

Power Flow Adjustment for Smart Microgrid Based on Edge Computing and Deep Reinforcement Learning

Tianjiao Pu (✉ tjpu@epri.sgcc.com.cn)

China Electric Power Research Institute

Fei Jiao

China Electric Power Research Institute

Yifan Cao

Tianjin University <https://orcid.org/0000-0003-2436-7790>

Zhicheng Liu

Tianjin University

Chao Qiu

Tianjin University

Ji Qiao

China Electric Power Research Institute

Shuhua Zhang

China Electric Power Research Institute

Research Article

Keywords: Power flow calculating, machine learning, edge intelligence

Posted Date: March 11th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-268782/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

RESEARCH

Power Flow Adjustment for Smart Microgrid Based on Edge Computing and Deep Reinforcement Learning

Tianjiao Pu^{1*}, Fei Jiao¹, Yifan Cao², Zhicheng Liu², Chao Qiu², Ji Qiao¹ and Shuhua Zhang¹

*Correspondence:

tjpu@epri.sgcc.com.cn

¹China Electric Power Research Institute, Haidian District, 100192 Beijing, China

Full list of author information is available at the end of the article

Abstract

As one of the core components that improve transportation, generation, delivery, and electricity consumption in terms of protection and reliability, smart grid can provide full visibility and universal control of power assets and services, provide resilience to system anomalies and enable new ways to supply and trade resources in a coordinated manner. In current power grids, a large number of power supply and demand components, sensing and control devices generate lots of requirements, e.g., data perception, information transmission, business processing and real-time control, while existing centralized cloud computing paradigm is hard to address issues and challenges such as rapid response and local autonomy. Specifically, the trend of micro grid computing is one of the key challenges in smart grid, because a lot of in the power grid, diverse, adjustable supply components and more complex, optimization of difficulty is also relatively large, whereas traditional, manual, centralized methods are often dependent on expert experience, and requires a lot of manpower. Furthermore, the application of edge intelligence to power flow adjustment in smart grid is still in its infancy. In order to meet this challenge, we propose a power control framework combining edge computing and machine learning, which makes full use of edge nodes to sense network state and power control, so as to achieve the goal of fast response and local autonomy. Furthermore, we design and implement parameters such as state, action and reward by using deep reinforcement learning to make intelligent control decisions, aiming at the problem that flow calculation often does not converge. The simulation results demonstrate the effectiveness of our method with successful dynamic power flow calculating and stable operation under various power conditions.

Keywords: Power flow calculating; machine learning; edge intelligence

Introduction

With the continuous development of power grid construction, a large number of terminal devices will be connected to the power grid, and a large number of heterogeneous transmission will be generated. The demand of data analysis and processing puts forward higher requirements to the power system data processing and business operation ability. In the face of demand response and real-time interaction of power flow, information flow and control flow in power system, there are a series of shortcomings in scalability, utilization efficiency and deployment cost of fixed redundant allocation of computing resources in the past. In contrast to the cloud computing model of centralized control in the acquisition, computing and transmission of the

various pressures. Edge computing can realize real-time and efficient perceptual response, reduce the load of central site resources, and support regional autonomy, which can effectively respond to the trend of intelligent power grid.

As an important research problem in the smart grid, power flow calculation according to the given power grid structure and operation parameters such as power supply, to determine the power system steady state parameter calculation, parts can analyze the power supply and demand changes impact on safe operation of the whole system, and refactoring support grid, fault processing, reactive power optimization and problem analysis of reducing the loss. Power flow calculation, however, often encounter such as computational convergence problem, previous solution can only be artificial adjustment has been conducted on the basis of power flow calculation program, on the one hand, because of many adjustable parameters in the system to adjust the efficiency is very low, on the other hand, the adjustment process relies heavily on the expert experience and also consume a large amount of human resources. Further, as the smart grid can be adjusted in the equipment supply and demand of rapid development, component features a variety of adjustable power supply and demand, makes in the smart grid, renewable energy and micro power grid tide control more conditions and uncertainties exist, therefore needs to be a trend of convergence algorithm of automatic adjustment control framework and precise in support of the smart grid control.

Combined with edge computing and artificial intelligence technologies, some studies have made preliminary attempts to apply edge intelligence to smart grid. In terms of edge computing, some works advocate that enabling smart grid with edge computing to overcome the defects of bandwidth and latency existing in cloud computing, which have a lot of application basis and design ideas. In terms of artificial intelligence, some studies focus on how to apply feature engineering or expert systems to manage power flow, however, some of them have shortcomings in the scalability and performance, and how to carry out power flow calculation more effectively with edge intelligence is still in its infancy.

In this paper, a power flow calculation adjustment framework based on deep reinforcement learning and edge computing is proposed for multiple power flow control problems in microgrids. Firstly, the typical business scenario requirements of power balance in microgrid are analyzed. Then, combined with the edge of computing and grid system design a micro grid current control framework, once again, based on the depth of intensive study defines the trend of the control of state space, space and set the rewards and punishments, and finally, with the IEEE 30 nodes by Pandapower system to put forward the framework, the simulation experiment verifies the framework described in this paper, the feasibility of the control issues in micro grid tide, for the intelligent power edge provides a certain reference.

The main contributions of the paper are summarized as follows: 1) We present a edge-computing-based comprehensive framework for smart grid management and control, which enables the data sensing, processing and controlling of smart grid to realize the functions of real time demand and local autonomy. 2) An learning-based algorithm is presented for power flow calculating considering the grid's requirement and current state. Deep reinforcement learning are also applied to improve the practicability and efficiency of the algorithm. 3) The simulation results demonstrate

the effectiveness of our method with successful dynamic power flow calculating and stable operation under various power conditions.

The structure of the paper is as follows. After the introduction in Section I, we summarize related works in Section II, and propose our framework and algorithm in Section III. Then, we present the configuration and evaluation results of simulation experiments in Section IV. Finally, in Section V the conclusions and further work are detailed.

Related works

Power flow adjustment is one of the most important problems in smart microgrid. This is due to the complexity of power supply and demand, since the balance of supply and demand in a power grid is determined by multiple adjustable power supply and demand devices, which mathematically is essentially the solution of nonlinear equations. Previous researchers have done a lot of research on power flow control, but how to apply edge intelligence to power flow calculation has a lot of problems.

Smart grid based on edge computing has triggered an unprecedented upsurge in recent years, and changed the model of power management in the past. Trajano designs a mobile edge computing based system architecture for a smart grid communication network that allows smart grid application to run at the mobile network edge, which provides a stable and low latency communication network between customers and providers to manage electrical power efficiently [1]. With a hardware-implemented architecture, Barik adopts fog computing in smart grid that offloads the cloud backend from multi-tasking, and improve efficacy in low power consumption, reduced storage requirement and overlay analysis capabilities [2]. Huang proposed an edge computing framework for real-time monitoring in smart grid with an efficient heuristic algorithm, which can increase the monitoring frame rate up to 10 times and reduce the detection delay up to 85% compared with cloud framework [3]. Similarly, awadi proposes a fog computing model to detect abnormal patterns in electricity consumption data in advance through the collaboration of distributed devices at the edge of smart grid, which the proposed model was tested reliable with low latency and cyber-resilient on real micro grid [4]. Chen proposes an edge computing system for IoT based smart grids, where electrical data is analyzed, processed and stored at the edge of the network [5]. Different strategies (privacy, data predication, preprocessing) are deployed on the system and simulation results shows that the proposed system supports connection and management of substantial terminals, real-time analysis and processing of massive data. The above work proposed a series of architectures and frameworks for applying edge computing to smart grid. However, they did not specifically consider the application of edge intelligence to microgrid. Hisham proposed a hybrid solution where edge computing is used for smart grid information processing, the cloud for power distributing, and a machine learning engine to establish the communication between different layers, which achieved higher power grid throughput and power utilization [6]. It's worth noting that this paper applies edge intelligence to distributed grids, but does not consider power flow calculations between microgrids.

