Charging Station Recommendation based on Mean Field Multi-agent Reinforcement Learning

Kaili Cui[1, 2], Youchan Zhu[1, 2, *], Yulong Chang[1, 2], Jinlei Qin[1, 2], Zheng Li[1, 2]

[1]School of Control and Computer Engineering, North China Electric Power University, Baoding 071003, China

[2]Engineering Research Center, Ministry of Education, Complex Energy System Intelligent Computing (North China Electric Power University), Baoding 071003, China

*zyc_hd@sina.com

Abstract

With the increase of charging stations and electric vehicles, the driver's decision-making without fixed rules would exacerbate the unbalanced utilization of charging piles, resulting in an increase in the time cost of charging. To reduce the overall charging waiting time, this paper proposes a charging station recommendation framework based on multi-agent reinforcement learning for complex charging scenarios, which can be extended to scenarios with a variable number of agents and low delay requirements. First of all, the state and reward are defined by taking electric vehicles as an agent and combining the characteristics of electric vehicles and charging stations. Secondly, to reduce the recommendation delay, an actor-critic algorithm based on mean field theory is designed, and a distributed decision is adopted to make recommendations for multiple charging requests simultaneously. In this way, each agent can take full advantage of future data when training the Q network. Finally, considering the influence of state transition time on the recommendation results, three state spaces with different time steps are proposed in the simulation experiment to obtain the optimal time steps, and the results are compared with the shortest distance method using sequential decision making, single-agent, and multi-agent algorithm. Experimental results show that the proposed algorithm has the best performance.

Keywords

Multi-agent Reinforcement Learning; Mean Field Theory; Distributed Decision Making; Real-Time Recommendation; Charging(Batteries).

1. Introduction

The development of electric vehicles has emerged as a crucial initiative for sustainable development and to relieve the pressure of the energy crisis and the environment in light of the global environmental degradation. According to the China Charging Alliance, there were 3.368 million electric vehicles on the road in 2019 and 516,000 public charging stations. The manufacturing of new energy cars surged by 194.9 percent yearly from January to July 2021. The number of charging piles will increase more quickly due to the rapid development of new energy vehicles and the national initiative to accelerate the building of charging piles for the "new infrastructure".

At the same time, many domestic charging station operators such as Tesco, Star Charging ,and third-party platforms such as Baidu and Amap have also introduced information of mainstream charging stations in the market. However, faced with numerous charging stations, drivers tend to make choices based on own habits or blindly, thus making charging completion take more time and charging costs. Electric vehicle users prefer charging platforms to guide them to make the best choice [1].

The recommendation of charging stations from the perspective of the user has been the subject of much research. They have taken into account a number of variables to improve the efficacy of the recommendations, such as user preference, travel cost, traffic conditions, and time. For example, in order to determine the user's preference for charging stations, Bu et al. [2] employed a collaborative filtering algorithm. This information served as the foundation for their recommendation. Wang et al. [3] used a factorization machine approach to predict recommendation results and combined federal learning to improve cross-platform data security. Jia et al. [4]'s method of cab trajectory prediction allowed them to select the charging station that would travel the least distance between the starting point and the intended destination. However, these studies do not take into account the impact between vehicles and charging stations at different times. Not considering the charging intentions of other users may lead to longer queues at charging stations for electric vehicles [1,5]. To address this problem, Wang et al. [1] used Pareto optimality to recommend charging stations for a group of EVs in a short period of time, resulting in an overall reduction in queuing time. The queuing up time prediction algorithm does not account for users who arrive at the charging station directly without sending a charging request although this method predicts numerous charging requests in a minute. A charging mechanism created by Zhang et al. [6] continuously monitors the condition of charging stations and changes the suggested stations list in real-time. In order to manage cars with various priority, Cao et al. [7] employed information on vehicle reservations. The information on electric vehicles and charging stations does, however, change frequently over time, making it extremely difficult for communication to continuously detect this information and feedback.

Recently, reinforcement learning (RL) has been applied to games, transportation, and other fields due to its ability to effectively solve sequential decision problems in complex environments, and has been effective in autonomous driving [8,9] and vehicle order scheduling [10,11]. In contrast to charging time prediction, which requires more assumptions and rules, reinforcement learning will fully consider the impact of current decisions on the future, i.e., to maximize the expected cumulative payoff, interact directly with the dynamically changing complex environment, and train using historical data with real-time data to obtain the overall optimal policy.

