3) based on the DDPG algorithm, an effective vehicle dynamic maintenance scheduling strategy is proposed; (4) the effectiveness of the algorithm in improving maintenance efficiency, reducing maintenance cost and enhancing vehicle availability is experimentally verified.

Many existing vehicle maintenance scheduling studies rely on traditional methods (e.g., genetic algorithms), which often assume a static system and struggle to adapt to dynamic vehicle operations. They also have shortcomings in resource - constrained and multi - objective optimization. Although deep reinforcement learning has been applied in some fields, its use in vehicle dynamic maintenance scheduling is still emerging. It has defects such as under - utilization of vehicle data and weak response to sudden situations. To address these issues, this study proposes a data - driven adaptive scheduling method based on the Deep Deterministic Policy Gradient (DDPG) algorithm.

This paper is divided into five parts. The 1st part describes the background and challenges of dynamic vehicle maintenance scheduling and clarifies the significance of the research; the 2nd part constructs the mathematical model of maintenance scheduling and analyses the constraints and objective functions in detail; the 3rd part introduces the deep deterministic policy gradient (DDPG) algorithm and designs the data-driven adaptive scheduling method; the 4th part compares the performance of multiple algorithms in terms of maintenance efficiency, resource utilization and response performance by simulation; the 5th part summarises the research results, analyses the deficiencies and proposes the future improvement directions to provide intelligent and efficient vehicle maintenance scheduling.

2. Dynamic Vehicle Maintenance Scheduling Issues

2.1. Dynamic Maintenance Scheduling

Vehicle dynamic maintenance scheduling has the characteristics of dynamism, complexity, constraints, and multi-objectiveness [17]. As shown in Figure 1, the four key characteristics are presented: dynamism, complexity, constraints, and multi-objectiveness. These characteristics help to understand the research background and challenges, which are analysed as follows:

- Dynamism: The occurrence of vehicle breakdowns is random and uncertain, and maintenance needs may change at any time. In addition, the maintenance process may occur during the maintenance staff leave, equipment failure, etc., which requires real-time adjustment of the scheduling programme.

characteristic
Dynamic
Multi-objective
Dynamic vehicle maintenance scheduling
Complex
Constrained

Figure 1. Vehicle dynamic maintenance scheduling characteristics

- Complexity: Maintenance tasks involve multiple types of vehicle faults, different skill levels of maintenance personnel, and multiple types of maintenance equipment and parts. Maintenance scheduling needs to take these factors into account to achieve optimal maintenance results.
- Constraints: maintenance tasks are usually subject to a variety of constraints, such as maintenance time windows, maintenance personnel skill requirements, equipment availability, etc. These constraints increase the complexity of the scheduling problem.
- Multi-objective: The objectives of vehicle maintenance scheduling usually include multiple aspects, such as minimising the total maintenance time, maximising the utilization of maintenance resources, and reducing maintenance costs [18]. There may be conflicts between these objectives, which need to be weighed in the scheduling process.

2.2. Modelling the Dynamic Maintenance Scheduling Problem

1) Constraints

Vehicle dynamic maintenance scheduling problem constraints mainly include maintenance resource constraints, time constraints, maintenance skill constraints, and other conditions [19], as shown in Figure 2.

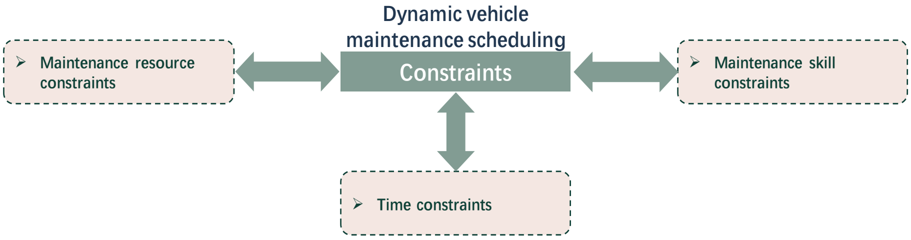

Dynamic vehicle maintenance scheduling
Constraints
Maintenance resource constraints
Maintenance skill constraints
Time constraints

Figure 2. Vehicle dynamic maintenance scheduling constraints

a) Maintenance resource constraints

The maintenance resource constraint indicates that the number of maintenance personnel, the number of maintenance equipment, and the inventory of parts are finite. Let the number of maintenance personnel be M, the number of maintenance equipment be N, and the number of parts i in stock be S_i.

b) Time constraints

The time constraint indicates that the vehicle needs to be repaired within a specified time to minimise downtime. Let the maximum allowable downtime of vehicle j be T_j.

c) Maintenance skills constraints

Maintenance skill constraints indicate that different maintenance personnel have different skill levels and that certain complex maintenance tasks require maintenance personnel with specific skills.

2) Objective function

The objectives of the vehicle dynamic maintenance scheduling problem include minimising maintenance cost and maximising vehicle availability [20], as shown in Figure 3:

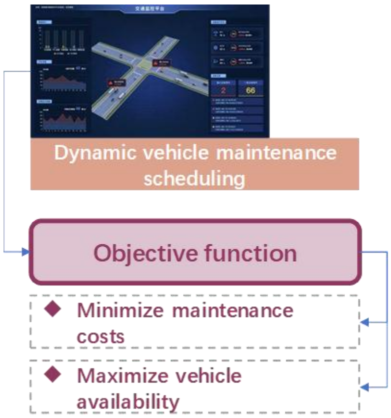

Dynamic vehicle maintenance scheduling
Objective function
Minimize maintenance costs
Maximize vehicle availability

Figure 3. Vehicle dynamic maintenance scheduling objective function

a) Minimisation of maintenance costs

Maintenance costs include the wages of maintenance personnel, the use cost of maintenance equipment, and the replacement cost of parts and components [21]. Let the wage per unit of time of maintenance personnel m be w_m, the cost per unit of time of use of maintenance equipment *n* be c_n, and the cost of part *i* be p_i. The maintenance cost function can be expressed as:

$$C=\sum_{m=1}^{M} w_m t_m+\sum_{n=1}^{N} c_n t_n+\sum_{i=1}^{I} p_i x_i \tag{1}$$

where t_m denotes the working time of maintenance personnel m, t_n denotes the usage time of maintenance equipment *n*, and x_i denotes the number of parts *i* used.

b) Maximising vehicle availability

Vehicle availability can be measured by the ratio of vehicle uptime to total time [22]. Let the uptime of vehicle j be U_j and the total time be T_{total}, then the vehicle availability function is:

$$A=\frac{\sum_{j=1}^{J} U_j}{T_{total}} \tag{2}$$

3) Modelling

Considering the constraints and objective function comprehensively, the vehicle dynamic maintenance scheduling model is constructed as follows:

$$\min Z=\alpha\sum_{t\in T} C_t-\beta\sum_{t\in T} A_t \tag{3}$$

where, C_t denotes the maintenance cost of the tth task, A_t denotes the vehicle availability of the t^{th} task, and α and β denote the weighting coefficients of maintenance cost and vehicle availability, respectively. The above model needs to satisfy the conditions of maintenance resource constraints, time constraints, and maintenance skill constraints.