Along with power consumers' increasing demand for flexibility and autonomy of power service, the microgrid framework has become a crucial component for the

modern smart grid. Shu proposed a real-time scheduling strategy based on deep reinforcement learning to economically dispatch microgrid energy storage considering operational uncertainty [7]. The agent is tested on the actual data, and the results show that the algorithm can achieve smaller operating costs in complex situations. Fang proposed a multiagent reinforcement learning approach for auction-based microgrid power scheduling market [8]. It reaches utility balance and supply-demand balance of the whole microgrid. Bi proposed a learning-based control microgrid scheduling strategy for economic energy dispatching [9]. The proposed solution does not require an explicit model that requires predictors to estimate stochastic variables with uncertainties. Simulation results in real data environment demonstrate the effectiveness of the proposed method. Etemad used a reinforcement learning based charging strategy for microgrid battery and renewable energy to improve electrical stability, power quality and the peak power load [10]. The results show that the model improves the use of renewable energy and battery and reduces the annual payments and peak consumption times. Liu proposed a collaborative reinforcement learning algorithm to solve the distributed economic scheduling problem of microgrid [11]. The algorithm reduces the coupling of nodes in the microgrid and improves the efficiency of distributed economic scheduling. The validity of the approach is verified by experiments with real data. Jayaraj employed Q learning algorithm, a variant of reinforcement learning, to carry out economic scheduling of a microgrid with photovoltaic cells and accumulators [12]. The experiments results show that the proposed method is effective and can reduce net transaction cost. The above work proposed a series of strategies and approaches for economic energy management. However, they did not specifically consider the changing of microgrid configuration. Dabbaghjamanesh proposes an approach for finding the optimal switching of reconfigurable microgrids based on a deep learning algorithm. The algorithm learns the network topology characteristics that vary with time and make real-time reconfiguration decisions [13]. Using the reconfiguration technique as a fast, reliable, and effective response can enhance the reliability and performance of the microgrid network.

For the problem itself, Ma discusses the application difficulties of deep learning in power flow calculation, proposes the network structure and training process of deep neural network, as well as the method to solve the over-fitting problem [14]. To solve the problem of manpower and time cost consumption caused by strict non-convergence of power flow in large-scale power grid calculation, Wang proposes an adjustment method of power flow convergence based on knowledge experience and deep reinforcement learning [15]. To quantify the impact of the correlation among multi-dimensional wind farms on the power system, Zhu proposes a probabilistic power flow calculation framework with learning-based distribution estimation approach [16]. Yang *et al.* propose a model-based deep learning approach to quickly solve the power flow equations, with the main application of speeding up probabilistic power flow calculations [17]. Compared with the pure data-driven deep learning method, the proposed method can comprehensively improve the approximate accuracy and training speed. Compared with the current situation that traditional machine learning algorithms are mostly used for state identification and evaluation, Su *et al.* propose a power system control algorithm embedded in deep confidence

network [18]. By combining NSGA-II algorithm and deep confidence network, the control optimization strategy can be solved quickly and stably. Some of the above work considers how to apply the deep learning method to the power flow calculation problem. However, the research on the application of edge intelligence to the problem of microgrid is still in the preliminary stage.

From the point of view of all researchers in the literature, there is hardly research literature has considered how to apply edge intelligence to the power flow calculation of microgrid. The existing methods have poor adaptability to the edge computing framework and are unable to deal with local autonomy, or lead to the failure of power flow calculation convergence, thus leading to system instability. Different from the above work, our research proposes a flow control framework based on edge computing and deep reinforcement learning, while considering resource efficiency and workload arrangement. Considering the complexity of the work flow, we focus on the situation that the system does not converge, design the algorithm framework based on deep reinforcement learning, and fully consider how to deal with the problem of unbalanced power environment.

The Framework of Power Flow Adjustment based on Edge Intelligence

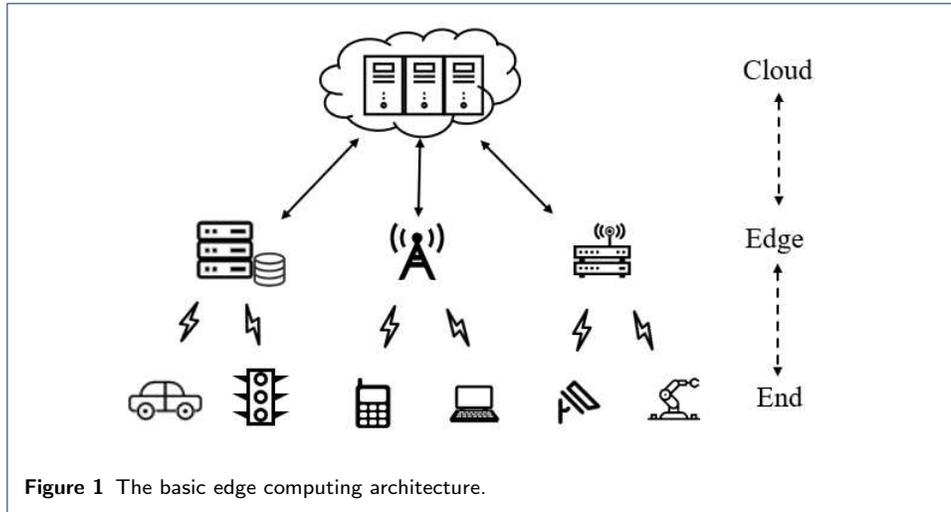
Edge Computing

Due to the rapid increase in the number of mobile devices, conventional centralized cloud computing is struggling to satisfy the QoS for many applications. With 5G network technology on the horizon, edge computing will become the key solution to solving this issue. It is mainly required by some delay-sensitive applications, such as virtual reality, which has stringent delay requirements. Thus, the edge computing paradigm by pushing the cloud resources and services to the edge, enables mobility support, location awareness, and low latency.

Generally speaking, the structure of edge computing can be divided into three levels: end device, edge server, and cloud. Figure 1 illustrates the basic architecture of edge computing. This hierarchy represents the computing capacity of edge computing elements and their characteristics. End devices (e.g., sensors, actuators) provide more interactive and better responsiveness for users. However, due to their limited capacity, resource requirements must be forwarded to the servers. Edge servers can support most of the traffic flows in networks as well as numerous resource requirements, such as real-time data processing, data caching, and computation offloading. Therefore, edge servers provide better performance for end users with a small increase in the latency. Cloud servers provide more powerful computing (e.g., big data processing) and more data storage with a transmission latency. The goal of this architecture is to execute the compute-intensive and delay-sensitive part of an application in the edge network, and some applications in the edge server communicate with the cloud for data synchronization.