However, the following problems occur when reinforcement learning is used to make recommendations for charging stations: consider L charging stations where the state space size is $S = S_1 \times S_2 \times \cdots \times S_L$ and the action space size is $A = A_1 \times A_2 \times \cdots \times A_L$, meaning that the space size is exponential. The charging environment in a city with many charging stations has a wide state and action space, which is not good for the stability of network training. Zhou et al. [12] described the charging station recommendation problem as a single-agent action-value function learning task using an improved DQN (Deep Q-Networks) algorithm that takes into account information about surrounding charging stations when estimating the value function, utilizes graph convolutional neural networks for training, and reduces the state information input dimension. Nevertheless, in addition to the state and action high-dimensional problem, another major issue of directly learning a centralized agent system is the high latency associated with obtaining the overall state data and handing it off to the agents for computation, which is not suitable for large-scale charging scenarios that request for real-time recommendations.

In large-scale environments, multi-agent reinforcement learning (MARL) can reduce latency [13,14]. In [15], a distributed training method with performance comparable to centralized training was developed to address the central server congestion problem by sharing parameters only with the neighboring agents during the training process. Wu et al. [16] designed a distributed computing architecture to reduce the network latency in the Nash actor-critic algorithm-based traffic signal control. Chu et al. [17] added a long and short-term memory network to the network structure of the value function, using historical data and the current state as input, to improve the stability of training.

Zhang et al. [18] treated each charging station as an independent agent and considers EV charging recommendation as a multi-objective optimization task. Each autonomous agent has a constant-level action space, which can be expanded to include settings with greater complexity, in this way. However, for the case of multiple requests in a short period time, the independent recommendation strategy of each charging station is still essentially a centralized sequential decision, which is difficult to process in parallel, i.e., it cannot take into account the actions taken by other charging requests in the same state at the same time, prolonging the wait time for EVs in the case of multiple requests in a short period of time.

In this paper, a distributed multi-agent reinforcement learning model is designed. The overall goal is to minimize the overall driving time and queuing time at charging stations in a day. Present a distributed multi-agent reinforcement learning framework, to request per minute charging electric cars as the agent, on the one hand, can take into account the future behavior of the agent, on the other hand, can coordinate cooperation between multiple agents in order to reduce decision time delay, using distributed decision-making method, each agent chooses according to their local observations charging stations. Mean field theory is employed concurrently to address the problem of the variable number of agents.

2. Charging Environment

The first part of this section describes the procedure from charging request through charging completion. The fundamental components of multi-agent reinforcement learning for charging environments are described in the second part.

2.1 Charging Process

In continuous time, the moments when the vehicle sends a charging request and the state of the charging station is bound to change are called "charging important time points", and the whole charging process is described by these moments. As shown in Fig. 1: At the moment T0, the user has a charging demand and sends a charging request to the platform to go to a recommended charging station or chooses a charging station according to his habits. At the moment of T2, two possible events will happen: (1) the electric car leaves without charging due to the long queue time; (2) in the second case: there are free charging piles, and the electric car starts charging and leaves at the moment of T3.

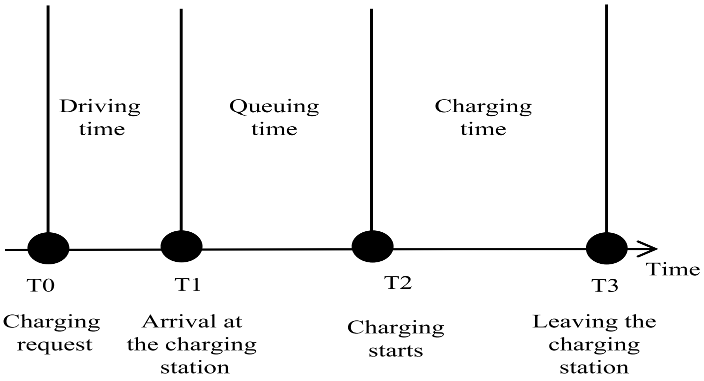

Figure 1. Charging process of electric vehicle

2.2 MARL Model for Charging Environments

$Q_t^i = (loc_t^i, time_t^i, qid_t^i)$ is defined as a charge request at time t, including the request location, request time, and remaining power.

Denote the sum of travel time and waiting time from Q_t^i to the target charging station by $time\_cost$, i.e., $time\_cost = T2 - T1$. To minimize the overall $time\_cost$, the charging recommendation task is described as a multi-agent reinforcement learning problem:

Agent: In 75% of the day, a city-scale charging environment may experience more than 10 charging events every minute [1], necessitating the need for several charging recommendations quickly. Each minute's worth of charge requests from vehicles are regarded as agents, and recommendations are given for each Q_t^i . Each agent is identified as $agent_i, i=1,2,\cdots m$,and there are m charging requests each minute.