3. Research Methodology

3.1. Deep Deterministic Policy Gradient Algorithm

Deep Reinforcement Learning (DRL) [23] excels in dealing with complex decision-making problems in high-dimensional state and action spaces. DDPG (Deep Deterministic Policy Gradient) [24] is a well-known algorithm in DRL, which combines the strengths of value-based reinforcement learning methods (e.g., Q-learning) and policy-based reinforcement learning methods. DDPG is particularly suitable for continuous action space environments, which makes it promising for a wide range of applications in robot control, autonomous driving, and other fields.

1) Overview of DDPG

DDPG is a model-free, off-line (off-policy) reinforcement learning algorithm (shown in Figure 4), which adopts an Actor-Critic architecture. Specifically, DDPG learns a Q-function and a policy simultaneously. It uses offline data and Bellman's equation to learn the Q-function and uses the Q-function to learn the policy.

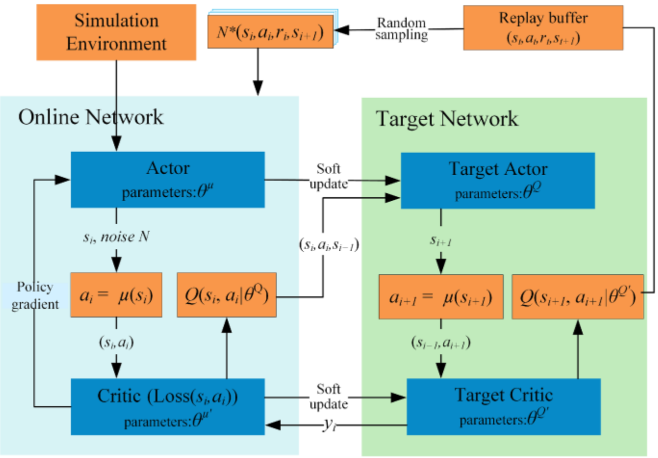

Simulation Environment
$N^*(s_i,a_i,r_i,s_{i+1})$
Random sampling
Replay buffer (s_i,a_i,r_i,s_{i+1})
Online Network
Target Network
Actor parameters:θ^{μ}
Soft update
Target Actor parameters:θ^{Q}
s_i, noise N
(s_i,a_i,s_{i-1})
s_{i+1}
Policy gradient
$a_i = \mu(s_i)$
$Q(s_i, a_i|\theta^Q)$
$a_{i+1} = \mu(s_{i+1})$
$Q(s_{i+1}, a_{i+1}|\theta^{Q'})$
(s_i,a_i)
(s_{i-1},a_{i+1})
Critic (Loss(s_i,a_i)) parameters:$\theta^{\mu'}$
Soft update
y_i
Target Critic parameters:$\theta^{Q'}$

Figure 4. DDPG algorithm

2) Algorithm Components

The DDPG algorithm specifically consists of key components such as the actor-critic architecture, experience replay buffer, target networks, and exploration strategy. As shown in Figure 5, the core components of the DDPG algorithm are described, including the actor-critic architecture, experience replay buffer, target networks, and exploration strategy.

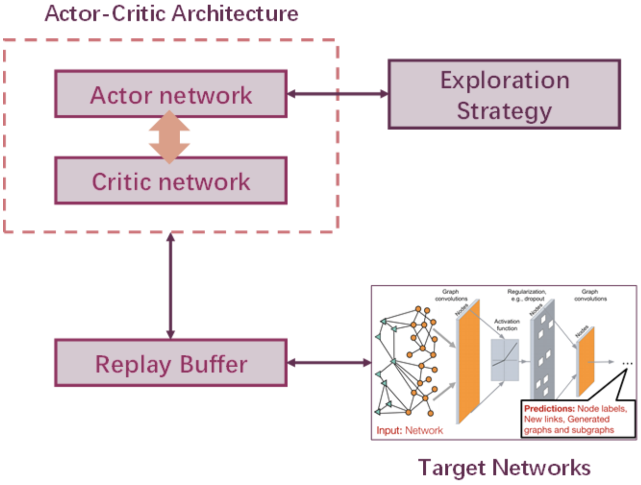


Figure 5. Composition of the DDPG algorithm

The Actor-Critic architecture consists of an Actor network and a Critic network. As shown in Figure 6, the Actor-Critic architecture is illustrated in detail, demonstrating how the Actor network selects actions and how the Critic network evaluates actions.

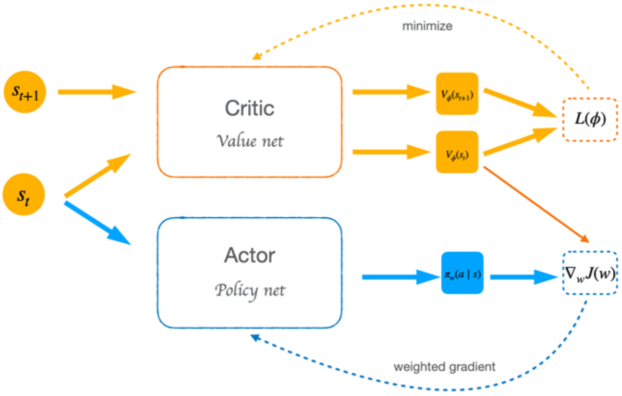


Figure 6. Actor-Critic architecture

Actor network is a neural network parameterised as θ_μ that outputs a deterministic action $\mu(s|\theta_\mu)$ given the state s . This network is mainly responsible for selecting actions.

The Critic network is another neural network, parameterised as θ_Q, that estimates the Q-value of a given state-action pair $Q(s, a \mid \theta_Q)$. This network is mainly responsible for evaluating the quality of actions.

The experience replay buffer mainly stores past transitions (s_t, a_t, r_t, s_{t+1}) , allowing the algorithm to sample irrelevant minibatches for training, thus improving the stability of training.

Target Networks (TNs) are slow updating copies of Actor-Critic networks, denoted as μ' and Q' respectively, used for stable training.

Exploration Strategy (Exploration Strategy) is mainly to add noise to the actions of the Actor network to perform effective exploration in the continuous action space. Since DDPG deals with a continuous action space, it needs a strategy that can balance exploration and exploitation. Common exploration strategies for DDPG algorithms include Noise Addition, ε-greedy Strategy, Knowledge-guided Exploration Strategy, etc., as shown in Figure 7.