The hierarchical architecture of edge computing encompasses the following attributes.

- 1) Proximity and low latency: Near to the end of the edge computing both in a physical and a logical sense supports more efficient communication and information distribution than the far-away centralized cloud.



2) Intelligence and control: The performance of a modern edge node is sufficient for the high rate transmission, large data storage and sophisticated computing programs for a set of local users.

3) Less concentration and privacy: Many edge computing servers could be private-owned cloudlets and these less concentration of information shall ease the concern of information leakage in cloud computing caused by the separation of ownership and management of data.

4) Heterogeneous and scalability: Edge computing that scales to a large number of sites, is a cheaper way to achieve scalability than fortifying the servers.

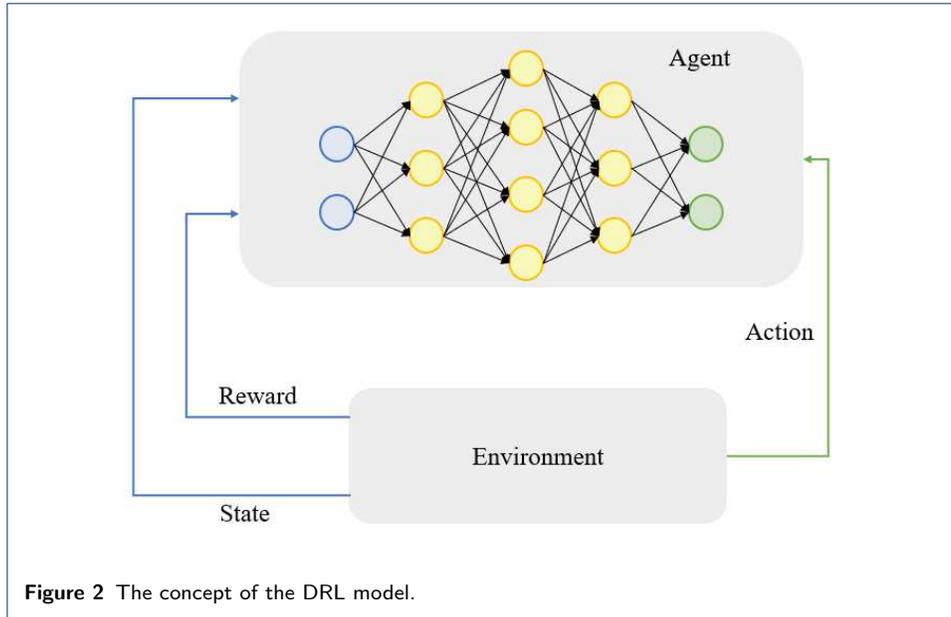
Deep Reinforcement Learning

An RL task is defined by $M = (S, A, T, r)$. At each time-step t , the agent receives a state $s_t \in S$ and selects an action $a_t \in A$ according to its policy $\pi : a_t = \pi(s_t)$. A distribution of state transition $T = p(s_{t+1}|s_t, a_t)$, which is a mapping from state-action pairs (s_t, a_t) to a probability distribution over the next states. After interacting with the environment, the agent reaches the next state s_{t+1} and receives a reward $r_t = r(s_t, a_t)$.

The expected discounted return at time t is given by $R_t = \sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'}$ where $\gamma \in [0, 1]$ is the discount factor, and the goal of the RL agent is to maximize its expected return. The action-value function, Q function, is defined as $Q^\pi(s, a) = \mathbb{E}[R_t | s_t = s, a_t = a, \pi]$, which represents the expected discounted return after observing the state s and taking the action a depending on the policy π . The optimal Q function Q^* satisfies the following Bellman equation:

$$Q^*(s, a) = \mathbb{E}_{s' \sim p(\cdot | s, a)} [r(s, a) + \gamma \max_{a' \in A} Q^*(s', a')].$$

DRL is composed of DNNs and RL. As illustrated in Figure 2, the goal of DRL is to create an intelligent agent that can perform efficient policies to maximize the rewards of long-term tasks with controllable actions. DQN algorithm is a model-free approach for RL using DNNs in environments with discrete action spaces, which optimizes neural networks to approximate the optimal Q function Q^* . In DQN, the



expected discounted future return of each possible action is predicted at timestep t and the RL agent take the action with the highest predicted return: $\pi_Q(s_t) = \arg \max_{a \in A} Q(s_t, a)$. During training the RL agent collects the tuples (s, a, r, s') from its experience and stores them in an experience replay memory, which is a key technique to improve training performance in the DQN algorithm. The purpose of the replay memory is to remove correlations between samples experienced by the agent. The neural network to approximate $Q^*(s, a)$ is trained using a mini-batch gradient descent approach and minimizes the following loss by using samples (s, a, r, s') from the replay memory: $L = \mathbb{E}_{s,a,r,s'} [(Q(s, a) - y)]^2$, where $y = r + \gamma \max_{a' \in A} Q(s', a')$. In DQN, the RL agent uses a separate target Q-network, which has the same architecture as the original Q-network but with frozen parameters. The purpose of the target network is to temporarily fix the Q value targets because non-stationary targets make a training process unstable and reduce performance. The parameters of the target Q-network θ^- are updated with that of the original Q-network θ every fixed number of iterations. For the use of the target Q-network, the loss function can be reformulated as follows:

$$L(\theta) = \mathbb{E}_{s,a,r,s'} \left\{ \underbrace{Q(s, a; \theta)}_{\text{prediction}} - \underbrace{\left[r + \gamma \max_{a' \in A} Q(s', a'; \theta^-) \right]}_{\text{target}} \right\}^2.$$

Advantage Actor Critic (A2C)

The Actor-Critic algorithm uses two neural networks to approximate the policy. One is a neural network that approximates policy, and an object that selects an action using this network is called an Actor. This neural network that approximates the policy is called a policy network. The other is a neural network that judges whether the action selected by the Actor is good or bad behavior. Using this network, an object that predicts the value of the action that the Actor selected is called the value

network. The value network approximates a Q function that directly represents the value of an action that an actor chooses in a specific state. Let the weights of the policy network at time t be θ_t , the state at time t be s , the selected action be a , the learning rate be α , and the policy that has parameter θ be π_θ . The update equation of parameter θ of the policy network is

$$\theta_{t+1} \approx \theta_t + \alpha[\nabla\theta \log\pi_\theta(a|s)Q_\pi(s, a)].$$

$Q_\pi(s, a)$ is the total value that can be obtained by continuing the action selection along the policy π after selection action a in the current state s . In the equation above, the Q function to which the value network approximates is not normalized. Therefore, if the value of Q predicted by Critic using the value network is too large, the parameter θ changes too much at a time. Conversely, if predicted value is too small, θ does not change much. Instead of the predicted Q value, the value obtained by subtracting the value of the previous state from the Q value is used, which is called advantage. This advantage implies an increment of value obtained by action a . If the value function at time-step t is $V(s_t) = \mathbb{E}[R_t|s_t = s]$, the advantage function is

$$\begin{aligned} A(s_t, a_t) &= Q(s_t, a_t) - V(s_t) = \mathbb{E}[R_t|s_t, a_t] - V(s_t) \\ &\approx r_t + \gamma V(s_{t+1}|s_t, a_t) - V(s_t) = \delta(s_t). \end{aligned}$$

The gradient of the actor is $\nabla\theta \log\pi_\theta(a|s)\delta(s_t)$, then

$$\theta_{t+1} \approx \theta_t + \alpha[\nabla\theta \log\pi_\theta(a|s)\delta(s_t)].$$

The loss function of updating value network is given as $\delta(s_t)^2$.

