Automation and Machine Learning (2023)
Clausius Scientific Press, Canada

DOI: 10.23977/autml.2023.040106
ISSN 2516-5003 Vol. 4 Num. 1

Research on Electric Vehicle Routing Problem Based on Reinforcement Learning

Shaomin Zhang[a,*], Kunpeng Wang[b], Baoyi Wang[c]

School of Control and Computer Engineering, North China Electric Power University, Baoding, Hebei, 071003, China
[a]zhangshaomin@126.com, [b]m15287121872@163.com, [c]wangbaoyi@126.com
[*]Corresponding author

Keywords: Electric vehicle, Deep learning, Reinforcement learning, Path planning

Abstract: As an emerging means of transportation, electric vehicles have been regarded as having broad application prospects due to their advantages in energy conservation, emission reduction and carbon neutrality. However, due to the limitation of cruise range and the inconvenience of charging process, the promotion of electric vehicles is not smooth. So we introduces the application of reinforcement learning in this field and proposes a deep reinforcement learning scheme based on D3QN (Dueling Double DQN) to solve it. Finally, we compare D3QN algorithm with the current general DQN and DDQN algorithms in terms of success rate and reward value through comparative experiments.

1. Introduction

As an emerging representative of the new energy industry, electric vehicles have gained market favor and attention due to their advantages in energy conservation, emission reduction, and environmental protection.

As a well-known combinatorial optimization problem, vehicle routing problem was firstly proposed by Dantzig and Ramser [1] as a truck scheduling problem in 1959, and was proved by Lenstra and Kan to be NP-hard problem. However, most of them are targeted at specific problems and need to be solved from the beginning each time. This new method includes using model learning heuristic instead of handwriting logic currently required by existing methods. [2]The latest development of reinforcement learning and the prospect of finding new algorithms without manual heuristics make this technology an obvious candidate for solving the challenges faced by combinatorial optimization problems: scalability and versatility.

In order to resolve the dynamic random electric vehicle routing problem, literature [3] proposed a safe reinforcement learning solution, using Monte Carlo simulation to understand the random customer request and energy consumption offline, so as to plan the route safely and predictably online.

2. Related technology

2.1. Reinforcement learning

Reinforcement learning is a planning algorithm involving Markov decision process. Model based and model free reinforcement learning algorithms are the two most important types of reinforcement learning algorithms [2]. The difference between them is only whether to model the real environment. By comparison, model-free reinforcement learning is more simple and intuitive, and has better generalization.

We define some symbols in reinforcement learning as follows:

- State set S = $\{s_1, s_2, s_3, \cdots\}$
- Action set A = $\{a_1, a_2, a_3, \cdots\}$
- State transition probability function T = T$(s'|s,a) = P[S_{t+1} = s'|S_t = s, A_t = a]$, which means. The probability of transition from one state to another when taking action.
- Reward function r = r$(s,a) = E[R_{t+1}|S_t = s, A_t = a]$, This means giving the expected value of s and a rewards.
- Strategy π = π$(a|s) = P[A_t = a|S_t = s]$, which means the probability of selecting action a after given state s.
- Discount factor γϵ[0,1].

2.2. Dueling Double DQN (D3QN)

In order to reduce the overestimation of action values, the literature [5] proposed the double-depth duel Q network (D3QN) algorithm combining the double-depth Q network (DDQN) and dueling Q network (Dueling DQN). Literature [6] proposed a feasible idea of using two duel DQN networks to reduce the overestimation of action value during training.

In D3QN algorithm, the method to calculate the target value is:

$$y_t = r_{t+1} + \gamma Q(s_{t+1}, argmax_a Q(s_{t+1}, a; w_e); w_t) \tag{1}$$

Wherein, w_e represent the parameters of the evaluation network and w_t represent the parameters of the target network.

3. Scheme design

3.1. Design ideas

Markov decision process can model the uncertainty in the continuous decision of the system. Markov decision process includes action space a, state space s, reward r and state transition probability matrix p [7]. This paper randomly sets the distance and charging time between the electric vehicle and the charging pile, and designs a reward function to evaluate the generated solution.[8]

3.2. Scheme model

After referring to the electric vehicle charging algorithm based on DQL proposed in document [9], we propose an electric vehicle charging scheduling algorithm based on D3QN. Firstly, the initial state of the electric vehicle is defined as $s = [x, y, e]$, where e represents the remaining electricity of this electric vehicle, and (x, y) represents its position coordinates.

