Active Power Correction Strategies Based on Deep Reinforcement Learning—Part II: A Distributed Solution for Adaptability

Siyuan Chen, *Student Member*, *IEEE*, Jiajun Duan, *Member*, *IEEE*, Yuyang Bai, Jun Zhang, *Senior Member*, *IEEE*, Di Shi, *Senior Member*, *IEEE*, Zhiwei Wang, *Senior Member*, *IEEE*, Xuzhu Dong, *Senior Member*, *IEEE*, and Yuanzhang Sun, *Senior Member*, *IEEE*

***Abstract*—This article is the second part of Active Power Correction Strategies Based on Deep Reinforcement Learning. In Part II, we consider the renewable energy scenarios plugged into the large-scale power grid and provide an adaptive algorithmic implementation to maintain power grid stability. Based on the robustness method in Part I, a distributed deep reinforcement learning method is proposed to overcome the influence of the increasing renewable energy penetration. A multi-agent system is implemented in multiple control areas of the power system, which conducts a fully cooperative stochastic game. Based on the Monte Carlo tree search mentioned in Part I, we select practical actions in each sub-control area to search the Nash equilibrium of the game. Based on the QMIX method, a structure of offline centralized training and online distributed execution is proposed to employ better practical actions in the active power correction control. Our proposed method is evaluated in the modified global competition scenario cases of "2020 Learning to Run a Power Network - Neurips Track 2".**

***Index Terms*—Active power correction strategies, distributed deep reinforcement learning, Nash equilibrium, renewable energies, stochastic game.**

NOMENCLATURE

A. Functions

$C_{\text{loss}}(\cdot)$	Function of power loss cost.
$C_{\text{redis}}(\cdot)$	Function of generation redispatch cost.
$C_{\text{bkt}}(\cdot)$	Function of blackout cost.
$R(\cdot)$	Function of cumulative reward.
$r_{i,t}(\cdot)$	Immediate reward function of agent i at time t.
$g_t(\cdot)$	Performance function of agent i at time t.
$V_i(\cdot)$	Function of state value.
$p(\cdot)$	Function of state transition probability.
$Q_i(\cdot)$	Q function of agent i.
$Q_{\text{tot}}(\cdot)$	Joint Q function of multi-agents.
$L(\cdot)$	Loss function of Q network.

B. Variables

T_{e}	Duration of the blackout.
r_l	Resistance of line l.
y_l	Active power of line l.
$P_{G_{i,t}}$	Active power output of generator i at time slot t.
$P_{j,t}$	Energy loss in the presence of blackout.
H_l	The binary variable of BBS action.
$\pi_{i,k}$	Strategy of agent i.
$\theta, \hat{\theta}$	Parameters of Q network.
τ_i	Action-observation history of agent i.
$\boldsymbol{a}_k$	Joint action of multi-agents.
$\boldsymbol{\pi}_k$	Joint strategy of multi-agents.
$\boldsymbol{\pi}_k^*$	Optimal joint strategy of multi-agents.
$\boldsymbol{\tau}$	Joint action-observation history of multi-agents.
$\boldsymbol{\theta}$	Network parameters of multi-agents.

C. Sets

$\boldsymbol{N}_l$	Set of lines.
$\boldsymbol{N}_{\text{Gen}}$	Set of generators.
$\boldsymbol{N}_{\text{Load}}$	Set of loads.
$\boldsymbol{A}_{\text{ag}}$	Set of multi-agents.
$\boldsymbol{S}$	State space of multi-agents.
$\boldsymbol{O}_i$	Partial observation space of agent i.
$\boldsymbol{A}_i$	Actions space of agent i.
$\boldsymbol{T}$	Operation duration of power system.

D. Constants

$N_l^{\max}$	Maximum number of allowable lines switching.
ρ_l	Usage of lines l capacity.
φ	Penalty coefficient of heavy load.
ψ	Penalty coefficient of overload.
κ	Penalty term of blackout in reward function.
ζ	Coefficient of soft update.
ε	Greedy parameter.

Manuscript received December 30, 2020; revised Feberuary 20, 2021; accepted April 14, 2021. Date of online publication September 10, 2021; date of current version November 11, 2021.This work was supported by the National Key R&D Program of China under Grant 2018AAA0101502.

S. Y. Chen, Y. Y. Bai, J. Zhang (corresponding author, e-mail: jun.zhang.ee@whu.edu.cn; ORCID: 0000-0001-6908-2671), X. Z. Dong, and Y. Z. Sun are with the School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China.

J. J. Duan, D. Shi, and Z. W. Wang, are with GEIRI North America, San Jose, CA 95134, USA.

DOI: 10.17775/CSEEJPES.2020.07070

I. INTRODUCTION

LARGE-SCALE plug-in distributed renewable energy resources (RESs) are gradually becoming a significant feature of a power grid. The randomness and volatility of

generation from RESs may overload the capacity of transmission lines when the power system has on-peak demand or scheduled maintenance. Considering that the uncertainty of the power system increases with the penetration rate of RESs, active power correction control (APCC) is a primary method to maintain the stability of the power system's active power. For APCC strategies, a large amount of research focuses on generation redispatch, load shedding, or demand response. Several studies have exploited a less costly method with great potential, namely grid topology reconfiguration. In [1], simulations were conducted to analyze more than 1.5 million kinds of faults based on the actual cases of the American power grid, and results demonstrated that topology change, e.g., bus bar switching, transmission switching, etc., can effectively eliminate or alleviate the transmission line overload in most fault scenarios.

The expanding scale of the power grid and the widespread application of power electronic devices have continuously increased the degree of dimensionality and non-linearity of the power grids. Power grids are increasingly difficult to model accurately using traditional mathematical and physical mechanisms; thus, they provide an opportunity for applications involving artificial intelligence technology in resolving dispatching and control problems of the power system. Deep reinforcement learning (DRL) combines reinforcement learning (RL) and deep learning technologies, which learn directly from the agents' interactions with an environment. In [2], a Q-learning-based method is proposed for optimal reactive voltage control, which solves the convergence problem of traditional reactive power optimization for non-linear integer programming models. In [3] and [4], a DRL method is applied to the scenarios of voltage instability, in which low-voltage load shedding strategies and shunt capacitor switching schemes are constructed. The effectiveness of the proposed method is verified through several unknown fault scenarios. In [5], a voltage control strategy based on Deep Q-Network (DQN) and Deep Deterministic Policy Gradient (DDPG) algorithms is applied for automatic voltage control (AVC). It ensures the voltage of each bus stays within the standard range, which is based on the information collected by the SCADA or PMUs. At present, DRL has gained extensive attention and remarkable achievements in the fields of games [6], medical treatment [7], autonomous driving [8], unmanned aerial vehicles [9], smart grid [10], [11], and other fields. However, it is difficult for DRL to perform well in high-dimensional action space, i.e., those which generally involve more than 10^4 kinds of actions. "The curse of dimensionality, " caused by the complexity of a power system, has become one of the biggest obstacles to practical applications of DRL in this field.