Knowledge and experience of power flow convergence

Analysis of grid power flow unsolved and generator active power output

In the actual power grid, the unreasonable arrangement of generator sets may result in excessive active power transmission, exceeding the transmission capacity of the network [19]. In response to this situation, adding reactive power compensator or changing the transformer ratio on the line can improve the transmission capacity of the network to a certain extent. But when faced with extreme unreasonable arrangements, these methods are difficult to achieve a better trend adjustment. Therefore, in order to ensure that the active power transmitted by the line in the power grid does not exceed the upper limit of its transmission capacity, the output of each generator in the generator set needs to be adjusted [20].

Analysis of grid power flow unsolved and line transmission limit

The transmission capacity of the transmission line reaching the limit is the main factor for the static stability of the system. Under general condition, the transmission power of the lines in the grid changes with the changes in the generator output and the active and reactive power of the load.

There are two situations when the active power of a transmission line reaches its transmission power limit: **(i)** With the continuous increase of the injected power,

the active power of the transmission line reaching the limit will not continue to increase (or increase very little), and the increase in injected power is transmitted through other transmission channels; **(ii)** The active power of the transmission line increases with the injected power, but the reactive power transmission of the line reaches the limit.

The line reaching its transmission power limit is a necessary condition for the power system to lose static stability. In this case, the system power flow has no solution and the calculation does not converge. Based on the above analysis, it is possible to add the corresponding reactive power compensation and adjust the method of power injection by finding the line that reaches the transmission limit in the system as knowledge and experience. In this way, the distribution of the power flow in the network can be changed, and the purpose of power flow non-convergence adjustment can be achieved.

Adjustment method of unsolved power flow

a). Adjustment of generator output

For small-scale distribution networks with power supply path compensation and direct power supply without boosting, adjusting generator output is a relatively economical power flow adjustment method. At this time, there is no need to add additional electrical equipment for adjustment, just change the generator terminal voltage to achieve good results. For power supply systems with long lines and multiple voltage levels, the adjustment of generators alone cannot meet the requirements of power flow calculation convergence.

b). Adjustment of transformer ratio

Changing the transformer ratio can increase or decrease the voltage of the secondary winding. There are several taps for selection on the high-voltage side winding of the double-winding transformer and the high-voltage side and medium-voltage side winding of the three-winding transformer. The one corresponding to the rated voltage is called the main connector.

c). Reactive power compensation

The generation of reactive power basically does not consume energy, but the transmission of reactive power along the power grid will cause active power loss and voltage loss. Reasonable configuration of reactive power compensation and changing the reactive power flow distribution of the network can reduce the active power loss and voltage loss in the network.

d). Line series capacitor

Changing line parameters and voltage regulation can be aimed at resistance and reactance, but it is not economical to reduce resistance by increasing the radius of the wire. Therefore, capacitors can be connected in series on the line to offset the reactance and reduce voltage loss .

Automatic Adjustment of Power Flow Calculation Convergence Based on DRL

In some MADRL algorithm, each agent requires observation information such as the opponent's policy from other agents in addition to its observation from environment. However, in microgrid, it is unrealistic to

Deep reinforcement learning process design

In view of the above statements, a deep reinforcement learning method based on knowledge and experience of power flow is proposed to automatically adjust the case that power flow does not converge, when the balance of active power and reactive power is considered simultaneously. For agents in deep reinforcement learning,

State

For the agent, the state means the variables observed from environment, which will affect the exploration efficiency of the agent. Therefore, in the selection of state variables, we mainly consider the output of each generator and the switching of the reactive power reactive power compensator on each bus. Therefore, for the data of m samples, a state space of $m(n + p)$ is constructed, where n is the total number of generators and p is the total number of buses.

Action

Action is the actual policy made by the agent in the process of exploration. It is the key to truly affect the convergence of real-time power flow. We consider the regulation of both active power and reactive power, that is, the output of each generator and the number of capacitor switches on each heavy-load bus. Therefore, for the data of m samples, an action space of $m(n + q)$ is constructed, where n is the total number of generators and q is the total number of heavy-load buses.

Reward

In order to make use of the knowledge related to power flow calculation and improve the efficiency of agents' exploration, we set up multiple reward mechanisms. First of all, if the power flow of a sample converges, the highest positive reward value R_1 can be obtained, and finally the negative reward value R_2 will be added if the power flow doesn't converge. Then, the upper limit of generator output is considered. If the active power output of the generator is greater than its maximum active power limit, the negative reward value R_3 is added. Similarly, if the reactive power output of the generator is greater than its maximum reactive power limit, the negative reward value R_4 is added. The line load is also a vital part when calculate power flow. If the line load rate exceeds its maximum line load rate, the agent will get a negative reward R_5 . Additionally, we also consider the voltage level on the bus. If the voltage on the bus is within its specified maximum or minimum voltage limit, then add the positive bonus value R_6 . Finally, the reward value R for each step of the agent is equal to the sum of the above six types of rewards $R_1, R_2, R_3, R_4, R_5, R_6$.

Actor and Critic Networks Design

The detailed DRL network and process of automatic adjustment policy is based on a deep policy gradient method in [21]. We adopt A2C as our deep reinforcement learning algorithm, which consist of two deep neural networks, namely the actor network and the critic network.

The actor network is used to explore the policy, and the critic network estimates the performance and provides the critic value, which helps the actor to learn the gradient of the policy.

The A2C is obtained by combining the value function approximation algorithm with the policy gradient descent method. To put it simply, it is composed of two networks, in which the Actor network is the one that actually executes the action in the environment, and it contains the strategy and its parameters, which is responsible for selecting the action. The critic network does not do any action; it is used to evaluate actions using value function approximations, and it evaluates the actor network by replacing the real Q-values with approximate Q-values. According to critic, the actor adjusts the parameters of its network to make an update in a better direction.

Numerical Results

Simulation Setting

In the experimental part, based on the Python3.7 environment, we use and modify Pandapower [22], an open source third-party library, as a power flow calculation and analysis tool, which can not only calculate the convergence of power flow, but also provide intermediate results of power flow calculation process. In the power flow calculation algorithm, we choose the Newton-Raphson method with Iwamoto multiplier. In the environment construction part of deep reinforcement learning, we used the environment provided by OpenAI Lab to build the interface Gym [23]. In the implementation part of deep learning algorithm, we adopted the Stable Baselines [24] interface provided by OpenAI Lab. In terms of parameters, the upper limit of iteration number of flow convergence is set as 10, the total number of samples is 208, and the number of test episodes is 10.