Based on these discussions, this paper uses Markov decision process to define reinforcement learning tasks, and represents them as quaternion $\langle S, A, P, R \rangle$, namely state set, action set, state transfer function and reward function. Each element is detailed as follows:

(1) State set: marked as $S_t = \{s_t^1, s_t^2, \cdots, s_t^n\}$, where vector $s_t^N = \{x, y, e_t^n\}$ represents the state of electric vehicle n at time t, $e_t^n \in (0,10)$ represents the state of battery charge of vehicle v at time t, (x, y) represents the coordinates of vehicle n at time t.

(2) Action set: recorded as discrete behavior space $A_t = \{a_t^1, a_t^2, \cdots, a_t^n\}$, A_t includes all possible behaviors. $a_t^n = \{0,1,2,3,4\}$ represents the action taken by the electric vehicle v in the state of s_t^n. 0 represents right driving, 1 represents upward driving, 2 represents left driving, 3 represents downward driving, and 4 represents charging.

(3) Reward space: defined as $R_t = \{r_1^t, r_2^t, \cdots, r_t^n\}$, Wherein, r_t^n represents the reward value of vehicle v after it acts at time t.

The following is the reward function proposed in this article:

1) If the car arrives at the fast charging pile for the first time, it will receive a+1.5 reward to encourage the car to arrive at the fast charging pile, but after many times of arrival, it will gradually reduce the reward to prevent the car from frequently arriving at the charging pile;

2) Similarly, if the car arrives at the slow charging pile for the first time, it will receive a+1.0 reward (slightly lower than the fast charging pile) to encourage the car to arrive at the slow charging pile, but after many times of arrival, it will also gradually reduce the reward to prevent the car from frequently arriving at the charging pile;

3) If the car is in the fast charging pile and its battery is low, it performs the charging action and gets a+0.5 reward to encourage the car to charge when it is out of power;

4) If the car is in a slow charging pile and its battery is low, it performs the charging action and gets a+0.25 reward to encourage the car to charge when it is out of power;

5) If the vehicle is located at a fast or slow charging pile and still performs charging action when its own power is sufficient, it will be rewarded with - 0.1 to prevent the vehicle from frequent charging;

6) If the car is not in the fast or slow charging pile, but has carried out the charging action, get - 0.1 reward to prevent the car from charging in other places;

7) If the car runs out of electricity before reaching the destination, you will get - 1.0 reward;

8) If the car does not reach the destination within 100 steps, it will get - 1.0 reward;

9) If the car successfully reaches the destination, it will receive a large positive reward (2.0 * the remaining steps);

10) If the car has sufficient power, it will receive a reward that is inversely proportional to the distance from the end (the closer it is to the end point, the greater the reward, with the maximum is 0.5), which will guide the car to the end point.

Interaction details: The amount of charging that the car gets when charging the fast charging pile is twice that of the slow charging pile, and the maximum number of steps per interaction cycle is 100.

3.3 Scheme implementation process

In order to realize deep Q learning, the first thing we need is deep Q network [10]. In order to improve efficiency, this paper needs to use a neural network with an input state and output an approximate Q value for each possible action. There are many methods for function fitting in machine learning. This paper uses a small Q network with two hidden layers.

In order to ensure that the intelligent body can explore the environment, this paper uses the "ϵ - greedy" strategy (that is, the probability of 1 - ε will determine the action according to the Q function), which can be written as

$$a = \begin{cases} \underset{a}{\text{argmax}}\, Q(s,a), \text{there is a probability of } 1 - \varepsilon \\ \qquad\qquad \text{random, otherwise} \end{cases} \tag{2}$$

Generally, ε is set to a small value. In this paper, ε =0.1 is set. The self-learning process of D3QN is completed by calculating the state s, reward r and action a, and iteratively updating the objective function until the objective function completes the learning process and reaches the optimal control. In order to maximize the charging benefits, the objective function of electric vehicle charging, namely the optimal action value function, is as follows:

$$Q(s,a) = Q(s,a) + a\left[r + \gamma \max_{a' \in A} Q(s_{i+1}, a_{i+1}) - Q(s,a)\right] \tag{3}$$

In the formula, γ is the learning rate. Here, set γ =0.001, s_{i+1}, a_{i+1} to represent the state and action of the next state. Formula (3) use time series differential learning objective