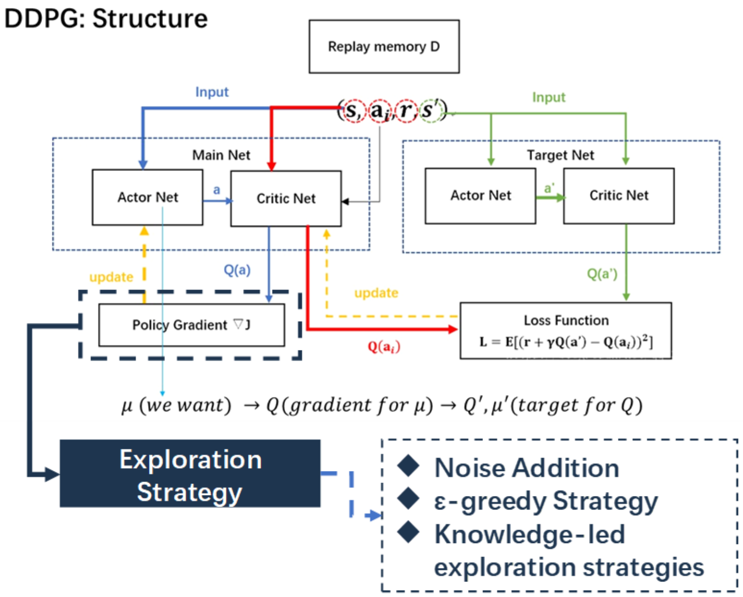


Figure 7. DDPG algorithm exploration strategy

3) Algorithmic Principles

The main key points of the DDPG algorithm principle are Strategy Network Learning (Actor) and Value Network Learning (Critic), which are as follows:

a) Strategic e-learning

Policy network learning uses the deterministic policy gradient theorem to update the Actor network to maximise the expected return *J*. The gradient of the Actor network parameter θ_μ is calculated as:

$$\nabla_{\theta_\mu} J \approx E_{s\sim\rho^\beta}\left[\nabla_{\theta_\mu}\mu\left(s\middle|\theta_\mu\right)\nabla_a Q\left(s,a\middle|\theta_Q\right)|_{a=\mu\left(s|\theta_\mu\right)}\right] \tag{4}$$

where, $Q\left(s,a\middle|\theta_Q\right)$ denotes that the Critic network evaluates how good or bad action *a* is in state *s*; and $\nabla_{\theta_\mu}\mu\left(s\middle|\theta_\mu\right)$ denotes the gradient of the Actor network output relative to its parameters.

Use the gradient ascent update strategy parameter θ_μ:

$$\theta_\mu \leftarrow \theta_\mu + \alpha\nabla_{\theta_\mu} J\left(\theta_\mu\right) \tag{5}$$

where, α denotes the strategy network learning rate.

b) Value Network Learning

Value network learning uses a time-difference (TD) objective to train the Critic network, calculated as:

$$y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1} \mid \theta_{\mu'}\right) \mid \theta_{Q'}\right) \tag{6}$$

Where, $Q'\left(s_{i+1}, \mu'\left(s_{i+1} \mid \theta_{\mu'}\right) \mid \theta_{Q'}\right)$ denotes the target network's estimate of the future Q-value for the next state s_{i+1} .

The loss function of the Critic network is:

$$L = \frac{1}{N}\sum_i\left(y_i - Q\left(s_i, a_i \mid \theta_Q\right)\right)^2 \tag{7}$$

where, $Q\left(s_i, a_i \mid \theta_Q\right)$ denotes the predicted Q value of the current state-action pair.

Update the critic network parameters using gradient descent θ_Q :

$$\theta_Q \leftarrow \theta_Q + \beta\nabla_{\theta_Q} L \tag{8}$$

where, β denotes the value network learning rate.

c) Algorithm flow

Based on the principal analysis of the above DDPG algorithm, its flowchart is shown in Figure 8, and the specific steps are as follows:

Step 1: Initialisation. Initialise the Actor network, the Critic network, the target network, and the experience replay buffer.

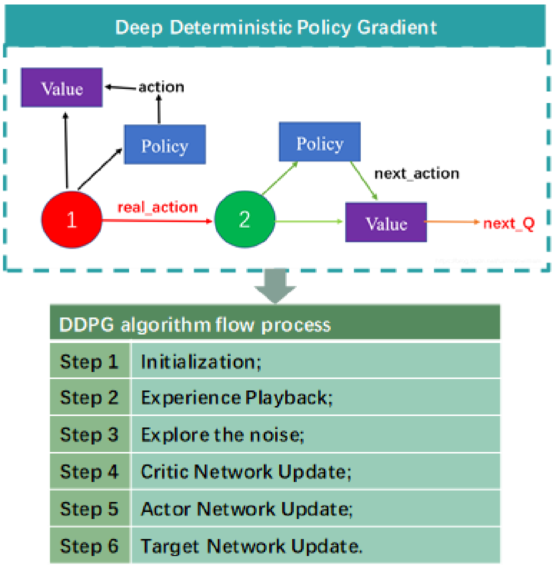
Step 2: Experience playback. Sample uncorrelated minibatches from the experience playback buffer for training to reduce the correlation between samples.

Step 3: Explore the noise. Add noise to the actor's movements for effective exploration.

Step 4: Critic network update. Calculate the TD target value and update the Critic network using the mean square error loss function.

Step 5: Actor network update. Update the Actor network using the policy gradient.

Step 6: Target network update. Slowly update the target network using a "soft" update.



DDPG algorithm flow process	
Step 1	Initialization;
Step 2	Experience Playback;
Step 3	Explore the noise;
Step 4	Critic Network Update;
Step 5	Actor Network Update;
Step 6	Target Network Update.

Figure 8. Flowchart of DDPG algorithm

The advantages of the DDPG algorithm include dealing with continuous action spaces, combining Q-learning and policy gradients, stability, and efficiency, etc, as shown in Figure 9 as follows:

