MADDPG-based Active Distribution Network Dynamic Reconfiguration with Renewable Energy

Changxu Jiang, Zheng Lin, Chenxi Liu, Feixiong Chen, and Zhenguo Shao

***Abstract*—The integration of distributed generations (DG), such as wind turbines and photovoltaics, has a significant impact on the security, stability, and economy of the distribution network due to the randomness and fluctuations of DG output. Dynamic distribution network reconfiguration (DNR) technology has the potential to mitigate this problem effectively. However, due to the non-convex and nonlinear characteristics of the DNR model, traditional mathematical optimization algorithms face speed challenges, and heuristic algorithms struggle with both speed and accuracy. These problems hinder the effective control of existing distribution networks. To address these challenges, an active distribution network dynamic reconfiguration approach based on an improved multi-agent deep deterministic policy gradient (MADDPG) is proposed. Firstly, taking into account the uncertainties of load and DG, a dynamic DNR stochastic mathematical model is constructed. Next, the concept of fundamental loops (FLs) is defined and the coding method based on loop-coding is adopted for MADDPG action space. Then, the agents with actor and critic networks are equipped in each FL to real-time control network topology. Subsequently, a MADDPG framework for dynamic DNR is constructed. Finally, simulations are conducted on an improved IEEE 33-bus power system to validate the superiority of MADDPG. The results demonstrate that MADDPG has a shorter calculation time than the heuristic algorithm and mathematical optimization algorithm, which is useful for real-time control of DNR.**

***Index Terms*—Distribution network reconfiguration, active distribution network, deep deterministic policy gradient, multi-agent deep reinforcement learning.**

I. Introduction

In order to reach carbon peaking and carbon neutrality, distributed generations (DGs), represented by wind turbines and photovoltaics, will be widely integrated into the distribution network in the coming decades [1]. DGs can help regulate the bus voltage of distribution networks and minimize network active power loss [2]. However, due to the randomness and fluctuations of DGs, the connection of excessive DGs will dramatically change the power distribution of the networks, leading to voltage imbalance, harmonic pollution and other problems that put the safety, stability, and economy of the distribution networks at risk [3], [4]. Additionally, as a substantial quantity of novel equipment like electric automobiles and power electronic equipment are integrated into the distribution networks, the direction of power flow will change, and problems such as voltage distribution and harmonics will also endanger the security and stability of power systems [5]. Distribution network reconfiguration (DNR) is a primary control approach aimed at these problems by turning on/off the states of switches to control the topology of the distribution networks [6]. Therefore, exploring DNR technology is advantageous for enhancing the security, stability and economy of distribution networks.

Numerous experts have conducted research on the algorithms of DNR, which can be categorized into three types: mathematical optimization algorithms, heuristic algorithms, and advanced intelligence algorithms. Mathematical optimization algorithms utilize mathematical theories for DNR solutions, such as the simplex method, mixed integer nonlinear programming (MINLP), and second-order cone programming (SOCP). Reference [7] proposes a MINLP model to maximize power system benefits by coordinating electric vehicle charging/discharging strategies and dynamic DNR strategies. Reference [8] proposes a mixed integer SOCP model to minimize daily active power loss by adopting hourly reconfiguration, and it is solved via the Mosek solver. The mathematical optimization algorithms have the advantage of typically ensuring global optimality of solutions. However, when the network grows larger, the number of variables in the solution significantly increases, which poses challenges in solving it efficiently and effectively. Furthermore, although these references all mention the outputs of the DGs, they regard DGs as negative loads and do not take into account the randomness of DGs. The mathematical

Received: December 12, 2023
Accepted: May 20, 2024
Published Online: November 1, 2024
Changxu Jiang, Zheng Lin, Chenxi Liu, Feixiong Chen, and Zhenguo Shao (corresponding author) are with the School of Electrical Engineering and Automation, Fuzhou University, Fuzhou 350108, China (e-mail: cxjiang@fzu.edu.cn; 943562135@qq.com; 220120055@fzu.edu.cn; feixiongchen@yeah.net; shao.zg@fzu.edu.cn).
DOI: 10.23919/PCMP.2023.000283

optimization algorithms dealing with stochastic optimization problems may encounter problems of rapidity and it is difficult to deal with the uncertainty of sources and loads. For example, the solution results will be conservative to some extent when robust optimization methods are employed.