$r + \gamma \max_{a' \in A} Q(s_{i+1}, a_{i+1})$ to add Measure update $Q(s,a)$, that is to say, make $Q(s,a)$ to the time series difference error target $r + \gamma \max_{a' \in A} Q(s_{i+1}, a_{i+1})$ is close. Therefore, for a group of data $\{(s_i, a_i, r_i, s_{i+1})\}$, it is natural to construct the loss function of Q network into the form of mean square error:

$$\omega^* = \underset{\omega}{\operatorname{argmin}} \frac{1}{2N} \sum_{i=1}^{N} \left[Q_\omega(s_i, a_i) - \left(r + \gamma \max_{a' \in A} Q(s_{i+1}, a_{i+1})\right)\right]^2 \tag{4}$$

The goal of the final update of D3QN algorithm is to make $Q_\omega(s_i, a_i)$ approach

$$r + \gamma \max_{a' \in A} Q(s_{i+1}, a_{i+1}).$$

4. Scheme implementation

This section provides simulation results to test the performance of the D3QN-based deep reinforcement learning algorithm. And the code is based on Python and Python.

4.1. Simulation environment settings

The experiment randomly defines the position (two-dimensional variable) and current electric quantity (one-dimensional variable) of the electric vehicle, namely the triple (x, y, battery short), as well as the coordinates of the destination, the first fast charging pile and the two slow charging piles. There are five states of electric vehicles, including: right driving, up driving, left driving, down driving and charging. The amount of charging that the car gets when charging at the fast charging pile is twice that of the slow charging pile. The maximum number of steps per interaction cycle is 100.

In order to realize the comparison with DQN algorithm and DDQN algorithm, this paper not only trains the D3QN model, but also trains the DQN and DDQN models. The learning rate of the three models is set to 0.001, and the convergence effect of the two algorithms is tested when the number of iterations is 20000.

4.2. Comparison of results

After training the DQN, DDQN and D3QN models, we conducted an experiment to simulate the path guidance of electric vehicles, and drew the success rate curve and reward curve according to the experimental results.

The reward curve drawn by recording the reward values obtained by the three models in the iteration process is shown in Figure 1. The figure shows that the reward values obtained by DQN and DDQN are both low when they tend to converge. The reward values obtained by DDQN model are generally higher than those obtained by DQN model. Although the reward values obtained by D3QN model have been bullied, the value has been far ahead of the DQN and DDQN models. Figure 2 is the success rate curve drawn based on the success rate of the three models in the iteration process. From the success rate curve, it can be clearly seen that D3QN>>DDQN>DQN at the end of convergence. The success rate of the model using DQN is still very low even if it is iterated several times in the face of relatively complex path problems. Although DDQN has significantly improved compared with it, the range is limited, but it has only increased to 0.05. The model based on D3QN reached 0.35 after it became stable.