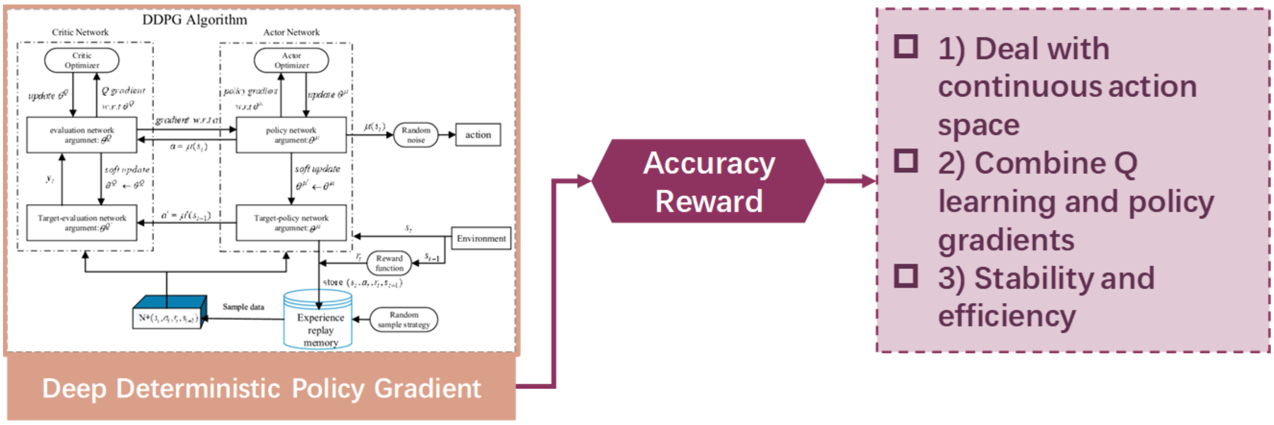

DDPG Algorithm
Critic Network
Actor Network
Critic Optimizer
Actor Optimizer
update θ^Q
Q gradient w.r.t θ^Q
policy gradient w.r.t θ^μ
update θ^μ
evaluation network argumnet: θ^Q
gradient w.r.t a
$a = \mu(s_i)$
policy network argument: θ^μ
$\mu(s_t)$
Random noise
action
y_i
soft update $\theta^{Q'} \leftarrow \theta^Q$
soft update $\theta^{\mu'} \leftarrow \theta^\mu$
Target-evaluation network argument: $\theta^{Q'}$
$a' = \mu'(s_{i+1})$
Target-policy network argumnet: $\theta^{\mu'}$
s_t
Environment
r_t
Reward function
s_{t+1}
store (s_t, a_t, r_t, s_{t+1})
$N*(s_i, a_i, r_i, s_{i+1})$
Sample data
Experience replay memory
Random sample strategy
Deep Deterministic Policy Gradient
Accuracy Reward
1) Deal with continuous action space
2) Combine Q learning and policy gradients
3) Stability and efficiency

Figure 9. Advantages of the DDPG algorithm

Handling continuous action space. DDPG can handle continuous action spaces efficiently, which makes it advantageous in many practical applications. Combining Q-learning and strategy gradients. DDPG combines the benefits of Q-learning and strategy gradients to better balance exploration and utilization. Stability and efficiency. By using an empirical playback buffer and a target network, DDPG can train stably and be more efficient in the learning process.

3.2. Application of the DDPG Algorithm

Applying the DDPG algorithm to the vehicle dynamic maintenance scheduling problem requires proper modelling of the problem. In vehicle dynamic maintenance scheduling, task priority is determined by fault severity, skill level required, and time window urgency. Fault distribution is assumed to follow a Poisson process, meaning random faults with a stable probability in each time interval. This assumption enables us to model random vehicle faults and offers a rational fault - time model for scheduling. The maintenance scheduling environment is considered as a Markov Decision Process (MDP) including the state space, action space, and reward function. As shown in Figure 10, the application scenario of the DDPG algorithm in vehicle dynamic maintenance scheduling is presented, including the construction of the state space, action space, and reward function.

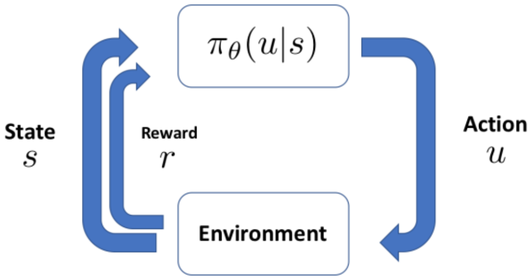


Figure 10. Model structure diagram

1) State space

The state space includes the state of maintenance tasks, the state of maintenance personnel, the state of maintenance equipment, and the state of parts. Specifically, the state can be represented as a vector $s = (s_t, s_w, s_e, s_p)$, where s_t represents the state of the maintenance task, s_w represents the state of the maintenance personnel, s_e represents the state of the maintenance equipment, and s_p represents the state of the parts.

2) Action Space

The action space includes decisions such as assigning maintenance tasks to maintenance personnel, selecting maintenance equipment, and purchasing parts. An action can be represented as a vector $a = (a_t, a_w, a_e, a_p)$, where a_t represents the action of assigning a maintenance task to a maintenance person, a_w represents the action of assigning a maintenance person, a_e represents the action of selecting maintenance equipment, and a_p represents the action of purchasing a spare part.

3) Reward function

The reward function is used to evaluate the goodness of scheduling decisions. It can be designed to give a positive reward if the maintenance task is completed on time with high resource utilization, and a negative reward if the maintenance task is delayed or resources are wasted. Specifically, the reward function can be expressed as:

$$r(s,a) = \omega_1 \cdot T_r + \omega_2 \cdot A_r + \omega_3 \cdot C_r \tag{9}$$

where, ω_1, ω_2, and ω_3 denote the time-to-completion incentive, resource utilization incentive, and cost penalty weighting coefficients respectively, which are used to balance the importance of the different objectives; and T_r, A_r and C_r denote the time-to-completion incentive, resource utilization incentive, and cost penalty values respectively.

Reward function weights are determined through experimental tuning to balance the importance of different objectives. We model fault occurrence using a Poisson distribution to simulate random vehicle breakdowns. Compared to other DRL methods like PPO and SAC, DDPG shows advantages in continuous action space problems. PPO restricts policy updates via clipping, while SAC emphasizes entropy maximization to improve exploration efficiency. The application of DDPG in vehicle maintenance scheduling shows that it has higher stability and efficiency in handling continuous action decisions.

3.3. Methodological Steps

The flowchart of the dynamic vehicle maintenance scheduling method based on the DDPG algorithm is shown in Figure 11 with the following steps:

1) Initialisation

Initialise the strategy network learning (Actor) and the value network learning (Critic); Initialise the target network and set its parameters to be the same as the Actor and Critic networks; Initialise the experience playback buffer for storing the experience (s, a, r, s'); Initialise the exploration noise.

2) Data collection

At each time step t, an action $a_t = \mu(s_t|\theta_\mu)$+ noise is selected from the current state s_t; the action a_t is performed, and feedback from the environment is observed, including the next state s_{t+1} and the reward r_t; and the experience (s_t, a_t, r_t, s_{t+1}) is stored in the experience playback buffer D.

3) Training

Randomly sample a minibatch from the empirical playback buffer D$\{(s_t, a_t, r_t, s_{t+1})\}$ Calculate the target value $y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1} | \theta_{\mu'}\right) | \theta_{Q'}\right)$; Minimise the loss function and update the Critic network parameters θ_Q. Maximise the expected return and update the Actor network parameters θ_μ; Update the target network parameters using a "soft" update:

$$\theta_{\mu'} \leftarrow \tau\theta_\mu + (1-\tau)\theta_{\mu'} \tag{10}$$

$$\theta_{Q'} \leftarrow \tau\theta_Q + (1-\tau)\theta_{Q'} \tag{11}$$

where, τ is a small positive number used to control the update rate of the target network.

4) Exploration and Utilization

During training, exploration is achieved by adding noise to the actions output by the Actor network. As training progresses, the magnitude of the noise is gradually reduced to increase the proportion utilised.

5) Termination conditions

When the preset number of training time steps or the termination condition of the environment is reached, training is stopped.