State: The definition of the state of a charging station is $S_{cs}=(O_{cs}^1,\cdots,O_{cs}^i,\cdots,O_{cs}^k), i=1,2,\cdots k$, where O_{cs}^i is the observed value of each charging station, which includes its charging power, the number of charging requests at nearby charging stations, and the number of charging points that are now available.

Each agent's anticipated arrival time at each charging station is given by the following $S_v=(S_v^1,\cdots,S_v^i,\cdots,S_v^m), i=1,2,\cdots,m$ .The use of charging stations after T1 determines how long electric car queues will last, assuming that the agent's recommended charging station is cs . Since the precise arrival time at T0 is unknown, the number of charging stations that are still available at cs 30 minutes after T0 is taken as an additional observation, designated as $S_{fur}=(t_{nur}^1,\cdots,t_{nur}^i,\cdots,t_{nur}^m), i=1,2,\cdots,m$ .The overall state of the charging environment is denoted as $S=\left\{S_{cs},S_v,S_{fur}\right\}$.

Action: For each Q_t^i , all agents can choose any charging station. If the number of charging stations is u ,the set of actions for each Q_t^i is $A=\{1,2,\ \cdots, u\}$. When $agent_i$ selects the first charging station j , the action is denoted as $a_i=j$.

Reward: After the driver arrives at the recommended charging station, if $time\_cost$ is less than 1 hour and leaves after the EV charging is completed, the EV charging is successful, otherwise, the charging fails. The maximum reward setting is 60 minutes. The reward function is defined as:

$$reward\_cwt=\begin{cases}\dfrac{(-time\_\cos t+60)}{60}, & \text{success}\\ 0, & \text{failure}\end{cases} \tag{1}$$

3. Multi-agent Reinforcement Learning Framework based on Mean Field Theory

3.1 Centralized Training Decentralized Execution Framework

The Centralized training decentralized Execution (CTDE) framework [19,20] uses information from other agents during training .It utilizes only the local states observed by itself when executing actions, which significantly reduces the state space. The framework has the advantage of distributed execution and is easy to deploy to practical applications. During the training process, CDTE can coordinate the communication and cooperation among agents using more comprehensive state information, actions of other agents ,and future information, and thus learn the action-value function effectively. When using the policy network to select actions, each agent uses only its observed local environment state without global information. This decentralized execution method can reduce real-time recommendation latency and improve recommendation efficiency.

3.2 Distributed Decision Making

The time interval between two adjacent charging requests is short during the whole charging process in a day, and this phenomenon increases significantly during peak charging periods. Most of the previous studies are based on a first-request-first-service strategy [21,22] and do not consider the important impact of decision order in the execution of intensive actions. For example, there are three charging requests (q_1,q_2,q_3) in a short period of time Δt .

The recommendation results, with a total decision-making time of 60 minutes, are presented in Fig. 2 when the choice is made in the order in which the request was made. If the decision on the recommendation for the three requests in Δt is made simultaneously, q_3 will be assigned to charging station 1, which will take 15 minutes; q_1, which will take 20 minutes; and q_2, which will take 15 minutes, for a total of 50 minutes.