The second type of the solving algorithm of DNR is the heuristic algorithm, which primarily includes the branch exchange method, particle swarm optimization (PSO) and genetic algorithm (GA). Reference [9] formulates a multi-period DNR method to decrease the number switch operations while satisfying the hourly capacity requirements. The proposed model is solved by the hybrid PSO algorithm for each partitioned time period. Reference [10] utilizes GA with variable population size to the DNR model, which improves the optimization efficiency of GA. Although heuristic algorithms are simpler to model than mathematical optimization algorithms, global searching and computational speed are challenging, making them less adaptable to DNR decisions with a large number of mixed variables and nonlinear constraints.

The third type of the solving algorithms of DNR is the advanced intelligence algorithm, specifically referring to the deep learning (DL) algorithms represented by an artificial neural network (ANN), reinforcement learning (RL) represented by Sarsa, Q-learning and deep reinforcement learning (DRL) represented by deep Q-network (DQN) [11]–[14]. DL algorithms have efficient feature extraction capabilities, enabling them to learn nonlinear mappings between system states and reconfiguration strategies. However, DL algorithms typically require numerous labeled data for training. Obtaining high-quality labeled data in the field of power systems poses challenges and incurs significant costs. The above algorithms have their own defects in solving the DNR problem. Hence, it is imperative to conduct research on novel approaches to tackle current DNR problems.

The development of RL and DRL has introduced novel concepts to address the problem of DNR. RL is an algorithm with self-learning ability, where agents interact with the environment to learn how to make a sequence of decisions aimed at maximizing the cumulative expected return. To effectively solve the problem of large state and action space dimensions of traditional RL, DRL algorithms combine the feature extraction capability of DL with the self-learning capability of RL. DRL algorithms have stronger generalization ability, and can effectively solve the sequential decision problem with multiple stochastic factors. Reference [15] proposes a method for adjusting tie-line power based on DQN and experiments are conducted on the IEEE 39-bus system. Reference [16] improves the traditional DQN and proposes a noisy net DQN-based DNR approach, which speeds up the training process of the agent. Reference [17] proposes a proximal policy optimization (PPO) to analyze the optimal power flow with DGs and storage systems. The aforementioned research demonstrates the extensive utilization of the DRL algorithm in power systems.

Multi-agent DRL (MADRL) is an expansion of single-agent DRL. The MADRL framework involves the presence of numerous agents in the environment and these agents might have either competitive or cooperative connections [18], [19]. Consequently, compared with single-agent DRL, MADRL can effectively improve the solving speed and obtain better performance. The MADRL algorithm mainly includes multi-agent deep Q-network (MADQN), multi-agent policy gradient (MAPG), and multi-agent deep deterministic policy gradient (MADDPG). Among them, MADQN can hardly adapt to non-stationary environments, and MAPG may experience a sharp increase in variance as the number of agents grows. MADDPG improves upon the first two algorithms by employing a framework of centralized training with decentralized execution. When executing actions, agents output action policies according to their own local observations. When updating the networks, the agents will update the network parameters through the global observations and the action information of all agents, and the policies of agents will be updated towards the optimal policy.

Since the DNR model is a non-convex nonlinear stochastic mathematical optimization model, the traditional mathematical optimization method will encounter problems in rapidity and heuristic algorithms will take a long time to solve the model, which will not be beneficial for the control of active distribution networks. Furthermore, the uncertainty of active distribution networks presents a challenge for mathematical optimization algorithms and heuristic algorithms. To address these challenges, a MADDPG-based active distribution network dynamic reconfiguration approach is proposed to achieve the safety and economy of the distribution network under the environment with multiple uncertainty factors, such as DG output and load. In comparison to traditional mathematical optimization and heuristic algorithms, MADDPG does not rely on explicit model construction [20]. It solves complicated stochastic problems through interactions between agents and real or simulated environments, demonstrating the ability to map distribution network states to DNR solutions. The neural network generalization capability of MADDPG allows agents to make online decisions for similar states without retraining. Additionally, MADDPG can represent the uncertainty of DG handling through a state transition function. In this paper, the distribution network environment can be decomposed into multiple fundamental loops (FLs), which aligns perfectly with the multiple agents in the MADDPG algorithm. Assigning the control decisions

for each FL to different agents is more conducive to the convergence of agent control strategies.

The contributions of this work are the following:

1) The optimization problem of dynamic DNR is transformed into a partially observable Markov decision process (POMDP) and a MADDPG-based active distribution network dynamic reconfiguration approach is proposed, which is a novel approach to DNR.