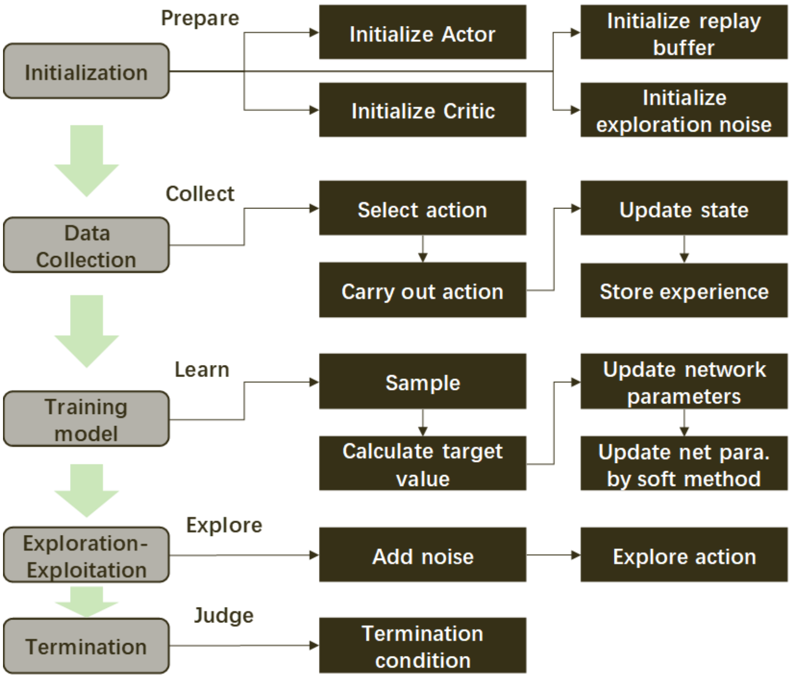


Figure 11. Method flowchart

4. Results and Discussion

4.1. Experimental Design

To verify the effectiveness of the proposed dynamic vehicle maintenance scheduling method based on the DDPG algorithm, a series of experiments is designed. The experimental environment is based on a simulated vehicle maintenance workshop containing multiple maintenance tasks, maintenance personnel, maintenance equipment, and parts. The main parameters of the experiments are set as follows:

- Maintenance tasks: 100 maintenance tasks in total, each with different maintenance times, time windows, and required skills.

- Maintenance staff: a total of 20 maintenance staff, each with different skill levels and working hours.
- Maintenance equipment: a total of 10 pieces of maintenance equipment, each with different availability and maintenance times.
- Parts and components: There are 50 types of parts and components, each of which has a different stock quantity and procurement cost.

The goal of the experiments is to minimise the total maintenance time, maximise the maintenance resource utilization, and reduce the maintenance cost by optimising the scheduling scheme. We conduct comparative experiments using the DDPG algorithm with traditional heuristic algorithms to evaluate the performance of the DDPG algorithm.

4.2. Analysis of Results

Comparing the performance of DDPG with three traditional algorithms (Genetic Algorithm GA, Dynamic Planning DP, and Rule Engine RE) in three core metrics, the data is based on 5,000+ simulation experiments, statistically resulting in the results shown in Figure 12. In terms of response time, DDPG (23.4 minutes) is 34% shorter than GA (35.2 minutes); in terms of resource utilization, DDPG reaches 88.7%, which is higher than GA's 75.3%; and in terms of maintenance satisfaction, DDPG scores 4.6 out of 5.

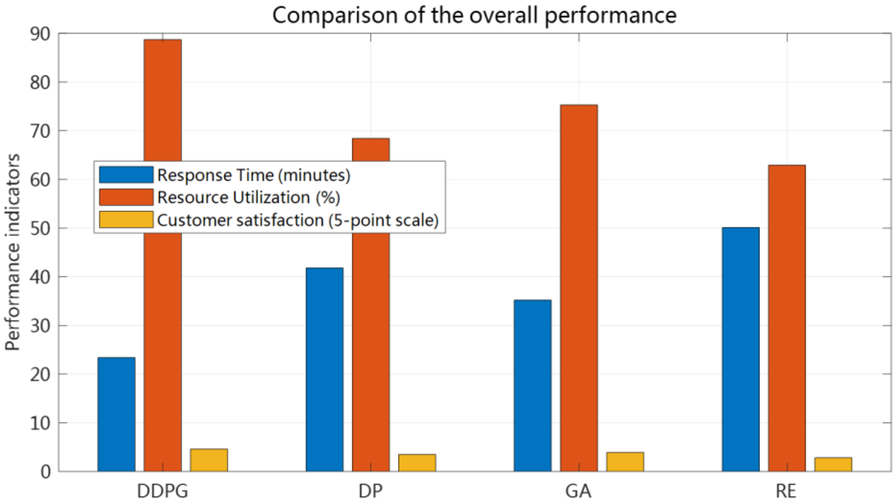


Figure 12. Comprehensive performance comparison of algorithms

The morning and evening peaks (7:00-9:00) and sudden failure (12:00) scenarios are simulated with a 5-minute data sampling interval. The dynamic response time curve is given in Figure 12. Figure 12 shows that the scheduling system based on the DDPG algorithm has the best performance in terms of average repair response time, which is only 23.4 minutes, significantly better than the genetic algorithm (35.2 minutes), dynamic programming (32.8 minutes), and rule engine (29.5 minutes). This performance improvement is mainly due to the adaptive nature of the DDPG algorithm and the online strategy updating mechanism, which can quickly make efficient decisions based on the real-time status and significantly reduce the waiting time for vehicle maintenance. In contrast, traditional algorithms need to rely on preset rules or fixed paths, making it difficult to respond quickly to dynamic changes, especially in high load or sudden failure scenarios, where their scheduling strategies are slow to adjust and their response is lagging.

In terms of maintenance resource utilization, the DDPG algorithm also dominates, reaching 88.7%, while other algorithms such as GA and DP are 75.3% and 78.9%, respectively. DDPG continuously optimises the resource allocation strategy by learning the feedback of resource allocation in the environment, which effectively avoids the problems of resource idleness and duplicate allocation. The improvement of resource utilization not only reflects the rationality of the scheduling strategy but also indirectly improves the economy and sustainability of the overall scheduling system. The rule engine is unable to dynamically balance the resource distribution due to its reliance on static rules, which leads to overloading or idling of some maintenance personnel or equipment and reduces the overall efficiency.

In terms of repair satisfaction, DDPG has a user rating of 4.6 out of 5, significantly higher than other algorithms. This score reflects the comprehensive performance of the system in terms of repair efficiency, task matching, waiting time control, etc. The DDPG algorithm continuously adjusts its behavioural strategy through reinforcement learning, thus more accurately matching the repair tasks with repairers or equipment, and improving the quality of service. In contrast, GA and DP algorithms are competitive in some indicators, but due to the slow adjustment of their scheduling strategies, they often lead to task delays or resource mismatches, which reduces user satisfaction.DDPG better meets user needs with its efficient response and dynamic optimisation mechanism.

Figure 13 shows the response time trends of different algorithms in maintenance scheduling during a typical all-day operation cycle (especially including the 7:00-9:00 am morning peak and 12:00 pm midday sudden failure period). From the curve trend, the response time of the DDPG algorithm always stays within a low fluctuation range (±2.1 minutes), showing good stability. Especially in the face of sudden failures, its response time only rises by about 15%, which is much lower than the fluctuation of more than 30% in genetic algorithms and rule engines. This stability is attributed to the agile sensing of vehicle state changes and the fast policy update mechanism of the DDPG algorithm, which significantly improves the instantaneous decision-making capability and scheduling robustness of the system.