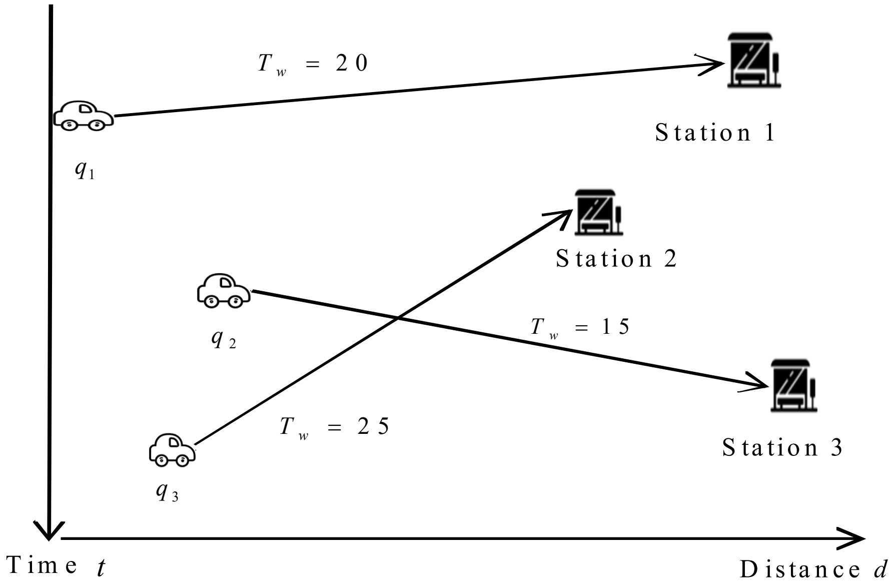

Figure 2. Recommendation results based on first-request-first-serve decision

Therefore, to improve the recommendation speed and reduce the overall time, a plurality of charging request vehicles in the Δt truck is used as agents. Because the number of charging requests is different and the action space is different in different states, each agent shares the same action-state value function network and policy network. At the same time, the mean field multi-agent reinforcement learning algorithm [23] is used to approximate the expected reward of each agent by averaging the action value of other agents.

In a multi-agent system, the agents make decisions simultaneously for multiple requests within Δt, i.e., the problem to be solved is the allocation of resources to achieve the shortest overall time task.

3.3 Recommendation of Charging Stations with Mean Field Approximation

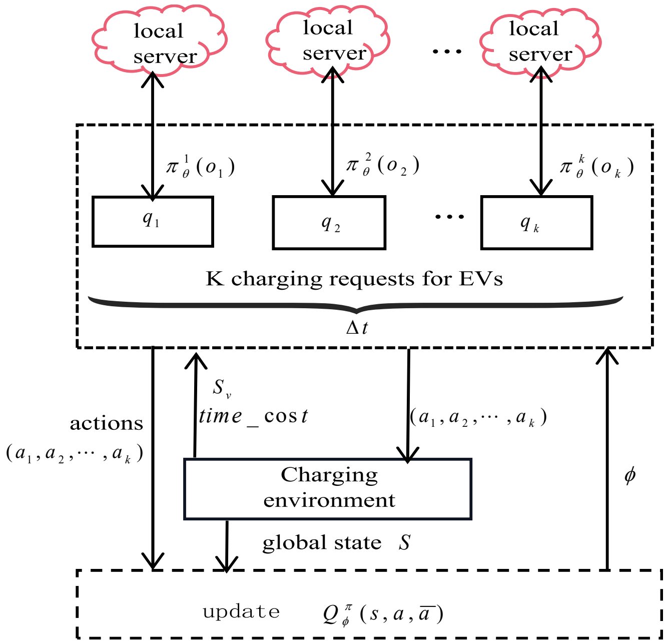

Figure 3. Distributed multi-agent reinforcement learning framework

This section presents a charging station recommendation algorithm using mean field theory (CSMF). Each agent's action-value function Q uses the global state, action, and average action values of other agents. Fig. 3 is a distributed multi-agent reinforcement learning framework. When making recommendations for electric vehicles, there is no need for unified calculation by the central server, only real-time data of electric vehicles are needed to calculate the recommendation results.

The critic's update is achieved by minimizing the loss function $L(\phi)$ neighbors:

$$\begin{aligned} L(\phi) &= \mathrm{E}_{(s,a,\bar{a},r,s')\sim D}[y - Q_{\phi}^{\pi}(s,a,\bar{a})]^2 \\ &\approx \frac{1}{N}\sum[y - Q_{\phi}^{\pi}(s,a,\bar{a})]^2 . \end{aligned} \tag{2}$$

$$y = time\_\cos t + \gamma Q_{\phi'}^{\pi'}(s',a',\bar{a}')\Big|_{a'=\pi'(o')} \tag{3}$$

Where ϕ is the parameter of the current Q function ,i.e.,$Q_{\phi}^{\pi}(s,a,\bar{a})$, π is the current policy, and ϕ' is the parameter of the target Q function , i.e.,$Q_{\phi'}^{\pi'}(s',a',\bar{a}')$. y is the estimated target value, and a' is obtained by the target policy π' . $\bar{a}$ is the mean action of neighboring agents. We define the action of $agent_i$ as a_i ,then the mean action of the other agents is as follows:

$$\bar{a}_{\ i} = \frac{1}{N_i}\sum_{j} a_j \tag{4}$$

where $\sum_j a_j$ is the sum of the action values corresponding to the charging stations selected by the other agents, and $\frac{1}{N_i}$ is the number of charging requests (except $agent_i$) in a short period of time Δt .