Data Preprocessing

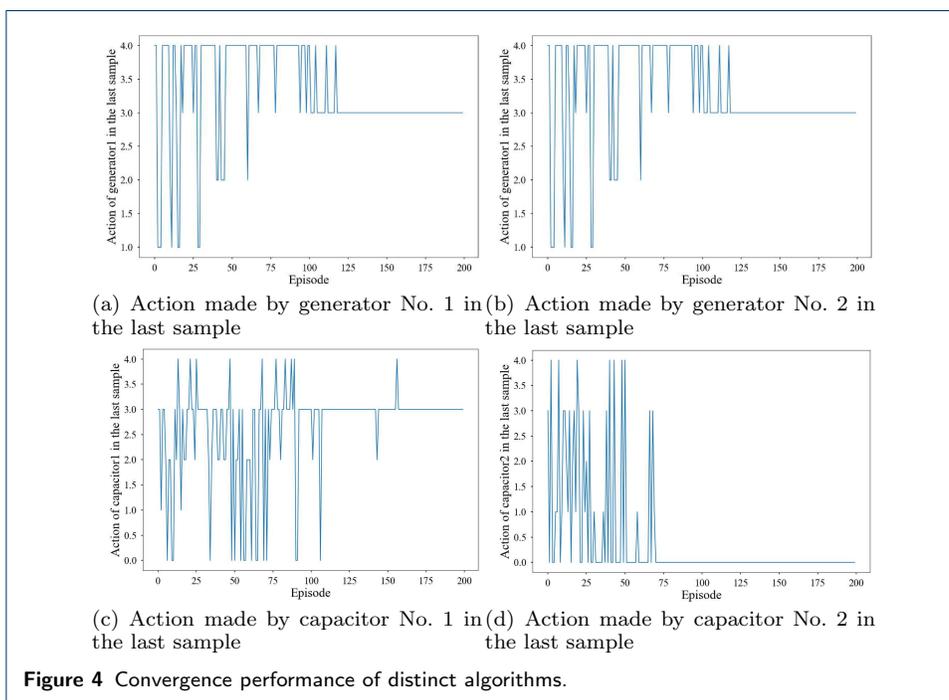
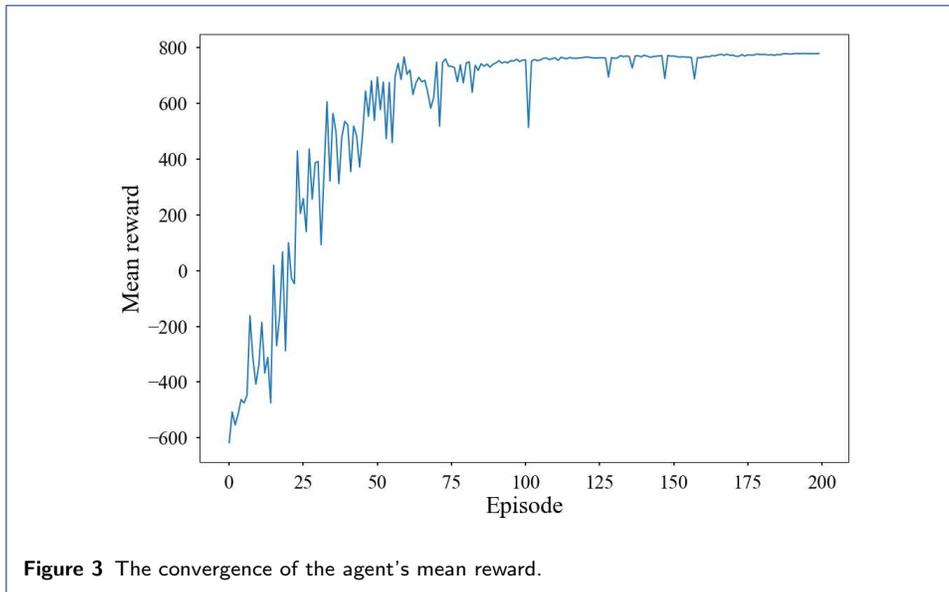
We choose IEEE 30 node system as the object of our experiment. The system represents a 345kV power grid in New England, USA, consisting of 10 generators, 12 double-winding transformers and 34 lines, with a base power of 100MVA. Based on the original convergent data of the system, 1000 sets of samples were regenerated by randomly adjusting the size of generator and load within the range of 0-2 times. The Newton-Raphson method with Iwamoto multiplier is used to calculate the power flow of 1000 samples one by one, and 208 non-convergent sample data are finally obtained as the data of non-convergent power flow adjustment.

Simulation Results

As can be seen from the average reward convergence Fig. 3, the reward value of agents' exploration in deep reinforcement learning increases steeply with the increase of the episode. It converges to a relatively stable value at around Episode 160. Thus, the setting of reward value in deep reinforcement learning can help continuously optimize the parameters in the power grid.

The four graphs in Fig. 4 are from the last sample adjusted to convergence, and the action convergence graphs of two generators and two reactive compensators are selected. As can be seen from the figure, the actions of some devices, such as generator No. 2, converge to certain values very early in the episode, while the actions of others, such as capacitor No. 1, take longer to converge.

For example, we compare our method with some baseline methods, such as random exploration and A2C without knowledge experience. For 100 samples, compare



the number of samples where each algorithm converges within 10 episodes. The experimental results show that knowledge experience plays a significant role in guiding the exploration process of deep reinforcement learning.

Conclusion

The conclusion goes here. In this article, we proposed an edge computing assisted comprehensive framework for smart grid management and control. Consequently, it assist smart grid to realize real-time demand response and local autonomy in data sensing, processing and controlling. Especially, a power flow calculation adjustment

Table 1 comparison of different methods

Adopted method	Convergent sample number per 100
A2C with knowledge and experience	93
random exploring	60
Ordinary A2C	68

algorithm based on deep reinforcement learning considering the grid knowledge and requirement in microgrids, which improve the efficiency and flexibility compared with traditional adjustment method. Finally, we adopt IEEE 30 nodes with Pandapower under various grid conditions to verify the effectiveness of our proposed algorithm.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

1) We present a edge-computing-based comprehensive framework for smart grid management and control, which enables the data sensing, processing and controlling of smart grid to realize the functions of real time demand and local autonomy. 2) An learning-based algorithm is presented for power flow calculating considering the grid's requirement and current state. Deep reinforcement learning are also applied to improve the practicability and efficiency of the algorithm. 3) The simulation results demonstrate the effectiveness of our method with successful dynamic power flow calculating and stable operation under various power conditions.

Acknowledgements

This work was supported in partly by the National Natural Science Foundation of China under Grant U2066213, and in partly by the project Transplanting Technology Service of Embedded Artificial Intelligence Algorithm Platform under Grant EPRI4130-201864. (Corresponding author: Chao Qiu.)

Author details

¹China Electric Power Research Institute, Haidian District, 100192 Beijing, China. ²College of Intelligence and Computing, Tianjin University, Jinnan District, 300350 Tianjin, China.