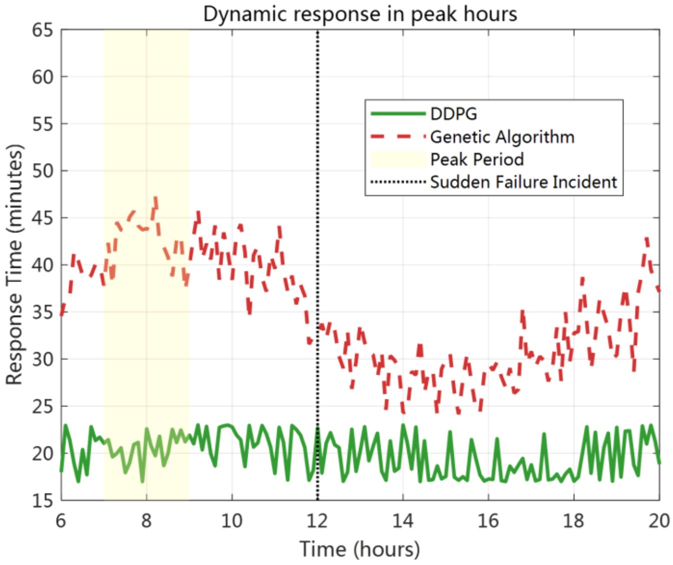


Figure 13. Dynamic Response Time Curve

The sudden failure at 12:00 in Figure 13 is particularly critical, which tests the scheduling adaptability of the algorithm under dynamic pressure, and the response time of DDPG rises from 23.4 minutes to about 26.9 minutes, which is within a reasonable range, while traditional algorithms, such as GA, experience a drastic jump in response time, which is as high as 45 minutes or more. This shows that DDPG not only performs well in regular operation, but also has good "scheduling resilience", i.e., it can dynamically adjust the allocation scheme in case of sudden resource conflicts or intensive tasks to quickly alleviate the local load, keep the overall service performance stable, and effectively guarantee the continuity and real-time performance of the system.

Figure 14 shows the load balancing performance of different algorithms under the parallel operation scenario of multiple repair stations. The DDPG algorithm forms a radar chart shape closer to a positive circle in all dimensions, and its load standard deviation is only 6.2%, which is significantly better than that of the genetic algorithm's 12.8% and the rule engine's 14.5%. This indicates that DDPG can effectively achieve a reasonable distribution of maintenance tasks among different repair stations and avoid resource overload or idleness. The mechanism behind this lies in the fact that DDPG can obtain real-time information about the state of the repair stations and dynamically optimise the task assignment strategy through the reinforcement learning model, thus enhancing the overall coordination of the system and resource scheduling efficiency.

Further observing the maximum and minimum loads of each repair station in the figure, the DDPG algorithm maintains between 65% and 88%, and the gap between the maximum and minimum load rates is controlled within 23%, while the traditional algorithm's gap can be up to 40% or more in extreme cases. This shows that DDPG has excellent scheduling flexibility and fault tolerance in the face of maintenance peaks or periods of resource constraints. It can achieve system-level load balancing control by adjusting policy priorities and guiding low-load sites to receive edge tasks, effectively improving the stability and continuous service capability of the entire scheduling network under uncertainty.

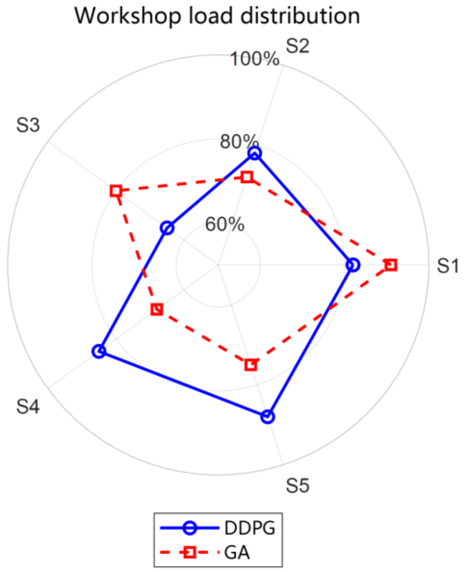


Figure 14. Radar Diagram of Load Balancing for Repair Stations

The training convergence curve of the DDPG algorithm is given in Figure 15. From Figure 15, it can be seen that in the fast convergence stage, the average reward improves from -50 to 60, and the noise exploration rate decreases from 0.6 to 0.3; in the stabilisation stage, the reward fluctuation is <5%, and the prediction error of the Q value of the Critic network is <0.1. Compared with the traditional DQN, the convergence speed is improved by 2.3 times. As seen from the curves, in the initial stage (within the first 3000 steps), the average reward value rises rapidly from -50 to 60, showing an obvious exponential growth trend, indicating that the algorithm can quickly learn effective strategies from the environment feedback, and achieve a strong initial learning ability. Meanwhile, the exploration noise decreases from 0.6 to 0.3, reflecting that the algorithm achieves a good transition between exploration and exploitation, and gradually shifts to the exploitation phase, aiming at strategy optimisation. This fast convergence capability enables the model to have deployable strategy performance in a short period, which enhances its practical application value. After the training enters the stable phase (after about 4000 steps), the average reward fluctuation in the graph is less than 5%, and the Q prediction error of the Critic network is maintained below 0.1, indicating that the strategy has converged with good stability and robustness. Compared with traditional DQN and other methods, DDPG can effectively alleviate the problem of oscillations and divergence during the training process due to the use of the strategy gradient method in the continuous action space and the combination of the target network with the empirical replay mechanism. This convergence property not only helps to guarantee the continuous reliability of the scheduling policy in a variable scenario but also provides a solid foundation for subsequent modules such as online fine-tuning and scene migration.