As an extension technology of DRL, distributed deep reinforcement learning (DDRL) has been applied in multi-agent systems (MAS), and this has effectively solved the problem of high-dimensional action space faced by DRL [12]. DDRL deploys multiple agents on buses of the large-scale power grid and divides the power grid into multiple sub-control areas. Due to the interaction and collaboration between multiple agents, the dimension of each agent's action space can be reduced to an acceptable level. In [13], a multi-agent reinforcement learning framework based on semantic information developed by Didi Chuxing, is presented, which can support both DQN and advantage actor-critic (A2C) algorithms to solve the problem of large-scale vehicle scheduling and management. In [14], by combining independent Q-learning (IQL) with DQN, a DDRL framework is proposed in which each agent has an independent Q network. However, since the IQL algorithm does not conduct interactions among multiple agents, the environment for each agent is unknown and non-static. This violates the principle of the Markov decision process (MDP) and cannot be proven to achieve the convergence of the algorithm. In [15], considering the dynamic instability of the environment caused by the IQL algorithm, a value-decomposition network (VDN) algorithm is proposed to obtain the joint action-value function by summing the Q-value of each agent. Numerical results demonstrate that the VDN algorithm can improve the convergence of the DDRL. In [16], based on VDN, a QMIX algorithm is proposed by constructing a mixing network to integrate local value functions. Global information is added to improve the network's ability to fit Q values in the training process. However, DDRL has not been studied and applied in the field of power systems.

In this paper, we propose a DDRL framework for joint control strategies of the APCC problem in large-scale power systems. Specifically, an APCC model with topology reconfiguration actions is established to formulate the fundamental problem. The fully cooperative stochastic game is then utilized to model the interactions between active power controllers (APC). A model-free model is adopted to search for the Nash equilibrium (NE) for the game. Considering that the control issue of the APCC problem is discrete, a QMIX method is adopted, which is modified from [16]. Based on the characteristic of the QMIX method, a structure that comprises centralized online training and offline distributed execution is proposed to satisfy the practical application requirements of large-scale power systems. Our method is verified in an open-source platform with relevant scenarios and cases.

The rest of this paper is organized as follows: The formulation of the APCC problem and the stochastic game is described in Section II. The method based on DDRL used to search for the NE is illustrated in Section III. Case studies in Section IV verify the performance of the proposed method. The conclusions and suggestions for future work are given in Section V.

II. Problem Formulation

A. APCC Model

Active power correction control (APCC) of power systems is a non-linear, mixed, integer programming problem usually solved by sensitivity and optimization programming methods. Generally, the APCC is achieved by generation redispatch and load shedding with limited effects on power flow control. Bus bar switching (BBS), which can switch the elements from a bus bar to another, is an effective means of correcting power flow quickly and effectively [17]. Considering that the influence of RESs requires rapid response to prevent branch power flow from going off-limit, this paper adopts the

optimization programming method of BBS. The mathematic model of the correction control is as follows:

$$\min_{y_l, P_{G_{i,t}}} \quad \sum_{t=1}^{t_{\text{end}}} (C_{\text{loss}}(t) + C_{\text{redis}}(t)) + \sum_{t=t_{\text{end}}}^{T_{\text{e}}} C_{\text{bkt}}(t) \tag{1}$$

$$C_{\text{loss}}(t) = \sum_{l \in \boldsymbol{N}_l} r_l y_l^2(t) \tag{2}$$

$$C_{\text{redis}}(t) = \alpha \sum_{i \in \boldsymbol{N}_{\text{Gen}}} |P_{G_{i,t-1}} - P_{G_{i,t}}| \tag{3}$$

$$C_{\text{bkt}}(t) = \sum_{j \in \boldsymbol{N}_{\text{Load}}} \beta P_{j,t} \tag{4}$$

where $C_{\text{loss}}(t)$, $C_{\text{redis}}(t)$, and $C_{\text{bkt}}(t)$ are power loss cost, generation redispatch cost, and blackout cost, respectively; t_{end} is the time when the power system blacks out; T_{e} is the duration of the blackout; $\boldsymbol{N}_l$, $\boldsymbol{N}_{\text{Gen}}$, and $\boldsymbol{N}_{\text{Load}}$ are set of lines, generators, and loads, respectively; r_l and y_l are the resistance and active power of line l, respectively; $P_{G_{i,t}}$ is the active power output of generator i at time slot t; $P_{j,t}$ is the energy loss in the presence of blackout.

In this paper, we focus on applying the topology actions of BBS, which is less costly than generation redispatch. A binary variable H_l is used to denote BBS action, which is 1 when line l is switched. Considering the stability of the power system, the BBS action in a time should be restricted, which is

$$\sum_{l \in N_l} (1 - H_l) \leq N_l^{\max} \tag{5}$$

where $N_l^{\max}$ is the maximum number of lines allowable for switching.

B. Stochastic Game and Nash Equilibrium

Limited by the action space dimension and the observation space size of a large-scale power system, a single agent cannot handle all the functionalities and management schemes in practical applications. A multi-agent system is essential to deal with "the curse of dimensionality," to implement a cooperative control and management scheme in large-scale power grids. In the framework of a DDRL, each agent follows the basic learning paradigm of reinforcement learning, which needs to consider both its exploration and the impact of other agents' strategies on the environment. The interaction among agents can be captured in the form of a cooperative stochastic game. The main components of the game include:

- Agent: APCs in the set $\boldsymbol{A}_{\text{ag}}$.
- State: the states s_t of the power system include active power outputs of generators, usage of lines capacity, electrical quantities, etc. $s_t \in \boldsymbol{S}$, where $\boldsymbol{S}$ is the state space.
- Observation: partial observation $o_{i,t}$ based on the functionality of agent i. $o_{i,t} \in \boldsymbol{O}_i$, where $\boldsymbol{O}_i$ is the partial observation space of agent i.
- Action: BBS actions $a_{i,t}$ of each agent or do nothing. $a_{i,t} \in \boldsymbol{A}_i$, where $\boldsymbol{A}_i$ is the action space of agent i.
- Reward: immediate reward $r_{i,t}$.

At the beginning of the time slot $t \in \boldsymbol{T}$, where $\boldsymbol{T}$ is the operation duration, RESs are connected randomly to the power grid, which may cause overloads of transmission lines. If the usage ratios of lines are controlled within a specific range, the power losses are reduced, and the power system is expected to survive longer. Hence, based on (1)–(4), the performance at time t can be formulated as follows [18]:

$$g_t = \sum_{l \in \boldsymbol{N}_l} (\max(0, 1 - \rho_l^2) - \varphi \cdot \max(0, \rho_l - 0.9) - \psi \cdot \max(0, \rho_l - 1)) \tag{6}$$

where ρ_l is the usage of lines l capacity; and, φ and ψ are the penalty coefficient of heavy load and overload, respectively.

Each APC obtains a cumulative reward when the power system maintains its normal operation, and get much larger negative rewards if a power system blackout occurs. Thus, the APCs perform a cooperative stochastic game to achieve a Nash equilibrium based on their observations. The immediate reward and cumulative reward of APC i at time k, can be formulated as

$$r_{i,t} = \begin{cases} \kappa & \text{if blackout} \\ \sum_{i=0}^{t} g_t & \text{otherwise} \end{cases} \tag{7}$$

$$R(s_k) = \sum_{t=k}^{\boldsymbol{T}} \gamma_i^{t-k} r_{i,t} \tag{8}$$

where κ is a negative constant; γ_i is the discount factor of APC I; and s_k is the states at time k.

The APCs aim to maintain the normal operation of the power system, i.e., the predicted reward. We can calculate the cumulative reward of all states unless the game is over. Thus, a value function is introduced to evaluate the potential future reward of states, which is:

$$\begin{aligned} V_i(s_k) &= \mathrm{E}[R(s_k)|S_t = s_k] \\ &= \mathrm{E}[R(s_{k+1}) + \gamma v_i(s_k)|S_t = s_k, A_t = \boldsymbol{a}_k] \\ &= \sum_{s_{k+1}, r_{i,k}} p(s_{k+1}, r_{i,k}|s_k, \boldsymbol{a}_k)[r_{i,k} + \gamma V_i(s_{k+1})] \end{aligned} \tag{9}$$

where $\boldsymbol{a}_k = [a_{1,k}, \cdots, a_{i,k}, \cdots, a_{N,k}]^{\mathrm{T}}$ is the joint action; and $p(s_{k+1}, r_{i,k}|s_k, \boldsymbol{a}_k)$ is state transition probability, that is, $p : s_k \times \boldsymbol{a}_k \times s_{k+1} \to [0, 1]$. (9) demonstrates that the stochastic game has Markov properties. $\pi_{i,k} : s_k \to \boldsymbol{a}_{i,k}$ denotes the strategy of APC i, and the joint strategy of the APCs can be described as $\boldsymbol{\pi}_k = [\pi_{1,k}, \cdots, \pi_{i,k}, \cdots, \pi_{N,k}]^{\mathrm{T}}$. The value function is related to the APCs' strategies, which can be denoted by $V_i(s_k, \boldsymbol{\pi}_k)$.