During the policy parameter learning process, each agent is trained based on its observed local state, without information from other agents. The strategy update uses the stochastic gradient descent method:

$$\begin{aligned} \nabla_{\theta} J(\theta) &= \mathrm{E}[\nabla_{\theta}\pi(o)\nabla_{a} Q_{\phi}^{\pi}(s,a,\bar{a})\big| a = \pi_{\theta}(o)] \\ &\approx \frac{1}{N}\sum[\nabla_{\theta}\pi(o)\nabla_{a} Q_{\phi}^{\pi}(s,a,\bar{a})\big| a = \pi_{\theta}(o)] \end{aligned} \tag{5}$$

Where θ is the parameter of the current strategy $\pi_{\theta}(o)$. N is the number of a minibatch for gradient descent. The online policy is used to explore the action during training

Table 1. Algorithm 1: CSMF

```
Initialize parameters $\phi, \phi', \theta, \theta'$, replay buffer $D$ ,episode $Ep = 0$
Input the parameters $\tau$ used to update the target function Q with the target policy function $\pi$ , $\Delta t$
For each episode:
Initialize state $s$
For each agent $agent_i$, sample action $a_i = \pi_{\theta}(o_i)$ ,get the actions of all agents $\mathbf{a}$
Compute the mean action $\bar{\boldsymbol{a}} = [\bar{a}_1, \bar{a}_2, \cdots, \bar{a}_m]$ by Eq. (4)
```

```
After each agent selects a charging station, it is rewarded with $\mathbf{r}=[r_1,r_2,\cdots,r_m]$ ;State transition, get the next state $s'$ .
Store $(s,\mathbf{a},\bar{\mathbf{a}},\mathbf{r},s')$ in $D$
Take the next state as the new state: $s \leftarrow s'$
Update the critic network by minimizing the loss Eq. (2)
Update the actor network using stochastic gradients Eq. (5)
Update the parameters of the target critic networks:
        $\phi' = (1-\tau)\phi' + \tau\phi$ , $\theta' = (1-\tau)\theta' + \tau\theta$
```

4. Experiment

4.1 Data Description

The number of charging stations is fixed at 10, and the area of 100 km^2 is divided into 100 grids of 1 km^2. A grid unit is occupied by each charging station. The number of charging requests every minute for each grid is determined by the Poisson distribution, and the time of day is divided into 1440 minutes. The training set for the electric vehicle charging suggestion simulator developed in this study consists of 30 days of operation, and the testing set consists of 10 days of operation.

4.2 Evaluation Metrics

Take into account q as a collection of charging requests that follow our advice and are successfully charged; q_{num} is the quantity of q, and M is the collection of requests. The waiting time for each request is $Wt(q)$. The average waiting time for all charging requests is measured in minutes to determine the overall waiting time for charging.

$$Mwt = \frac{\sum_{q\in M} Wt(q)}{q_{num}} \tag{6}$$

4.3 Algorithm

In CSMF, a mean field multi-agent reinforcement learning algorithm based on the actor-critic framework, the Q network builds a five-layer fully connected network using the ReLU activation function. The policy network uses a three-layer fully connected network and the output layer uses the SoftMax activation function. Meanwhile, with the increase of Δt, the number of agents and the future charging environment information change more. To get a better result for the CSMF algorithm, three different Δt ,i.e. $\Delta t = 1$, $\Delta t = 5$, and $\Delta t = 10$, are set to compare the three and choose the best Δt value.

Table 2 shows the test results of the CSMF algorithm for different Δt (in minutes). Where $hour\_mwt$ denotes the average waiting time in a certain two hours of the day, and T_{rate} denotes the ratio of the difference between the maximum and minimum values of $hour\_mwt$ at different Δt in that period time, as shown in equation (7). The performance gets better as t decreases, with $\Delta t = 1$ having the least $hour\_mwt$ in all periods. T_{rate} is the smallest between 12:00 and 14:00 during the charging congestion time. This demonstrates the CSMF algorithm's robustness in relieving charging congestion scenarios.

$$T_{rate}(t) = \frac{\underset{\Delta t}{Max}(hour\_mwt_t) - \underset{\Delta t}{Min}(hour\_mwt_t)}{\underset{\Delta t}{Max}(hour\_mwt_t)} \tag{7}$$

Table 2. The $hour\_mwt$ of CSMF at different Δt

Period of time	$\Delta t = 1$	$\Delta t = 5$	$\Delta t = 10$	T_{rate} (%)
6:00-8:00	**3.65**	4.76	6.47	43.6
8:00-10:00	**4.82**	7.74	12.66	61.9
10:00-12:00	**6.03**	9.59	16.92	64.4
12:00-14:00	**38.53**	39.73	41.39	**6.9**
16:00-18:00	**4.31**	6.74	9.64	55.3

The CSMF algorithm with Δt set to 1 is compared with the following three algorithms:

Nearest makes recommendations for electric vehicles based on the rule of choosing the closest charging station.