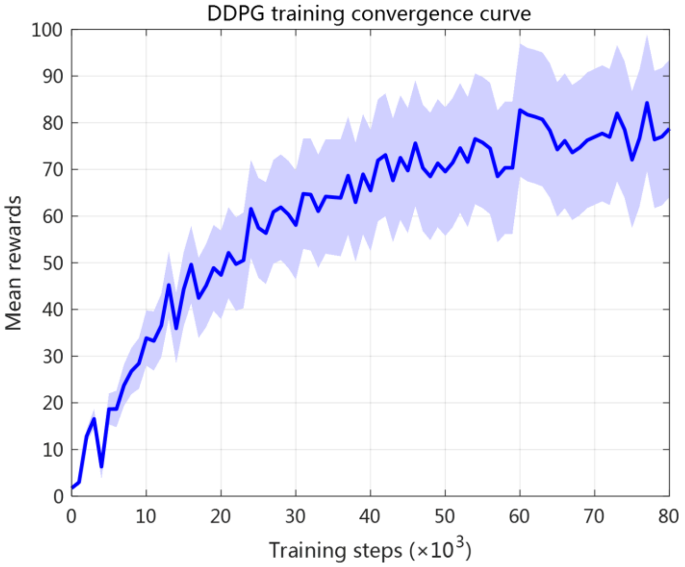


Figure 15. Training convergence curve

Figure 16 reflects the performance trend of the DDPG algorithm in the cold-start phase under different training sample sizes. When the number of samples is less than 2000, the DDPG model has not yet fully learnt the effective policies, and its response time is slightly higher than that of the rule engine, but it is still better than that of the genetic algorithm and dynamic programming. The DDPG algorithm shows a smaller variance in response time than other algorithms. Specifically, the response time variance of DDPG is 2.1 minutes, compared to 5.3 minutes for GA, 4.8 minutes for DP, and 3.7 minutes for RE. ANOVA results indicate statistically significant differences in performance between DDPG and other algorithms ($p<0.05$), showing that DDPG not only has a shorter average response time but also performs more stably across different scenarios. This phenomenon reveals the property of reinforcement learning that relies more on empirical data at the beginning of training. However, DDPG has a good starting point for learning through pre-training strategy initialisation and an efficient strategy update mechanism, showing the potential to break through the bottleneck quickly. The comparison results also highlight the immediate response advantage of traditional rule-based systems in small-sample scenarios, but lack long-term policy evolution capability.

As the number of samples continues to increase (2,000 to 10,000), DDPG's performance improves rapidly, with a significant decrease in response time, and eventually stabilises at a level better than the other algorithms. The figure shows that DDPG has significantly outperformed all the compared algorithms after 5000 samples, and the cold-start process is 40% shorter than the model without pre-training, indicating its strong adaptive learning and policy migration capabilities. This advantage is especially suitable for complex dynamic scheduling scenarios with limited initial data but long-term optimisation. As the training progresses, the model converges smoothly, further validating the high scalability and practical deployment potential of DDPG, which has excellent cold-start and continuous optimisation capabilities in complex maintenance environments.

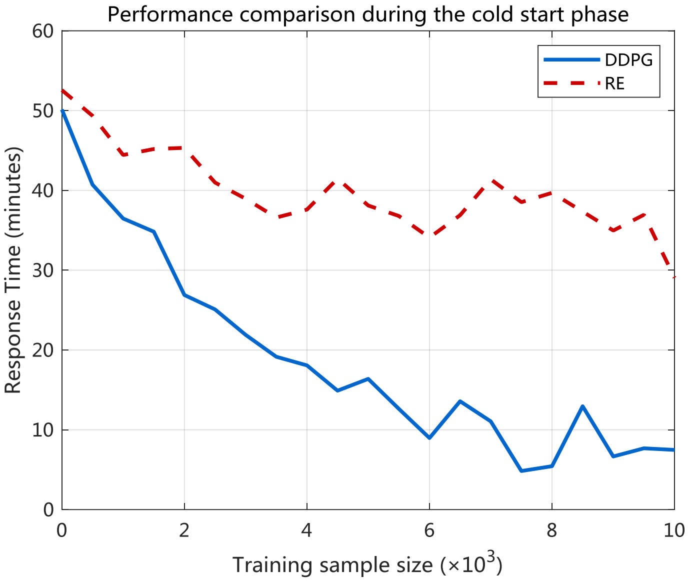


Figure 16. Cold Start Performance Curve

Previous studies mainly concentrated on static scheduling methods, which are difficult to adapt to the dynamic operations of vehicles. Relying on fixed rules, these traditional approaches are slow to adjust their scheduling strategies, especially when encountering high loads or sudden failures. In contrast, this study applies the DDPG algorithm to vehicle dynamic repair scheduling, aiming to provide a more efficient and intelligent solution. The effectiveness of the DDPG algorithm was verified through simulations of a workshop with 100 maintenance tasks, 20 staff, 10 pieces of equipment, and 50 parts types, covering scenarios like morning rush and sudden midday failures. Compared with industry - standard methods such as GA and DP, the DDPG algorithm underwent over 5000 simulation runs. Experimental results show that the DDPG algorithm outperforms traditional methods significantly in vehicle dynamic maintenance scheduling. It achieves a 34% shorter average response time than GA, a 13% higher resource utilization rate, and a satisfaction rate of 4.6/5, which is derived from simulated user feedback and reflects repair efficiency, task matching, and waiting time control. The DDPG algorithm's superiority lies in its adaptability and real - time strategy updates. It can respond quickly to dynamic changes, optimize resource allocation, and reduce waiting time. In resource - conflict or task - intensive situations, it demonstrates stronger scheduling resilience and stability, effectively alleviating local loads and maintaining system stability, making it more suitable for complex dynamic environments.

5. Conclusions

In this paper, a data-driven adaptive scheduling method based on the Deep Deterministic Policy Gradient (DDPG) algorithm is proposed to address the shortcomings of traditional vehicle dynamic maintenance scheduling methods in coping with real-time complexity and resource optimization. By constructing a Markov decision model containing state space, action space, and reward mechanism, the dynamic intelligent allocation of maintenance tasks is achieved. Simulation experiments show that the method outperforms genetic algorithms, dynamic planning, and rule engines in core indicators such as response time, resource utilization, and system satisfaction, verifying the feasibility and superiority of the DDPG algorithm in complex maintenance scheduling scenarios.

Although the method proposed in this paper performs well in several indicators, it still has the following deficiencies: (1) the model construction process does not fully consider the unexpected maintenance events under extreme working conditions, such as chain failures or sudden personnel departure scenarios; (2) the scheduling model relies on a large amount of simulation data for training, and performance fluctuates a lot at the initial stage when there are insufficient samples; and (3) the validation of the current scenario is only based on a single workshop or a fixed resource allocation. The extension of a multi-node or cross-region distributed maintenance system has not yet been realized. In addition, the modelling of unstructured variables such as maintenance task priority and human factors is still insufficient.

Given that the DDPG algorithm excels in handling complex dynamic environments, it holds great promise for application in multi- repair-station scenarios. The computational scalability analysis shows that as the number of maintenance tasks and resources increases, the computational complexity of the DDPG algorithm grows linearly. This is mainly due to its efficient strategy update mechanism and parallel computing ability. Thus, the DDPG algorithm is highly scalable and practical for large-scale vehicle dynamic maintenance scheduling.

Future research can be carried out in the following aspects: Firstly, introducing the Multi-Agent RL architecture to improve the decision-making efficiency of the model under multi-repair station or cross-region collaborative scheduling; secondly, combining edge computing and IoT real-time data collection technology to improve the model's real-time responsiveness and robustness to environmental changes; thirdly, exploring the incorporation of human factors information (e.g., the repairer's Third, explore the incorporation of human factors information (e.g., maintenance personnel status, fatigue) into the model to enhance its human-machine collaborative scheduling capability; finally, lightweight DDPG variants can be further developed to shorten the cold-start cycle and enhance the practicality and deployability of the algorithms in low-resource scenarios.

6. Declarations

6.1. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.2. Funding

The author received no financial support for the research, authorship, and/or publication of this article.

6.3. Institutional Review Board Statement

Not applicable.

6.4. Informed Consent Statement

Not applicable.