The solution of the stochastic game is NE, which is a state of the game where no agent can benefit by unilaterally changing strategies. Assuming that the NE solution of the APCC problem is denoted by $\boldsymbol{\pi}_k^* = [\pi_{1,k}^*, \cdots, \pi_{i,k}^*, \cdots, \pi_{N,k}^*]^{\mathrm{T}}$, the optimality of the NE solution can be described as:

$$V_i(s_k, \boldsymbol{\pi}_k^*) \geq V_i(s_k, \boldsymbol{\pi}_{k,-i}^*), \ \forall i \in N \tag{10}$$

where $\boldsymbol{\pi}_{k,-i}^* = [\pi_{1,k}^*, \cdots, \pi_{i,k}, \cdots, \pi_{N,k}^*]^{\mathrm{T}}$ are the optimal strategies of APCs excluding the APC i.

In the stochastic game of the APCC, we solve the optimization problem of the Q function to obtain the NE solution based

on the Bellman equation, which is:

$$\max_{\boldsymbol{\pi}_k} Q_i(s_k, \boldsymbol{a}_k) = \sum_{\boldsymbol{s}_{k+1} \in \boldsymbol{S}} p(s_{k+1}, r_{i,k} | s_k, \boldsymbol{a}_k) [r_{i,k} + \gamma V_i(s_{k+1})] \tag{11}$$

Therefore, we need to search for the maximum Q value to obtain the NE solution $\boldsymbol{\pi}_k^*$. To address this issue, some model-based methods in previous literature attempted to solve for the actions of APCs; however, the performance of these methods depends upon the accuracy of the models. Hence, a model-free method, i.e., DQN, is adopted in this paper to search for the NE solution.

III. Proposed DDRL Framework

A. Deep Recurrent Q Netowrk

In the APCC stochastic game, the exploration of each APC is a partial observation Markov decision process (POMDP). In this paper, the Deep Recurrent Q-Network (DRQN) algorithm is proposed to solve the problem with partial observation. Based on the basic structure of the DQN, DRQN replaces the first post-convolutional fully-connected layer with a recurrent long short-term memory (LSTM) network [19]. In the training process, the convolutional layer and LSTM layer are updated together. The Q value obtained by partial observation o_t can be much closer to the real Q value, which is $Q(o_t, a_t; \theta) \rightarrow Q(s_t, a_t; \theta)$.

At time-step t, after obtaining a state s_t from the environment, the agent estimates the Q-value through a fully connected neural network and chooses actions corresponding to the maximum Q-value. DRQN agents get a reward r_t and the state s_{t+1} at the next time step. Then, the experience $e(s_t, a_t, r_t, s_{t+1})$ is stored in a dataset named "replay buffer," which helps the DRQN eliminate the relationship between independent identically distributed training datasets.

Considering that the Q-learning algorithm may overestimate action values under certain conditions, a double Q network is applied to fit the Q function better. In the agent training process, the main network Q is primarily used to find the action a_{t+1} with the maximum Q value at the next time step, and then a target network $\hat{Q}$, which has the same structure as the Q-network, is adopted to estimate the Q-value of the action a_{t+1}. Since the target Q-value of the action a_{t+1} may not be the maximum in $\hat{Q}$, this procedure can effectively avoid overestimating suboptimal actions. The weights of the main Q-network are copied to the target network at a regular time step.

After drawing from the replay buffer, the main Q-network is trained by minimizing a sequence of the loss function:

$$L(\theta) = E_{(s_t, a_t) \sim p}[(y_i - Q(s_t, a_t; \theta, \alpha, \beta))^2] \tag{12}$$

where y_i is the target Q-value for iteration i computed by target network $\hat{Q}$, and p is the probability distribution of state-action pair (s_t, a_t). The DRQN network weights can be updated using the stochastic gradient with the gradient of the loss function $L(\theta)$. To avoid the algorithm from falling into the local optimum, a soft update method is adopted, which is:

$$\hat{\theta} \leftarrow \zeta\theta + (1 - \zeta)\hat{\theta} \tag{13}$$

where θ and $\hat{\theta}$ are the parameters of the main Q network and the target Q network, respectively, and ζ is the coefficient of the soft update.

B. QMIX Method

One main issue with DDRL is how to effectively learn the function and fit a proper approximation function when the parameters of the joint action-value function increase exponentially with the number of agents. Considering the complexity of the environment and the uncertainty of inter-agent communication, the problem of DDRL is aimed at a decentralized, partially observable Markov decision process (Dec-POMDP) [20].

QMIX is a monotonic value function factorization algorithm for DDRL, which maximizes the joint action-value function in a Dec-POMDP. This approach utilizes a hybrid network to combine the local value functions of agents and use global states information in the training process, to improve the performance of the DQN algorithm. QMIX adopts the "centralized training and distributed execution" framework, which uses global states in training and partial observations in execution.

Assuming that, at time slot t, agent i has partial observation $o_{i,t}$, action $a_{i,t}$, and the global states of the system is s_t. Then, $\boldsymbol{\tau} = [\tau_1, \cdots, \tau_i, \cdots, \tau_{\mathrm{n}}]$ is the joint action-observation history, τ_i is the action-observation history of a single agent, $\boldsymbol{a} = [a_1, \cdots, a_i, \cdots, a_{\mathrm{n}}]$ is the joint action of multi-agents. $\pi_i(\tau_i)$ and $Q_i(\tau_i, a_i; \theta_i)$ are the strategy and Q functions of agent i, respectively. $Q_{\mathrm{tot}}(Q_1, \cdots, Q_{\mathrm{n}})$ is the joint Q function of multi-agents. It can be seen that $Q_i(\tau_i, a_i; \theta_i)$ is related to τ_i, not to global state information s_t.

The structure of the QMIX network is shown in Fig. 1. The mixing network uses positive weights W_1 and W_2 to satisfy the monotonicity constraint, which is:

$$\frac{\partial Q_{\mathrm{tot}}}{\partial Q_i} \geq 0, \ \forall i \in \{1, 2, \cdots, n\} \tag{14}$$

As the Q functions of agents have the same monotonicity in the QMIX network, the maximization of the joint Q function

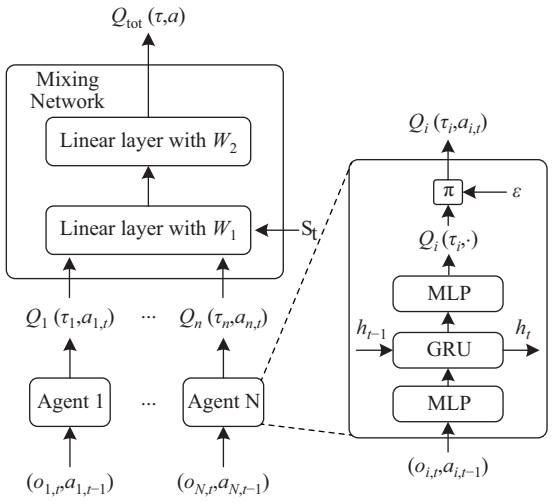


Fig. 1. The structure of the QMIX network.

is equivalent to the maximization of each local Q function. Therefore, we can obtain the optimal joint strategy by:

$$\arg\max_{\theta} Q_{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{a}; \boldsymbol{\theta}) = \begin{pmatrix} \arg\max_{\theta_1} Q_1(\tau_1, a_1; \theta_1) \\ \cdots \\ \arg\max_{\theta_n} Q_n(\tau_n, a_n; \theta_n) \end{pmatrix} \quad (15)$$

As shown in Fig. 1, each agent adopts DRQN to fit its Q function $Q_i(\tau_i, a_i; \theta_i)$. Considering the inputs of the APCC problem are values, we replaced the two convolutional layers with two full-connected layers in DRQN. Besides, we chose Gated Recurrent Unit (GRU) algorithm rather than LSTM in the Recurrent part since GRU is less computationally expensive. DRQN uses the observation $o_{i,t}$ and the action $a_{i,t-1}$ as input to calculate the Q value through a greedy algorithm with parameter ε.