References

- Trajano, A.F.R., de Sousa, A.A.M., Rodrigues, E.B., de Souza, J.N., de Castro Callado, A., Coutinho, E.F.: Leveraging mobile edge computing on smart grids using lte cellular networks. In: 2019 IEEE Symposium on Computers and Communications (ISCC), pp. 1–7 (2019). doi:10.1109/ISCC47284.2019.8969784
- Barik, R.K., Gudey, S.K., Reddy, G.G., Pant, M., Dubey, H., Mankodiya, K., Kumar, V.: Foggrid: Leveraging fog computing for enhanced smart grid network. In: 2017 14th IEEE India Council International Conference (INDICON), pp. 1–6 (2017). doi:10.1109/INDICON.2017.8487997
- Huang, Y., Lu, Y., Wang, F., Fan, X., Liu, J., Leung, V.C.M.: An edge computing framework for real-time monitoring in smart grid. In: 2018 IEEE International Conference on Industrial Internet (ICII), pp. 99–108 (2018). doi:10.1109/ICII.2018.00019
- El-Awadi, R., Fernández-Vilas, A., Díaz Redondo, R.P.: Fog computing solution for distributed anomaly detection in smart grids. In: 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 348–353 (2019). doi:10.1109/WiMOB.2019.8923222
- Chen, S., Wen, H., Wu, J., Lei, W., Hou, W., Liu, W., Xu, A., Jiang, Y.: Internet of things based smart grids supported by intelligent edge computing. *IEEE Access* **7**, 74089–74102 (2019). doi:10.1109/ACCESS.2019.2920488
- Albataineh, H., Nijim, M., Bollampall, D.: The design of a novel smart home control system using smart grid based on edge and cloud computing. In: 2020 IEEE 8th International Conference on Smart Energy Grid Engineering (SEGE), pp. 88–91 (2020). doi:10.1109/SEGE49949.2020.9181961
- Shu, Y., Bi, W., Dong, W., Yang, Q.: Dueling double q-learning based real-time energy dispatch in grid-connected microgrids. In: 2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), pp. 42–45 (2020). doi:10.1109/DCABES50732.2020.00020
- Fang, X., Wang, J., Yin, C., Han, Y., Zhao, Q.: Multiagent reinforcement learning with learning automata for microgrid energy management and decision optimization. In: 2020 Chinese Control And Decision Conference (CCDC), pp. 779–784 (2020). doi:10.1109/CCDC49329.2020.9164742
- Bi, W., Shu, Y., Dong, W., Yang, Q.: Real-time energy management of microgrid using reinforcement learning. In: 2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), pp. 38–41 (2020). doi:10.1109/DCABES50732.2020.00019
- Etamad, S., Mozayani, N.: Using reinforcement learning to make smart energy storage sources in microgrid. In: 2015 30th International Power System Conference (PSC), pp. 345–350 (2015). doi:10.1109/IPSC.2015.7827771
- Liu, W., Zhuang, P., Liang, H., Peng, J., Huang, Z.: Distributed economic dispatch in microgrids based on cooperative reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems* **29**(6), 2192–2203 (2018). doi:10.1109/TNNLS.2018.2801880

12. Jayaraj, S., T P, I.A., Kunju K, B.: Application of reinforcement learning algorithm for scheduling of microgrid. In: 2019 Global Conference for Advancement in Technology (GCAT), pp. 1–5 (2019). doi:10.1109/GCAT47503.2019.8978453
13. Dabbaghjamanesh, M., Zhang, J.: Deep learning-based real-time switching of reconfigurable microgrids. In: 2020 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT), pp. 1–5 (2020). doi:10.1109/ISGT45199.2020.9087729
14. Ma, D., Shen, C., Chen, Y., Li, D.: Feasibility study on the convergence criterion of power flow calculation based on deep learning. *Southern Power System Technology (Chinese)* **014**(002), 46–54 (2020)
15. Wang, T., Tang, Y., Guo, Q., Huang, Y., Chen, X., Huang, H.: Automatic adjustment method of power flow calculation convergence for large-scale power grid based on knowledge experience and deep reinforcement learning. *Proceedings of the Chinese Society for Electrical Engineering (Chinese)* **40**(8), 2396–2405 (2020)
16. Zhu, X., Liu, C., Su, C., Liu, J.: Learning-based probabilistic power flow calculation considering the correlation among multiple wind farms. *IEEE Access* **8**(99). doi:10.1109/ACCESS.2020.3011511
17. Yang, Y., Yang, Z., Yu, J., Zhang, B., Zhang, Y., Yu, H.: Fast calculation of probabilistic power flow: A model-based deep learning approach. *IEEE Transactions on Smart Grid* **11**(3), 2235–2244 (2020). doi:10.1109/TSG.2019.2950115
18. Su, T., Liu, Y., Shen, X., Liu, T., Qiu, G., Liu, J.: Deep learning-driven evolutionary algorithm for preventive control of power system transient stability. *Proceedings of the Chinese Society for Electrical Engineering (Chinese)* **40**(12), 103–114 (2020)
19. Y.Wang, y. publisher=Science press: *Power System Voltage Stability Analysis*
20. Song.Zhang, Y.Wang, D.Zhang, Y.Huang: *Power grid active power flow adjustment method and device based on improved DC power flow algorithm* (2020)
21. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015)
22. Thurner, L., Scheidler, A., Schäfer, F., Menke, J., Dollichon, J., Meier, F., Meinecke, S., Braun, M.: pandapower — an open-source python tool for convenient modeling, analysis, and optimization of electric power systems. *IEEE Transactions on Power Systems* **33**(6), 6510–6521 (2018). doi:10.1109/TPWRS.2018.2829021
23. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: OpenAI Gym (2016). *arXiv:1606.01540*
24. Hill, A., Raffin, A., Ernestus, M., Gleave, A., Kanervisto, A., Traore, R., Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y.: *Stable Baselines*. GitHub (2018)