6.5. Declaration of Competing Interest

The author declares that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

7. References

[1] Ma, X., Li, X., Zhang, J., Ma, Z., Jiang, Q., Liu, X., & Ma, J. (2025). Repairing Backdoor Model with Dynamic Gradient Clipping for Intelligent Vehicles. IEEE Transactions on Dependable and Secure Computing, 22(1), 804–818. doi:10.1109/TDSC.2024.3419040.

[2] Mayo, S. A., Shi, X., & Hu, Q. (2023). Computational fluid dynamic study on design and modification of underwater remotely operated vehicle. Journal of Physics: Conference Series, 2591(1), 012022. doi:10.1088/1742-6596/2591/1/012022.

[3] Luo, Q., Xu, Z., Zhang, Y., Igene, M., Bataineh, T., Soltanirad, M., Jimee, K., & Liu, H. (2025). Vehicle Trajectory Repair Under Full Occlusion and Limited Datapoints with Roadside LiDAR. Sensors, 25(4), 1114. doi:10.3390/s25041114.

[4] Eddy, C. W., Castanier, M. P., & Wagner, J. R. (2024). Predictive Maintenance of a Ground Vehicle Using Digital Twin Technology. SAE Technical Papers, 2024(4), 12. doi:10.4271/2024-01-2867.

[5] Fan, G., & Jiang, Z. (2023). Approach for Scheduling Automatic Guided Vehicles Considering Equipment Failure and Power Management. Journal of Marine Science and Application, 22(3), 624–635. doi:10.1007/s11804-023-00357-3.

[6] Zhang, F., Xu, G., Li, Z., & Li, M. (2023). Vehicle-bridge Interaction Analysis and Maximum Allowable Speed of a Deployable Emergency Repair Beam. KSCE Journal of Civil Engineering, 27(10), 4332–4351. doi:10.1007/s12205-023-1735-z.

[7] Liu, Y., Zuo, X., Ai, G., & Zhao, X. (2023). A construction-and-repair based method for vehicle scheduling of bus line with branch lines. Computers and Industrial Engineering, 178. doi:10.1016/j.cie.2023.109103.

[8] Fang, J., Chen, R., Chen, J., Xu, J., & Wang, P. (2023). A multi-objective optimisation method of rail combination profile in high-speed turnout switch panel. Vehicle System Dynamics, 61(1), 336–355. doi:10.1080/00423114.2022.2052327.

[9] Kuyu, Y. Ç., & Vatansever, F. (2024). A hybrid approach of ALNS with alternative initialization and acceptance mechanisms for capacitated vehicle routing problems. Cluster Computing, 27(10), 13583–13606. doi:10.1007/s10586-024-04643-9.

[10] Wang, Y., Lin, C., Zhao, B., Gong, B., & Liu, H. (2024). Trajectory-based vehicle emission evaluation for signalized intersection using roadside LiDAR data. Journal of Cleaner Production, 440, 140971. doi:10.1016/j.jclepro.2024.140971.

[11] Dalhatu, A. A., Saad, A. M., de Azevedo, R. C., & de Tomi, G. (2023). Remotely Operated Vehicle Taxonomy and Emerging Methods of Inspection, Maintenance, and Repair Operations: An Overview and Outlook. Journal of Offshore Mechanics and Arctic Engineering, 145(2). doi:10.1115/1.4055476.

[12] Tsallis, C., Papageorgas, P., Piromalis, D., & Munteanu, R. A. (2025). Application-wise review of Machine Learning-based predictive maintenance: Trends, challenges, and future directions. Applied Sciences, 15(9), 4898. doi:10.3390/app15094898

[13] Wang, W., Wang, X., Fu, J., Lu, Z., Zhou, R., Shao, Y., Wang, J., & Morgenthal, G. (2023). A novel frame-type crashworthy device for protecting bridge piers from vehicle collisions. Structures, 57. doi:10.1016/j.istruc.2023.105313.

[14] Zhang, J., Xu, W., Wang, J., Quan, Z., Su, L., Tan, W., Chen, T., & Liu, S. (2023). Comparative analysis of testing technologies in airfield pavement emergency repair. Journal of Shenzhen University Science and Engineering, 40(6), 693–704. doi:10.3724/SP.J.1249.2023.06696.

[15] Fu, W., Li, J., Liao, Z., & Fu, Y. (2025). A bi-objective optimization approach for scheduling electric ground-handling vehicles in an airport. Complex and Intelligent Systems, 11(4), 1–27. doi:10.1007/s40747-025-01815-x.

[16] Dolatabadi, A., Abdeltawab, H., & Mohamed, Y. A. R. I. (2022). Deep reinforcement learning-based self-scheduling strategy for a CAES-PV system using accurate sky images-based forecasting. IEEE Transactions on Power Systems, 38(2), 1608-1618. doi:10.1109/TPWRS.2022.3177704.

[17] Nian, T., Wang, M., Li, S., Li, P., & Song, J. (2024). Enhancing pavement structural resilience: analyzing the impact of vehicle-induced dynamic loads on RAP-recycled cement-stabilized crushed stone pavements with tip cracks. Materials and Structures/Materiaux et Constructions, 57(7), 1–20. doi:10.1617/s11527-024-02439-2.

[18] Chi, H., Sang, H. Y., Zhang, B., Duan, P., & Zou, W. Q. (2024). BDE-Jaya: A binary discrete enhanced Jaya algorithm for multiple automated guided vehicle-scheduling problem in matrix manufacturing workshop. Swarm and Evolutionary Computation, 89, 101651. doi:10.1016/j.swevo.2024.101651.

[19] Campbell, T. M., & Trudel, G. (2024). Protecting the regenerative environment: selecting the optimal delivery vehicle for cartilage repair—a narrative review. Frontiers in Bioengineering and Biotechnology, 12. doi:10.3389/fbioe.2024.1283752.

[20] Lou, P., Sun, Z., & Su, X. (2023). Temperature effects of ballastless track and dynamic responses of vehicle-track coupling system on tunnel floor with high temperature. Structures, 50, 1391–1402. doi:10.1016/j.istruc.2023.02.103.

[21] Guo, G., Hao, C., & Du, B. (2023). Static and dynamic response characteristics of a ballastless track structure of a high-speed railway bridge with interlayer debonding under temperature loads. Engineering Failure Analysis, 151. doi:10.1016/j.engfailanal.2023.107377.

[22] Toru, E., & Yılmaz, G. (2023). A multi-depot vehicle routing problem with time windows for daily planned maintenance and repair service planning. Pamukkale University Journal of Engineering Sciences, 29(8), 913–919. doi:10.5505/pajes.2023.99569.

[23] Wang, Q., Mao, J., Wen, X., Wallace, S. W., & Deveci, M. (2025). Flight, aircraft, and crew integrated recovery policies for airlines-A deep reinforcement learning approach. Transport Policy, 160, 245-258. doi:10.1016/j.tranpol.2024.11.011.

[24] Sun, Z., Jiao, J., Li, W., Li, Z., & Li, P. (2022). A task scheduling strategy for a power cloud data center based on an improved ant colony algorithm. Power System Protection and Control, 50(2), 95–101. doi:10.19783/j.cnki.pspc.210466.