Finally, the loss function of QMIX is presented as:

$$L(\theta) = \sum_{i=1}^{n} E_{(\boldsymbol{s},\boldsymbol{a})\sim p}[(y_i^{\text{tot}} - Q_{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{a}, \boldsymbol{s}; \boldsymbol{\theta}))^2] \quad (16)$$

where $y_i^{\text{tot}} = r_i + \gamma_i \max_{a'} \hat{Q}(\boldsymbol{\tau}', \boldsymbol{a}', \boldsymbol{s}'; \hat{\boldsymbol{\theta}})$, $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_n]$

C. Centralized Training Algorithm of DDRL

Based on the QMIX method introduced in Section III-B, qw propose a training structure of DDRL to obtain the NE of the APCC stochastic game. Before starting the training, the networks are initialized with random weights, and a replay buffer is established. In each episode, APCs obtain their partial observations from the system state at the beginning. Before APCs take their actions, we perform "do-nothing" actions for each agent in the simulation system, which can predict the next observation provided by the Grid2Op platform [21]. This procedure is to check whether the system is in danger. We predefine the danger status as that when the line capacity usage of any lines is above 0.95, or there is a system failure. If the system is not in danger, APCs still take "do-nothing" actions. When the danger flag is true in the simulation system, APCs explore their actions in their action spaces.

Considering that the action space of each APC is large enough, we should guide the exploration of each APC to improve the algorithm performance. Before exploring, each APC explores all actions if there are any disconnected lines in its control region. Otherwise, it only explores N_k actions in the front rank of all actions' Q-value. In short, these N_k actions are called "top N_k actions." Among the "top N_k actions," the action with the best simulation reward is chosen. Another set of N_{kb} actions in the front rank exclude the "top N_k actions," which are explored if there are no valid or effective actions. Furthermore, we choose an action with the best reward in historical actions when the APC cannot explore a feasible action.

After action selection is completed, the joint action composed of APC selections is executed in the environment. After obtaining the next state and reward of APCs, we store $(s_t, a^t, s_{t+1}, r_t, d_t)_\tau$ to the replay buffer with specific rules. When the memory size of the replay buffer reaches a threshold, batch samples are used to calculate the Q-values in the target QMIX network. The parameters of the QMIX network are updated based on the loss computed in (17). The training procedure operates iteratively until a specified

Algorithm 1: Distributed Active Power Correction Control (DAPCC)

Initialize DRQN network for each APC with random weights θ_i, initialize mixing network with random weights W_1 and W_2, initialize Replay Buffer Δ
for *episode = 1 to I* **do**
  Reset the environment
  for *t = 1 to T* **do**
    Obtain the partial observations $o_{1,t}, \cdots, o_{i,t}, \cdots, o_{N,t}$ of each agent from state s_t, check the danger flag (True for overload or system failure, False for normal state)
    if *danger flag is True* **then**
      for *i = 1 to N* **do**
        Choose the feasible action with best simulation reward in N_k in the condition of the gameover flag in simulation is True (True for overload or system failure, False for normal state)
        if *feasible action is None* **then**
          Choose the feasible action with the best simulation reward in N_{kb} in the condition of the gameover flag in simulation is True
          if *feasible action is None* **then**
            choose an action with the best reward in historical actions
          end if
        end if
      end for
      Validate the actions in the simulation system and combine each agent's action as the output action $a^t = (a_1^t, \cdots, a_i^t, \cdots, a_N^t)$
    else
      select do-nothing action $(a_1^0, \cdots, a_i^0, \cdots, a_N^0)$ as the output action a_t
    end if
    Execute action a^t in the environment and obtain next state s_{t+1}, reward r_t and d_t, store the transition $(s_t, a^t, s_{t+1}, r_t, d_t)_\tau$ in Δ
    Sample a batch of transitions from Δ
    Calculating the Q-values in target QMIX network: $y_j^{\text{tot}} = \begin{cases} r_i & \text{if } d_t \text{ is True} \\ r_i + \gamma_i \max_{a'} \hat{Q}(\tau', a', s'; \hat{\theta}) & \text{otherwise} \end{cases}$
    Update QMIX-network by losses: $L(\theta) = \sum_{i=1}^{n} E_{(s,a)\sim p}[(y_i^{\text{tot}} - Q_{\text{tot}}(\tau, a, s; \theta))^2]$
    Hard copy main network weights θ to the target network weights $\hat{\theta}$ regularly
    Update state $s_t = s_{t+1}$
  end for
end for

number of episodes is reached. The algorithm is denoted as DDRL Training Method for APCC, which is illustrated in Algorithm 1.

D. Distributed Execution Structure for APCC

Based on the training algorithm proposed in Section III-C, we summarize an online execution procedure to obtain the NE of the APCC game. In APCC, the primary purpose of agents is to maintain the normal operation of the power grid under different operation conditions. The system characteristics of high dimensionality and non-linearity make it challenging to predict the impact of each control action. For example, when the disturbance is minor, only a few agents taking action may outperform the strategy of all the agents taking action. Hence, an action optimization mechanism is proposed to obtain the optimal joint actions of APCs, whose flowchart is shown in Fig. 2.

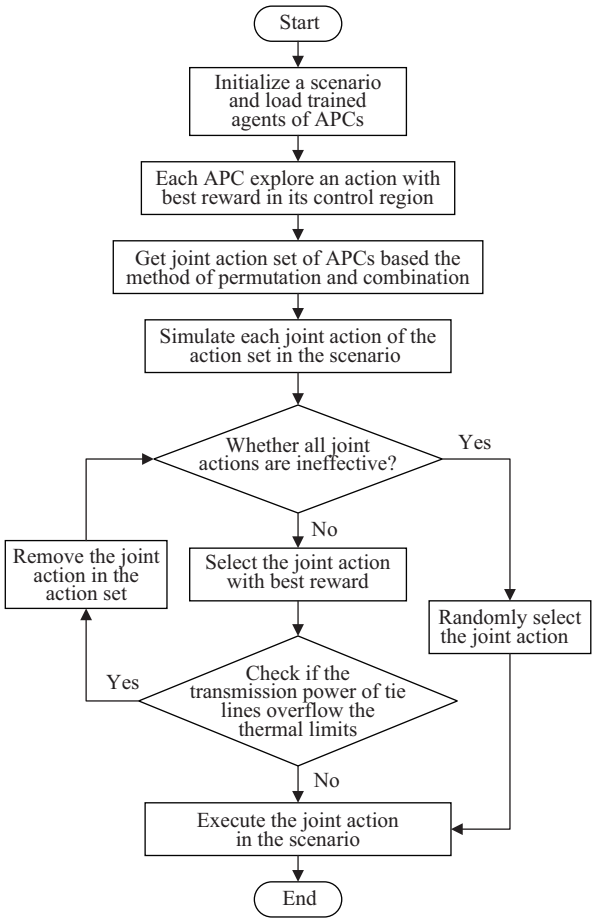


Fig. 2. The flowchart of distributed execution.