Figures

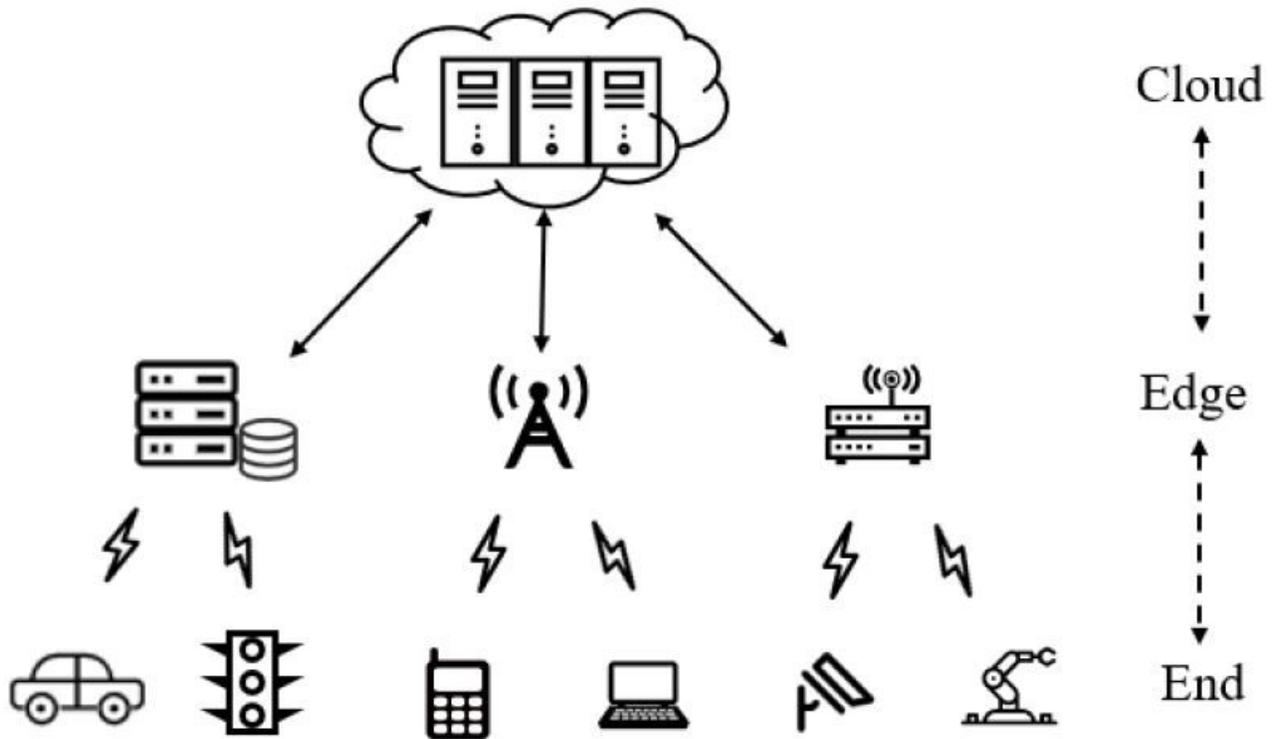


Figure 1

The basic edge computing architecture.

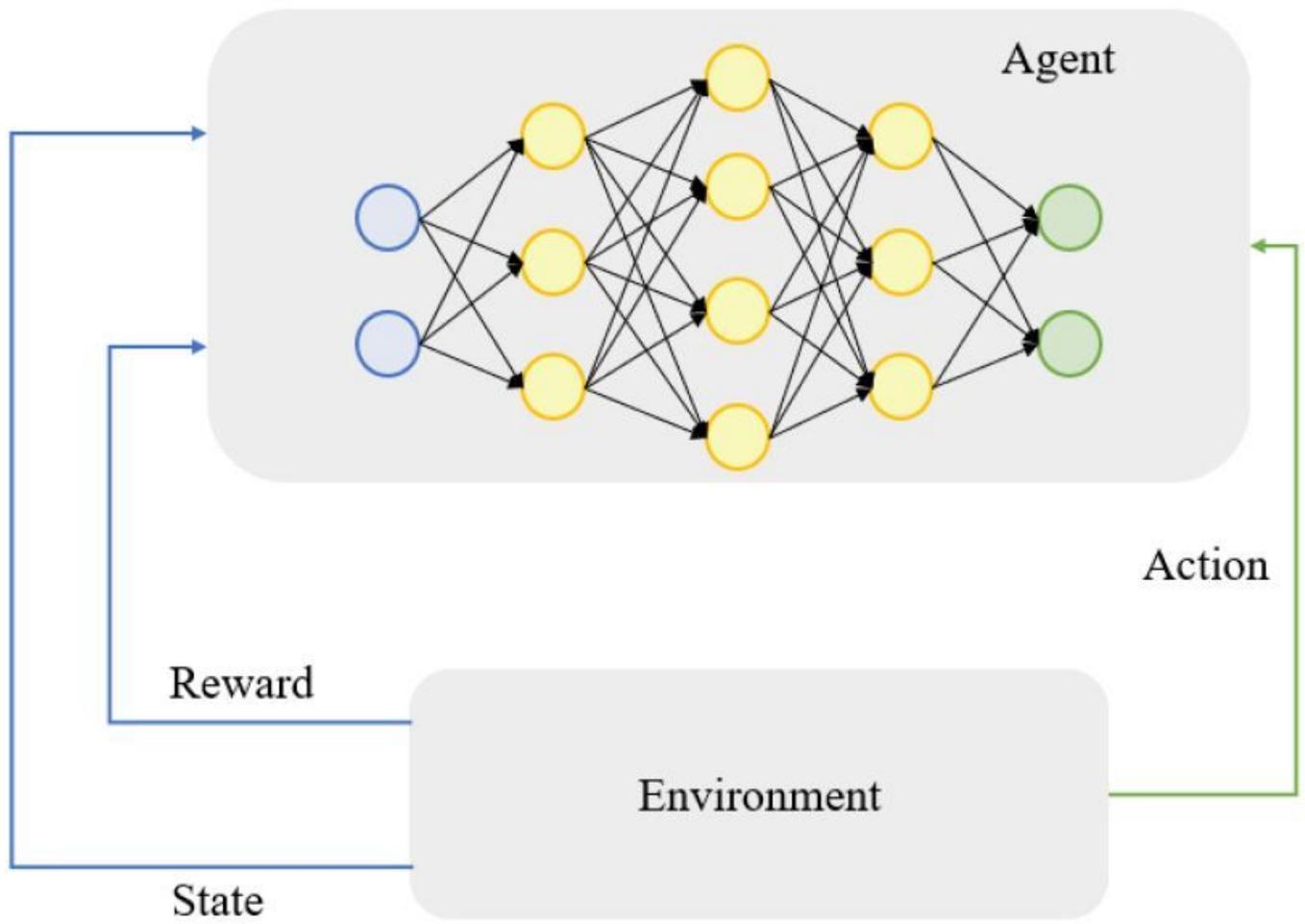


Figure 2

The concept of the DRL model.