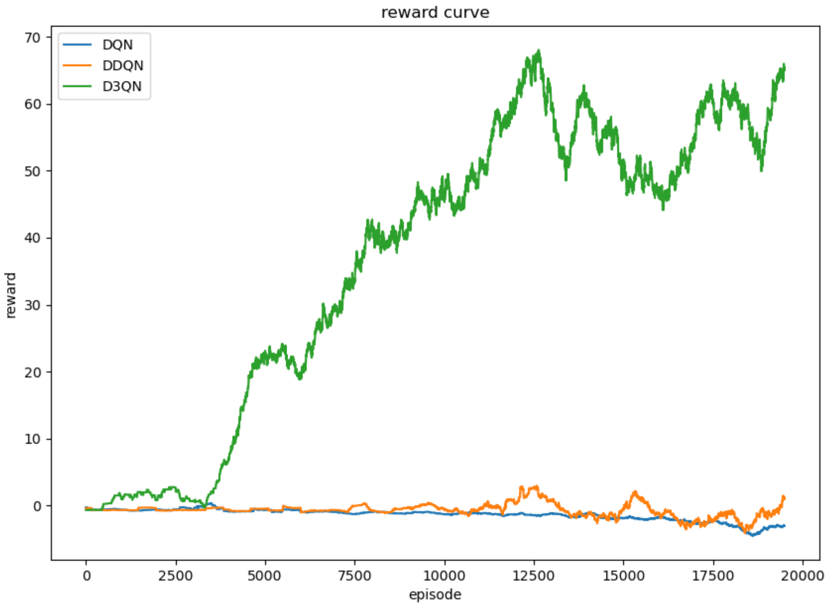

Figure 1: Reward Curve

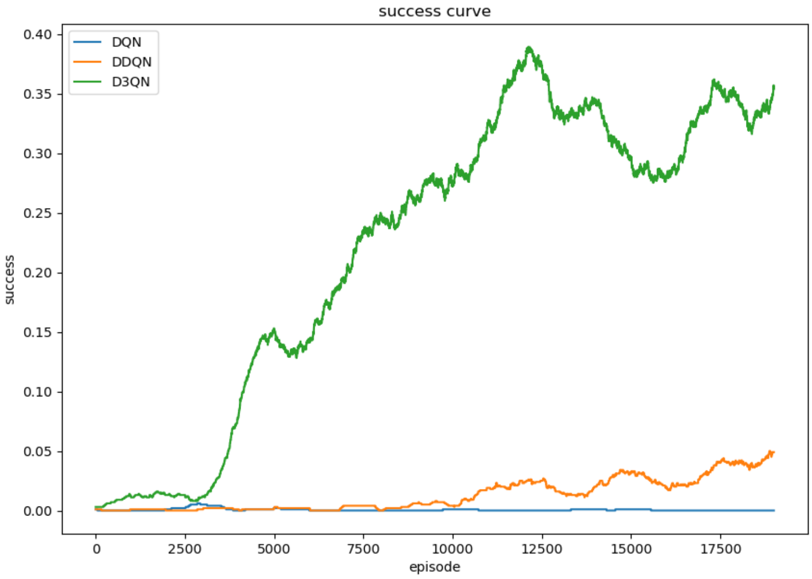

Figure 2: Success Curve

To sum up, the deep reinforcement learning method based on D3QN proposed in this paper finally learned to reach the predetermined charging strategy and achieved better performance.

5. Conclusion

In order to make a better judgment on the choice of charging pile for electric vehicles, the paper devises a deep reinforcement learning algorithm based on D3QN to train the agent to choose the route in the journey through local information. Finally, it compares it with DQN algorithm and DDQN algorithm through comparative experiments. The experiments show that D3QN algorithm has obvious advantages in improving the success rate and can obtain significantly higher reward value. In the future, this work can further study its scalability and experiments in the case of multiple electric vehicles.

References

[1] G. B. Dantzig and J. H. Ramser. The Truck Dispatching Problem [J]. Management Science, 1959, 6(1): 80-91.

[2] C. Zhang, Y. Liu, F. Wu, B. Tang and W. Fan, "Effective Charging Planning Based on Deep Reinforcement Learning for Electric Vehicles," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 1, pp. 542-554, Jan. 2021, doi: 10.1109/TITS. 2020.3002271.

[3] Rafael Basso, Balázs Kulcsár, Ivan Sanchez-Diaz, Xiaobo Qu, Dynamic stochastic electric vehicle routing with safe reinforcement learning, Transportation Research Part E: Logistics and Transportation Review, Volume 157, 2022, 102496, ISSN 1366-5545.

[4] Q. Zhang, K. Wu and Y. Shi, "Route Planning and Power Management for PHEVs With Reinforcement Learning," in IEEE Transactions on Vehicular Technology, vol. 69, no. 5, pp. 4751-4762, May 2020, doi: 10.1109/TVT. 2020.2979623.

[5] M. Ye, C. Tianqing and F. Wenhui, "A single-task and multi-decision evolutionary game model based on multi-agent reinforcement learning," in Journal of Systems Engineering and Electronics, vol. 32, no. 3, pp. 642-657, June 2021, doi: 10.23919/JSEE. 2021.000055.

[6] Y. Huang, G. Wei and Y. Wang, "V-D D3QN: the Variant of Double Deep Q-Learning Network with Dueling Architecture," 2018 37th Chinese Control Conference (CCC), 2018, pp. 9130-9135, doi: 10.23919/ChiCC. 2018.8483478.

[7] Queck B., Lau H. C. (2020). A Genetic Algorithm to Minimise the Number of Vehicles in the Electric Vehicle Routing Problem. In: Lalla-Ruiz, E., Mes, M., Voß, S. (eds) Computational Logistics. ICCL 2020. Lecture Notes in Computer Science, vol 12433. Springer, Cham.

[8] Ben Abbes S., Rejeb L., Baati L.: Route planning for electric vehicles. IET Intell. Transp. Syst. 16, 875– 889 (2022).

[9] C. Zhang, Y. Liu, F. Wu, B. Tang and W. Fan, "Effective Charging Planning Based on Deep Reinforcement Learning for Electric Vehicles," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 1, pp. 542-554, Jan. 2021, doi: 10.1109/TITS. 2020.3002271.

[10] R. Szczepanski, T. Tarczewski and K. Erwinski, "Energy Efficient Local Path Planning Algorithm Based on Predictive Artificial Potential Field," in IEEE Access, vol. 10, pp. 39729-39742, 2022, doi: 10. 1109/ ACCESS. 2022. 3166632.