We use permutations and combinations to obtain the joint action set of APCs. This mechanism can improve the adaptability of multiple agents control. The action exploring each APC is the same as the method mentioned in Algorithm 1. When executing in a scenario, we need to simulate each joint action of the action set to figure out whether it is effective. If all actions in the action set are ineffective, a randomly selected joint action is adopted. Otherwise, we select the joint action with the best reward. After action selection, the transmission power of tie lines is checked to determine whether the joint action is executed. If the joint action leads to an overload of any tie lines, we remove it and search for another feasible joint action. It can be seen that we primarily consider the transmission power of tie lines to ensure the stability of power flow interfaces.

IV. Case Study

In this section, the proposed DAPC method is evaluated on an open-source platform, "Grid2Op". To provide a typical application scenario for the DAPCC application, the 118-bus grid provided by the "2020 Learning to Run a Power Network - Neurips Track 2" global competition cases [22] is used to evaluate the algorithm performance. In the 118-bus grid, we divided the centralized control region into three distributed control regions, as shown in Fig. 3. Three agents based on the QMIX method are implemented to control the elements of three regions, respectively. We chose a multi-mix dataset—a set of cases with a varying number of renewables, as shown in Fig. 4.

Each mixed dataset includes 576 scenarios covering each month for 48 years. Each scenario contains data for 28 continuous days with a 5-minute resolution. The penetration rate of RESs changes in different scenarios, and the maintenance of lines is considered. We need to train our agents to adapt to renewable energy production in the grid with an increasing rate of less controllable renewable energies over the years. After training, the trained agents are tested on hidden new mixed datasets that are not present in the training set, to assess the adaptability of the agent. The 24 test scenarios are randomly extracted from the data every month and cover the characteristics of typical scenarios through one year.

As we implement three agents in our case study, the network structure of the QMIX method consists of three identical DRQNs and a mixing network. The DRQNs are composed of two full-connected layers and a GRU layer with 512 neurons. The mixing network is composed of a deep neural network (DNN) with 256 neurons. The rectified linear unit (ReLU) is used as an activation function, and the batch size is set to 64. RMS is adopted for the QMIX network, and the learning rate is set as 0.0005. The maximum number of steps in an episode is set to 4000. The reaction time and recovery time of the APCC problem are set to 3 time steps, i.e., 15 minutes. The simulations are conducted on a Linux server with 3 GPUs.

A. Centralized Learning Capability

In this part, we adopted the online centralized training algorithm proposed in Section III-C to train the three APC agents. Through the Monte Carlo tree search (MCTS) described in Part I, the amounts of selected effective actions of Region 1, Region 2, and Region 3 are 46, 540, and 565, respectively. It can be seen from the results of MCTS that the network complexity of Region 2 and Region 3 is higher than that of Region 1. The effective action matrixes of the three regions are used as action spaces for three APCs, respectively. Specifically, we add "do-nothing" action to the action space of each APC, which is explained in Section III-C. In each episode, the

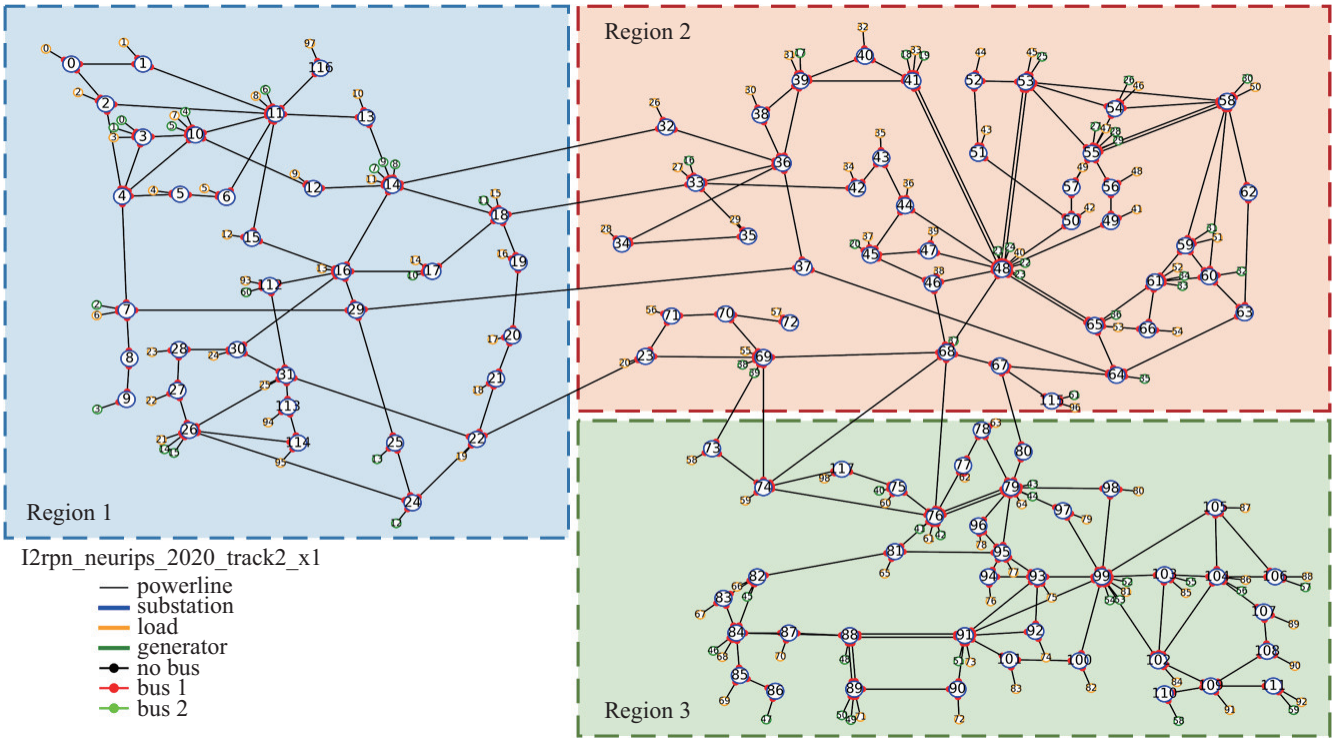


Fig. 3. The topology of the 118-bus power grid.

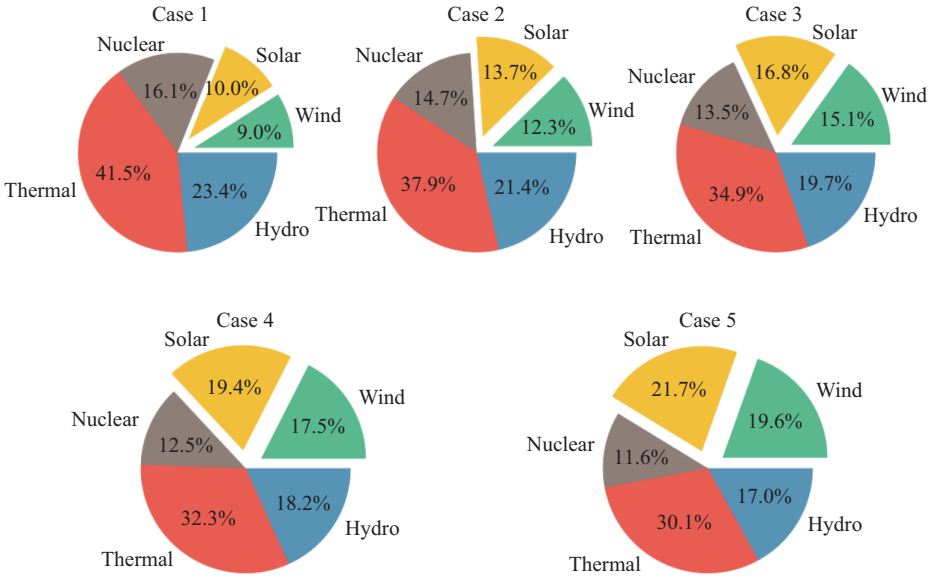


Fig. 4. Energy profiles of each case.

scenarios of various RES penetration rates in every case are randomly selected in the training process.