DQN [24] is a centralized deep Q-network approach where all charging stations are controlled by a centralized agent. DQN makes recommendations based on the state of all charging stations. The Q function in the experiments is a 3-layer fully connected network with hidden layers of dimension 256, using the ReLU activation function. The replay buffer size is 10000 and the batch size is 100. the learning rate is set to 0.0001. a greedy policy is used to select the actions.

MADDPG [25] is an effective multi-agent collaborative MARL algorithm. Each charging station is treated as a Q_t^i in the experiment. All the agents make recommendations for that Q_t^i simultaneously in the chronological order of charging requests actors act according to their own specific observations, but critics have access to the full state and joint actions in training. The Q-function network consists of a four-layer fully connected network with a hidden layer of dimension 256, using the ReLU activation function. The policy function network uses a three-layer fully connected network with a tanh activation function for the output layer. To extend the MADDPG to a large-scale charging environment, the critic network is shared among all the agents.

While CSMA utilizes distributed decision making to make suggestions simultaneously, Nearest, DQN, and MADDPG all use sequential decision making to make recommendations for each charging request.

Fig. 4 shows the process of the training phase: each algorithm interacts with the charging environment during the training process. The number of available charging posts at the charging station changes during this process with the actions selected by the agents. $Tmwt$ denotes Mwt in a day. Nearest is similar to Mwt for DQN. Among the reinforcement learning algorithms, the single agent DQN algorithm based on centralized learning performs the worst. The single agent DQN algorithm based on centralized learning performs the worst among reinforcement learning algorithms. The multi-agent reinforcement learning algorithm based on centralized training MADDPG and CSMA not only uses the current state but also adds the future data of the charging station; as a result, it performs better overall than DQN and has a shorter average waiting time. The CSMA algorithm performs the best since it simultaneously considers the actions of other charge requests.

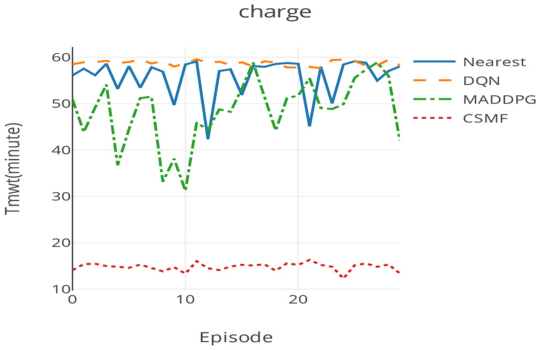

Figure 4. *Tmwt* during training for all algorithms

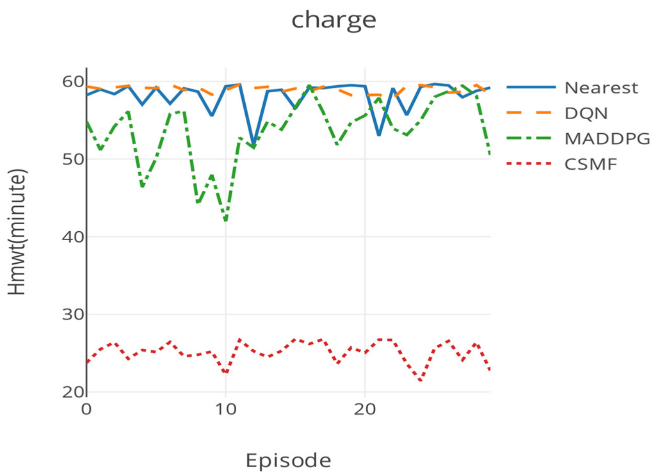

Figure 5. *Hmwt* during training for all algorithms

Table 3 shows the average waiting time for each charging request during the test phase. In order to compare the performance of the three algorithms, the environment is initialized with the same random seed, and the results are shown in Table 3. Multi-agent reinforcement learning algorithm achieves better results than Nearest and DQN. The *Tmwt* and *Hmwt* of the CSMA algorithm are reduced by 71.8 and 54.8 percent, respectively, when compared to MADDPG, showing that distributed decision making can significantly enhance the recommendation effect.