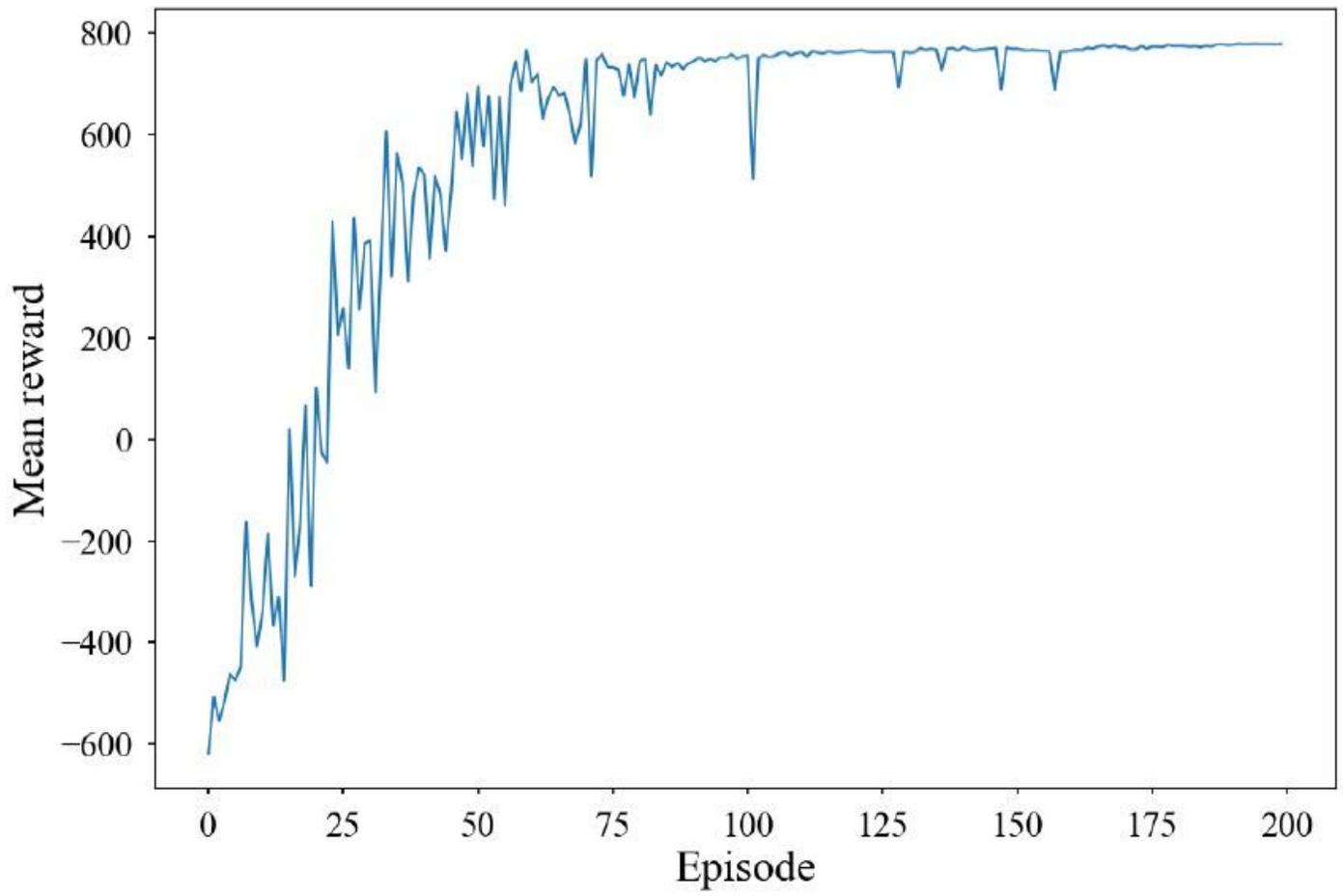
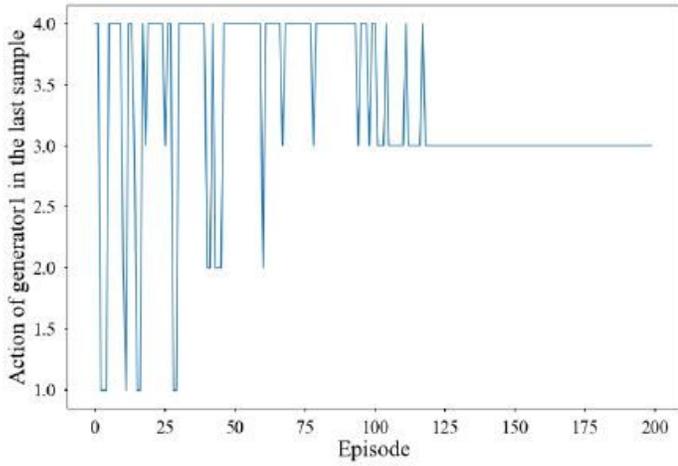
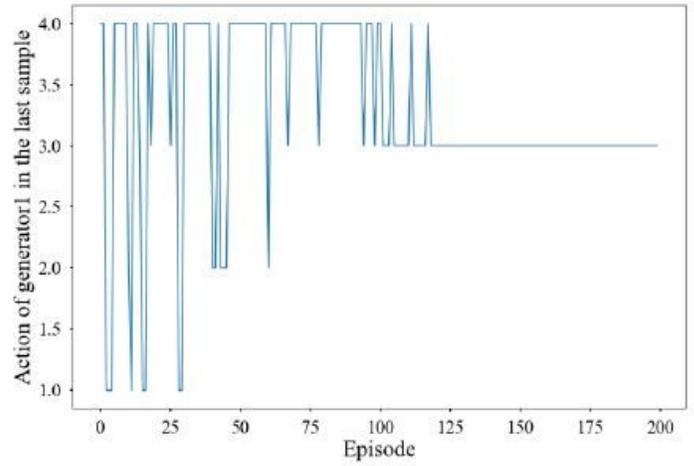


Figure 3

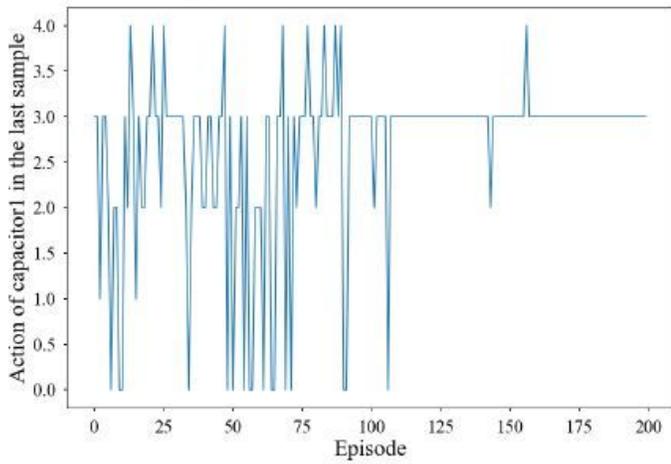
The convergence of the agent's mean reward.



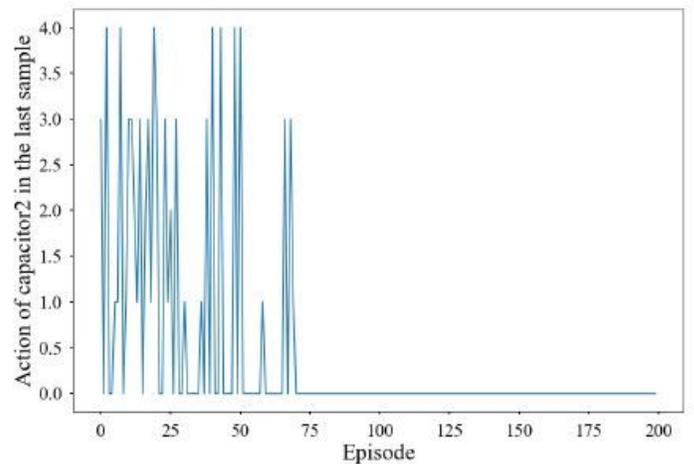
(a) Action made by generator No. 1 in the last sample



(b) Action made by generator No. 2 in the last sample



(c) Action made by capacitor No. 1 in the last sample



(d) Action made by capacitor No. 2 in the last sample

Figure 4

Convergence performance of distinct algorithms.