After 1000 episodes of training, we take the moving average value of cumulative rewards for every 200 episodes; the cumulative reward curve of the DAPCC is shown in Fig. 5. In the early phase of training, scenarios with a variable penetration rate of RESs are randomly selected. As a result, the trend of cumulative reward varies dramatically. From 200 episodes to 600 episodes, the curve of cumulative reward becomes stable and increasing. It can be seen that the cumulative reward finally begins to converge at around 800 episodes, which indicates that the NE of the APCC game may have been obtained.

B. Distributed Executing Performance

In this part, we implement the trained APC agents in Section IV-A to evaluate whether they can solve the problem of APCC. The information on 24 test scenarios is provided in Table I, which contains all kinds of cases.

In the evaluation process, we test our trained APCs in a sequence of the penetration rate of RESs and limit the maximum alive steps of a scenario to 4000. In Fig. 6, the

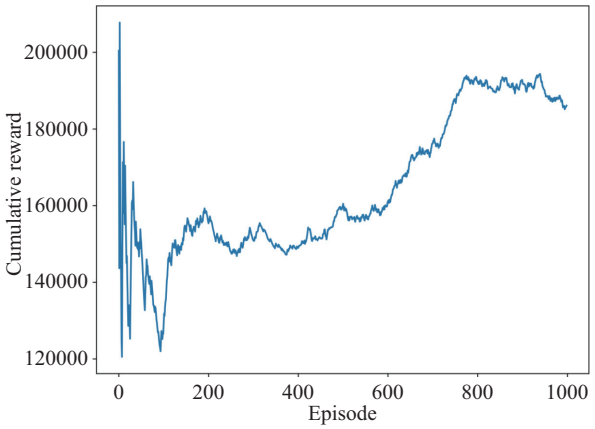


Fig. 5. Moving average curve of the cumulative reward of the proposed DAPCC.

TABLE I
THE INFORMATION OF 24 TEST SCENARIOS

Mix ID	Scenario numbers	The active load consumption (MV) in each scenario					
Case 1	3		2163	2217	2147		
Case 2	4		2081	2099	2217	2163	
Case 3	6	2104	2033	2228	2157	2145	2183
Case 4	6	2143	2161	2142	2130	2188	2143
Case 5	5		2093	1999	2175	2156	2110

legend "max-3-actions" (M3A) denotes that the number of APC actions allowed in a single time slot is limited to 3. Meanwhile, the legend "max-1-actions" (M1A) denotes that the number of actions is limited to 1. It may be noticed that the performance of the "do-nothing" action is used to compare with other schemes.

In the first 7 scenarios, the mixed data of environments are case 1 and case 2. It can be seen that the control action of DAPCC is effective, which can keep the power system alive longer. Considering the scenarios in case 1 and case 2 contain only 19% RESs, the performance of the M3A scheme is similar to that of the M1A scheme. In the following 12 scenarios of cases 3 and 4, the M3A scheme performs more effectively and stably than the M1A scheme, which indicates that the distributed executing scheme we proposed is beneficial to the APCC problem. The last 5 scenarios, which contain 41.3% RESs in case 5, are difficult for the agent with only BBS control. However, the APCs with the M3A scheme are also better than other action schemes. Notice that the average decision time for each time-step of case 1, case 2, case 3, and case 4 is around 40 milliseconds. This indicates that the proposed method has practicability in the APCC problem.

C. Adaptability of Proposed Control Method

To fully evaluate the performance of DAPCC, we randomly selected two scenarios, which are case 2 and case 4, to test our trained APCs. The comparison of the M3A action scheme and M1A action scheme is still set up in this part. The alive steps of the M3A scheme and M1A scheme are almost the same in case 2, while they are different in case 4.

We first investigated the maximum line capacity usage of each control region to reflect the effectiveness of the control strategy directly. As shown in Fig. 7, the system blackouts with a "do-nothing" action at around 500 steps. This illustrates that the plugged-in RESs lead to heavy overload in the power system if no control measure is taken. In Fig. 7(a), although the time-steps during which the system is "alive" using the M1A scheme and M3A scheme both reach 3000steps, the maximum

TABLE II
THE PARAMETERS OF TIE LINES

Line ID	Line Origin bus	Line extremity bus	Transmission capacity (MW)
108	14	32	208.10
109	18	33	**226.90**
117	29	37	456.00
94	22	23	674.20
7	69	73	480.00
8	69	74	**436.50**
9	68	74	681.80
12	68	76	**682.50**
185	67	80	997.10

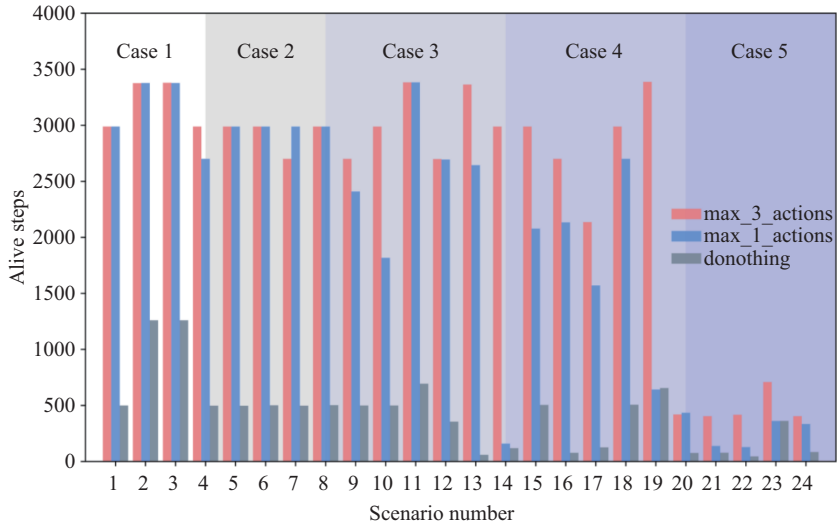
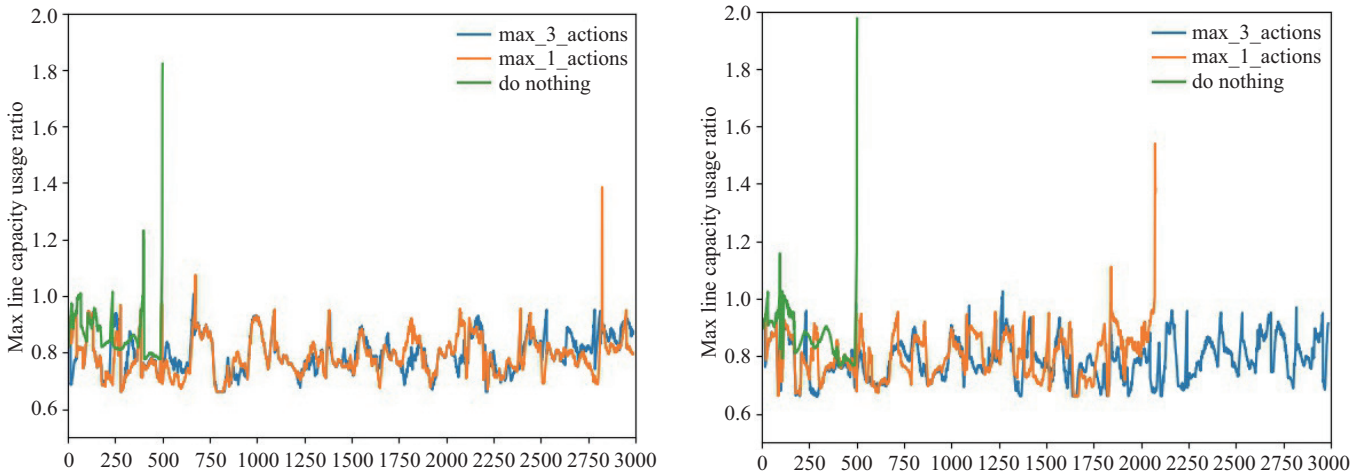