Table 3. Overall performance of each algorithm

Performance	*Tmwt* (minute)	*Hmwt* (minute)
Nearest	53.56	57.76
DQN	59.96	59.98
MADDPG	53.28	57.00
CSMF	15.05	25.76

5. Related Works

5.1 The Training Framework of Reinforcement Learning

There are now a number of popular agent training frameworks. A fully centralized training framework is the first. For instance, the CommNet suggested in [26] employed a central controller to manage all of the agents' actions. The controller is made up of a multi-layer neural network, which inputs the state of every agent, outputs every agent's action, and facilitates agent communication. A policy network and a Q network control all agents in the Bidirectionally-Coordinated Net (BiCNet) that Peng et al. [27] suggested. The second is a fully decentralized framework. Mnih et al. [28] designed a completely asynchronous parallel agent training method to speed up the training speed and applied it to Sarsa, Q-learning, and Actor-Critic single-agent reinforcement learning algorithms. Wen et al. [29] proposed a decentralized multi-agent reinforcement learning framework in which each agent finds its own best response according to the opponent's strategy. Tian et al. [30] used Kullback-Leibler (KL) divergence to model the opponent to improve the training performance of multi-agent. The third is an effective framework for decentralized execution and centralized training. This framework is more suited for multi-agent reinforcement learning tasks with a large state space and a non-stationary environment when compared to the other two frameworks. Based on this, certain studies have improved more successfully. Foerster et al. [31] proposed a counterfactual multi-agent (COMA). In order to reduce the noise in calculating the gradient, the CTDE framework is used to train to take into account the impact of the behavior of each agent on the global reward. To arrive at the best strategy for decentralized execution, Mahajan et al. [32] developed a novel action exploration method based on CTDE. In [33], a more all-encompassing method of value function decomposition is put out that may be applied to a wider variety of tasks.

5.2 Charging Station Recommendation

A significant portion of the associated research on the recommendation of charging stations is based on the algorithm in the recommendation system and utilizes the charging station attributes for driver preference recommendations. For example, in some studies [2,6], a collaborative filtering algorithm is used to calculate user preferences, and Wang et al. [3] used the factorization machine method. In [4], the recommended criterion with the shortest distance is adopted. The other part is based on the recommendation with the shortest waiting time. Related studies have taken into account the behavior of other electric vehicles [1,5,7]. In the large-scale and ever-changing actual charging environment, the effect of the reinforcement learning method is better [12,18].

6. Conclusion

In order to reduce the total amount of charging waiting time each day, we investigate the problem of recommending charging stations in this research. By using the finished training policy network to find recommended charging stations in a simulated charging environment, the superiority of the CSMF algorithm is demonstrated. The CSMF algorithm is much less in *Tmwt* and *Hmwt* than Nearest, DQN, and MADDPG. Personalized recommendations for charging stations will be made in the next work while taking user preferences into account.

References

[1] Guang Wang, Yongfeng Zhang, Zhihan Fang, et al. Fair Charge: A data-driven fairness-aware charging recommendation system for large-scale electric taxi fleets[C]//Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2020: 1–25.

[2] Fanpeng Bu, Shiming Tian, Jingjing Gao, et al. Recommendation method for electric vehicle charging based on collaborative filtering [J]. Science and Technology Review,2017,35(21):61-67.

[3] X Wang, X Zheng, X Liang. Charging station recommendation for electric vehicle based on federated learning[C]//Journal of Physics: Conference Series. IOP Publishing, 2021, 1792(1): 012055.

[4] Jian Jia, Linfeng Liu, Jiagao Wu. Charging pile recommendation method for idle electric taxis based on recurrent neural network [J]. Chinese Journal of Network and Information Security,2020,6(06):152-163.

[5] Z Tian, T Jung, Y Wang, et al. Real-time charging station recommendation system for electric-vehicle taxis[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(11): 3098-3109.

[6] T Zhang, L Zheng, Y Jiang, et al. A method of chained recommendation for charging piles in internet of vehicles[J]. Computing, 2021, 103(2): 231-249.

[7] Y Cao, T Jiang, O Kaiwartya, et al. Toward Pre-Empted EV Charging Recommendation Through V2V-Based Reservation System[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019,51(5), 3026-3039.

[8] M Zhou, J Luo, J Villella, et al. Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving[J] ,2020, arXiv preprint arXiv:2010.09776.

[9] P Palanisamy. Multi-agent connected autonomous driving using deep reinforcement learning [C]// International Joint Conference on Neural Networks (IJCNN), 2020: 1-7.

[10] Minne Li, Zhiwei Qin, Jiao Yan, et al. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning[C]//In The World Wide Web Conference, 2019: 983–994.

[11] K Lin, R Zhao, Z Xu, et al. Efficient large-scale fleet management via multi-agent deep reinforcement learning[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018: 1774-1783.

[12] C Blum, H Liu, X Hui. CoordiQ : Coordinated Q-learning for Electric Vehicle Charging Recommendation [J]. arXiv preprint arXiv:2102.00847, 2021.

[13] Y Shao, R Li, B Hu, et al. Graph attention network-based multi-agent reinforcement learning for slicing resource management in dense cellular network[J]. IEEE Transactions on Vehicular Technology, 2021, 70(10): 10792-10803.

[14] Q Wang, X Li, S Jin, et al. Hybrid beamforming for mmWave MU-MISO systems exploiting multi-agent deep reinforcement learning[J]. IEEE Wireless Communications Letters, 2021, 10(5): 1046-1050.

[15] B Liu, Z Ding. A distributed deep reinforcement learning method for traffic light control[J]. Neurocomputing, 2022, 490: 390-399.

[16] Q Wu, J Wu, J Shen, et al. Distributed agent-based deep reinforcement learning for large scale traffic signal control [J]. Knowledge-Based Systems, 2022,241: 108304.

[17] T Chu, J Wang, L Codecà et al. Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control [J]. IEEE Transactions on Intelligent Transportation Systems,2020,21(3): 1086-1095.

[18] Weijia Zhang, Hao Liu, Fan Wang, et al .Intelligent Electric Vehicle Charging Recommendation Based on Multi-Agent Reinforcement Learning[C]// Proceedings of the Web Conference 2021,2021:1856-1867.

[19] P. Hernandez-Leal, B Kartal, M E. Taylor A survey and critique of multiagent deep reinforcement learning[J]. Autonomous Agents and Multi-Agent Systems, 2019,33(6): 750-797.

[20] T Rashid, M Samvelyan, C Schroeder, et al. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning[C]//In International Conference on Machine Learning. PMLR,2018:4295–4304.

[21] Y Cao, N Wang, G Kamel ,et al. An electric vehicle charging management scheme based on publish/subscribe communication framework[J]. IEEE Systems Journal, 2017, 11(3):1822–1835.

[22] Y Cao, O Kaiwartya, R Wang, et al. Towards efficient, scalable and coordinated on-the-move EV charging management[J]. IEEE Wireless Communications., 2017, 24(2):66–73.

[23] Y Yang, R Luo, M Li, et al. Mean field multi-agent reinforcement learning[C]//In International Conference on Machine Learning .PMLR, 2018:5571–5580.

[24] V Mnih, K Kavukcuoglu, D Silver, et al. Human-level control through deep reinforcement learning[J]. Nature,2015,518(7540):529-533.

[25] R Lowe, Y Wu, A Tamar, et al. Multi-agent actor-critic for mixed cooperative-competitive environments [J]. Advances in neural information processing systems,2017,30.

[26] S Sukhbaatar, R Fergus. Learning multiagent communication with backpropagation[J]. Advances in neural information processing systems, 2016, 29.

[27] Peng Peng, Ying Wen, Yaodong Yang, et al. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play StarCraft combat games[J]. arXiv preprint arXiv:1703.10069, 2017.

[28] V Mnih, A P Badia, M Mirza, et al. Asynchronous methods for deep reinforcement learning [C]// International conference on machine learning. PMLR, 2016: 1928-1937.

[29] Y Wen, Y Yang, R Luo, et al. Probabilistic recursive reasoning for multi-agent reinforcement learning[J]. arXiv preprint arXiv:1901.09207, 2019.

[30] Z Tian, Y Wen, Z Gong, et al. A regularized opponent model with maximum entropy objective[J]. arXiv preprint arXiv:1905.08087, 2019.

[31] J Foerster, G Farquhar, T Afouras, et al. Counterfactual multi-agent policy gradients[C]//Proceedings of the AAAI conference on artificial intelligence. 2018, 32(1).

[32] A Mahajan, T Rashid, M Samvelyan, et al. Maven: Multi-agent variational exploration[J]. Advances in Neural Information Processing Systems, 2019, 32.

[33] K Son, D Kim, W J Kang, et al. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning[C]//International conference on machine learning. PMLR, 2019: 5887-5896.