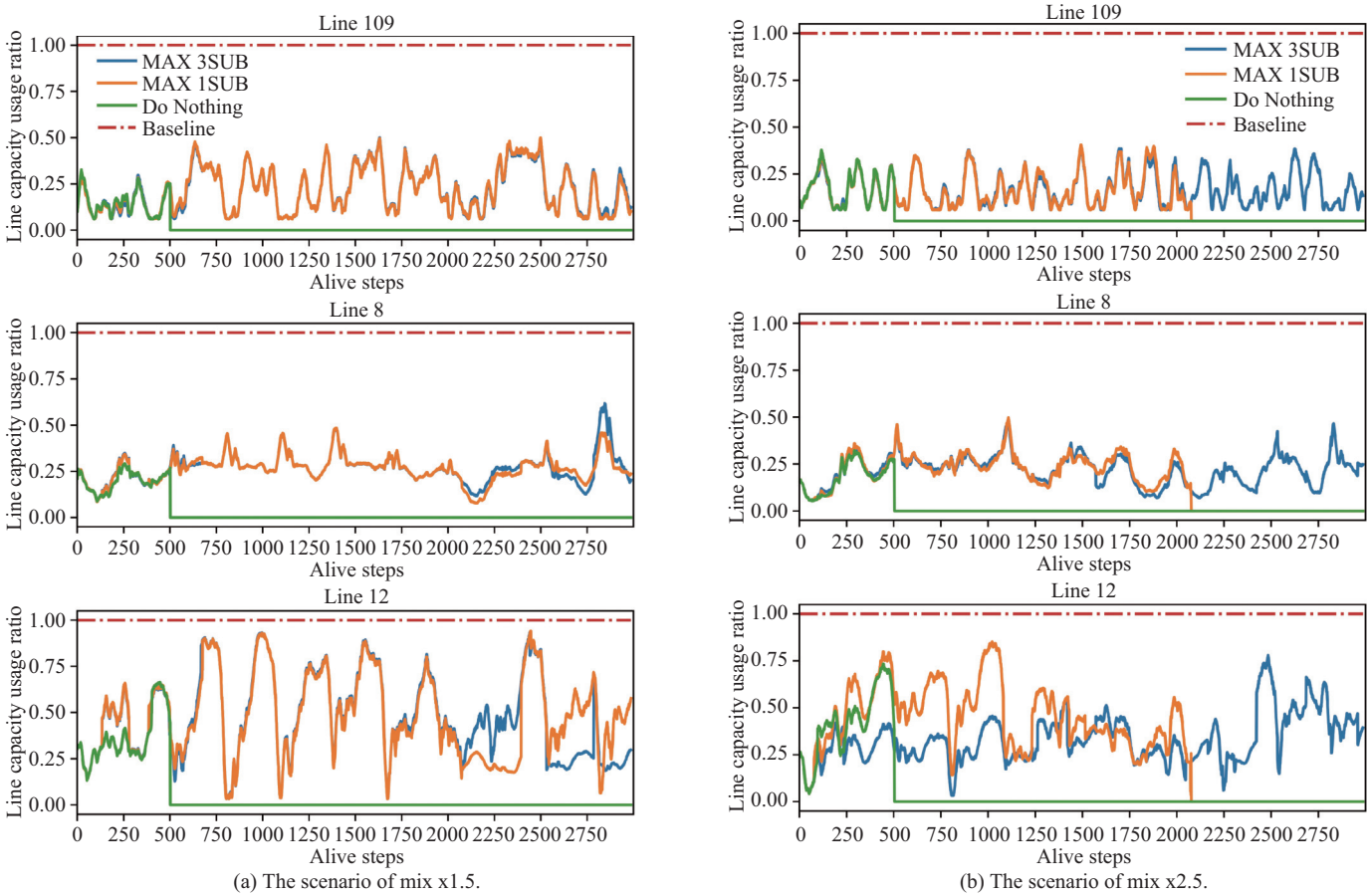
Fig. 6. Performance of DAPCC in 24 test scenarios.



(a) System operating status under different schemes in the scenario of case 2.

(b) System operating status under different schemes in the case 4 scenario.

Fig. 7. The comparison of system operating status under different schemes.



(a) The scenario of mix x1.5.

(b) The scenario of mix x2.5.

Fig. 8. The line capacity usage of tie lines in the scenarios.

line capacity usage ratio of the M1A scheme increase sharply at around 2800steps, which indicate that the robustness of the M3A scheme is better than that of the M1A scheme. From Fig. 7(b), it can be seen that with the penetration rate of RESs increasing, the "do-nothing" scheme and M1A scheme cannot eliminate or alleviate the overload as well as the M3A scheme does. This demonstrates the adaptability of our proposed control method.

After investigating the control performance of each APC, we focused on the line capacity usage of tie lines to evaluate the coordination between the APCs. The parameters of tie lines are shown in Table II. We recorded the line capacity usage in the test scenarios of case 2 and case 4. As shown in Fig. 3, the buses 18, 33, 68, 69, 74, and 76, which connect more

loads and lines than others, are regarded as heavy load buses. Hence, we checked the transmission power of line 109, line 8, and line 12, as shown in Fig. 8. It can be seen that the line capacity usage of these three tie lines can always be controlled below the baseline. This illustrates that the DAPCC method is available for multi-region coordinative control.

V. Conclusions

To improve the adaptability of the DRL algorithm applied in the APCC problem, a DAPCC method is proposed to conduct the framework of "centralized training and distributed executing" scheme for a large-scale power system. Based on the stochastic game theory, we adopted a QMIX method to obtain the Nash Equilibrium of APCs. Considering the cooperation of APCs, an action-optimization mechanism is proposed to obtain the global optimality of joint actions. The case studies demonstrate the convergence of centralized learning capability and the adaptability of distributed executing performance in large-scale APCC with plugged-in RESs. The application of DRL for controlling complex and complicated power systems is still in its infancy. Future work on this topic shall involve multi-faceted application scenarios, and these applications are expected to solve power system control problems with ultra-high dimensionality and non-linearity.

References

[1] X. P. Li, P. Balasubramanian, M. Sahraei-Ardakani, M. Abdi-Khorsand, K. W. Hedman, and R. Podmore, "Real-time contingency analysis with corrective transmission switching," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2604–2617, Jul. 2017.

[2] H. R. Diao, M. Yang, F. Chen, and G. Z. Sun, "Reactive power and voltage optimization control approach of the regional power grid based on reinforcement learning theory," *Transactions of China Electrotechnical Society*, vol. 30, no. 12, pp. 408–414, Jun. 2015.

[3] J. Y. Zhang, C. Liu, J. Si, J. Song, and Y. S. Su, "Deep reinforcement leaming for short-term voltage control by dynamic load shedding in China southem power grid," in *Proceedings of 2018 International Joint Conference on Neural Networks*, 2018, pp. 1–8.

[4] Q. L. Yang, G. Wang, A. Sadeghi, G. B. Giannakis, and J. Sun, "Two-timescale voltage control in distribution grids using deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2313–2323, May 2020.

[5] J. J. Duan, D. Shi, R. S. Diao, H. F. Li, Z. W. Wang, B. Zhang, D. S. Bian, and Z. H. Yi, "Deep-reinforcement-learning-based autonomous voltage control for power grid operations," *IEEE Transactions on Power Systems,* vol. 35, no. 1, pp. 814–817, Jan. 2020.

[6] O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser, J. Quan, S. Gaffney, S. Petersen, K. Simonyan, T. Schaul, H. van Hasselt, D. Silver, T. Lillicrap, K. Calderone, P. Keet, A. Brunasso, D. Lawrence, A. Ekermo, J. Repp, and R. Tsing, "StarCraft II: A new challenge for reinforcement learning," arXiv:1708.04782, Aug. 2017. [Online]. Available: https://arxiv.org/pdf/1708.04782.pdf.

[7] B. K. Petersen, J. C. Yang, W. S. Grathwohl, C. Cockrell, C. Santiago, G. An, and D. M. Faissol, "Deep reinforcement learning and simulation as a path toward precision medicine," *Journal of Computational Biology*, vol. 26, no. 6, pp. 597–604, Jun. 2019.

[8] J. Y. Chen, B. D. Yuan, and M. Tomizuka, "Model-free deep reinforcement learning for urban autonomous driving," in *Proceedings of 2019 IEEE Intelligent Transportation Systems Conference*, 2019, pp. 2765–2771.

[9] C. H. Liu, Z. Y. Chen, J. Tang, J. Xu, and C. Z. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 2059–2070, Sept. 2018.

[10] M. Khodayar, G. Liu, J. Wang and M. E. Khodayar, "Deep learning in power systems research: A review," in *CSEE Journal of Power and Energy Systems*, vol. 7, no. 2, pp. 209–220, Mar. 2021, doi: 10.17775/CSEEJPES.2020.02700.

[11] Z. Zhang, D. Zhang and R. C. Qiu, "Deep reinforcement learning for power system applications: An overview," in *CSEE Journal of Power and Energy Systems*, vol. 6, no. 1, pp. 213–225, Mar. 2020, doi: 10.17775/CSEEJPES.2019.00920.

[12] I. Adamski, R. Adamski, T. Grel, A. Jędrych, K. Kaczmarek, and H. Michalewski, "Distributed deep reinforcement learning: Learn how to play Atari games in 21 minutes," in *Proceedings of the 33rd International Conference on High Performance Computing*, 2018, pp. 370–388.

[13] K. X. Lin, R. Y. Zhao, Z. Xu, and J. Y. Zhou, "Efficient large-scale fleet management via multi-agent deep reinforcement learning," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1774–1783.

[14] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proceedings of the 10th International Conference Machine Learning*, 1993, pp. 330–337.

[15] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel, "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2018, pp. 2085–2087.

[16] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," in International Conference on Machine Learning. PMLR, 2018, pp. 4295–4304.

[17] A. Marot, B. Donnot, C. Romero, B. Donon, M. Lerousseau, L. Veyrin-Forrer, and I. Guyon, "Learning to run a power network challenge for training topology controllers," *Electric Power Systems Research*, vol. 189, pp. 106635, Dec. 2020.

[18] Universidad Nacional de Colombia. L2RPN-NEURIPS-2020. [Online]. Available: https://github.com/unaioperator/l2rpn-neurips-2020.

[19] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPs," arXiv:1507.06527, Jan. 2017. [Online]. Available: https://arxiv.org/pdf/1507.06527.pdf.

[20] C. Amato, G. Chowdhary, A. Geramifard, N. K. Üre, and M. J. Kochenderfer, "Decentralized control of partially observable Markov decision processes," in *Proceedings of the 52nd IEEE Conference On Decision and Control*, 2013, pp. 2398–2405.

[21] RTE-France. Grid2Op. [Online]. Available: https://github.com/rte-france/Grid2Op.

[22] RTE-France. L2RPN NEURIPS 2020 - robustness track. [Online]. Available: https://competitions.codalab.org/competitions/25426.

Siyuan Chen (S'19) received the M.S. degree in School of Electrical Engineering and Automation from Wuhan University, China, in 2018. He is currently pursuing a Ph.D. degree in School of Electrical Engineering and Automation from Wuhan University. His research interests include power system operation and control, artificial intelligence, and machine learning.

Jiajun Duan (S'13–M'18) received the B.S. degree in Power System and its Automation from Sichuan University, Chengdu, China, and M.S. degree in Electrical Engineering at Lehigh University, Bethlehem, PA in 2013 and 2015, respectively, and the Ph.D. degree in Electrical Engineering from Lehigh University in 2018. Currently, he is a Research Scientist in GEIRI North America, San Jose, CA, USA. His research interests include artificial intelligence, power system, power electronics, control systems, and machine learning.

Yuyang Bai received the B.S. degree in School of Electrical Engineering and Automation from Wuhan University, China, in 2019. He is currently pursuing an M.S. degree in School of Electrical Engineering and Automation from Wuhan University. His research interests include artificial intelligence, power system operation and control, and machine learning.

Jun Zhang (M'09–SM'14) received the B.E. and M.E. degrees in Electrical Engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2003 and 2005, respectively, and the Ph.D. degree in Electrical Engineering from Arizona State University, Phoenix, AZ, USA, in 2008. He is currently a Professor in the School of Electrical Engineering and Automation, Wuhan University. He has authored/coauthored more than 80 peer-reviewed publications. His research expertise is in the areas of complex systems, artificial intelligence, knowledge automation, and their applications in smart grid.

Di Shi (M'12–SM'17) received the B.S. degree in Electrical Engineering from Xi'an Jiaotong University, Xi'an, China, in 2007, and M.S. and Ph.D. degrees in Electrical Engineering from Arizona State University, Tempe, AZ, USA, in 2009 and 2012, respectively. He currently leads the AI & System Analytics Group at GEIRI North America, San Jose, CA, USA. His research interests include WAMS, Energy storage systems, and renewable integration. He is an Editor of IEEE Transactions on Smart Grid.

Zhiwei Wang ((M'16–SM'18)) received the B.S. and M.S. degrees in Electrical Engineering from Southeast University, Nanjing, China, in 1988 and 1991, respectively. He is President of GEIRI North America, San Jose, CA, USA. Prior to this assignment, he served as President of State Grid US Representative Office, New York City, from 2013 to 2015, and President of State Grid Wuxi Electric Power Supply Company from 2012-2013. His research interests include power system operation and control, relay protection, power system planning, and WAMS.

Xuzhu Dong (M'02–SM'10) received the Ph.D. degree in High Voltage Engineering from Tsinghua University, Beijing, China, in 1998, and the Ph.D. degree in Electrical Engineering from Virginia Tech University, USA, in 2002. He is currently a Professor in the School of Electrical Engineering and Automation, Wuhan University. His research interests include distribution automation, energy storage, renewable energy and micro-grid.

Yuanzhang Sun (M'99–SM'01) received the Ph.D. degree in Electrical Engineering from Tsinghua University, Beijing, China, in 1988. He is currently a Professor in the School of Electrical Engineering and Automation, Wuhan University, and a Chair Professor in the Department of Electrical Engineering and Vice Director of the State Key Laboratory of Power System Control and Simulation, Tsinghua University. His main research interests are power system dynamics and control, wind power, voltage stability and control, and reliability.