2) In the DNR, the loop-based coding method is adopted and each fundamental loop represents an agent of MADDPG. The agent will decide which switch to open based on the state of the active distribution network and historical experience.

3) The proposed approach adopts the framework of centralized training with decentralized execution, which increases the efficiency of training and decision-making of the agents.

4) The simulations are conducted on an improved IEEE 33-bus power system, and the results illustrate that MADDPG can improve the economy and safety of the distribution networks.

The remainder of this paper is organized as follows. The distribution network reconfiguration mathematical model is introduced in Section II. Loop-based coding method for DNR is introduced in Section III. Section IV details the proposed deep reinforcement learning modeling of DNR. The simulation cases are studied in Section V. Section VI presents the conclusion.

II. Distribution Network Reconfiguration Mathematical Model

In this study, a multi-objective dynamic reconfiguration model is constructed with the goal of minimizing the active loss and voltage offset. To achieve these goals, the objective function can be formulated as follows:

$$\max f = \sum_{t=1}^{T} \frac{P_{\text{loss},t} - P'_{\text{loss},t}}{P_{\text{loss},t}} + \sum_{t=1}^{T} \frac{\Delta V_t - \Delta V'_t}{\Delta V_t} \tag{1}$$

$$P_{\text{loss},t} = \sum_{i,j \in S_{\text{B}}} x_{ij,t} R_{ij} \frac{P_{ij,t}^2 + Q_{ij,t}^2}{V_{i,t}^2} \tag{2}$$

$$\Delta V_t = \sum_{i=1}^{N} \left| \frac{V_{i,\text{rate}} - V_{i,t}}{V_{i,\text{rate}}} \right| \tag{3}$$

where $P_{\text{loss},t}$ and $P'_{\text{loss},t}$ are the active loss before and after reconfiguration at time t, respectively; $x_{ij,t}$ represents the connection state of the *i-j* line at time t, $x_{ij,t}$ is 1 when the *i-j* line is closed and 0 when the *i-j* line is open; R_{ij} is the resistance of *i-j* line; $P_{ij,t}$ and $Q_{ij,t}$ represent the active and reactive power flowing through *i-j* line at time t, respectively; ΔV_t and $\Delta V_t'$ are voltage offsets before and after reconfiguration at time t, respectively; T is the total simulation time; N is the total number of buses; $V_{i,t}$ is the voltage amplitudes of bus i at time t; and $V_{i,\text{rate}}$ represents rated voltage of the bus i.

Additionally, the distribution network should satisfy the power flow balance constraint, expressed as follows:

$$P_{i,t} = V_{i,t} \sum_{j \in \Gamma_i} V_{j,t} [G_{ij} \cos \delta_{ij,t} + B_{ij} \sin \delta_{ij,t}] = P_{\text{DG},i,t} - P_{\text{load},i,t},\ i \in S_{\text{B}} \tag{4}$$

$$Q_{i,t} = V_{i,t} \sum_{j \in \Gamma_i} V_{j,t} [G_{ij} \sin \delta_{ij,t} - B_{ij} \cos \delta_{ij,t}] = Q_{\text{DG},i,t} - Q_{\text{load},i,t},\ i \in S_{\text{B}} \tag{5}$$

where $P_{i,t}$, $Q_{i,t}$ are the injected active and reactive power into bus i at time t; $P_{\text{DG},i,t}$, $Q_{\text{DG},i,t}$ are the active and reactive power generated by the DG of bus i at time t; $P_{\text{load},i,t}$, $Q_{\text{load},i,t}$ are the active and reactive load of bus i at time t; $V_{j,t}$ is the voltage amplitudes of bus j at time t; Γ_i is the set of buses adjacent to bus i; S_{B} is the set of buses in the distribution network; G_{ij}, B_{ij} are the conductance and susceptance of *i-j* line; and $\delta_{ij,t}$ is the voltage phase difference between bus i and j at time t.

Furthermore, the DNR also should satisfy the radial topology constraints, and the necessary and sufficient conditions are: 1) the number of closed branches should be equal to the number of network buses minus the number of substations; 2) the distribution network is connected.

The constraint of the above necessary and sufficient condition 1) is expressed as follows:

$$\sum_{b=1}^{N_{\text{L}}} x_b = N - N_{\text{S}} \tag{6}$$

where x_b denotes the branch status variable, and $x_b = 0$ means the switch of the bth branch is off, otherwise $x_b = 1$ means that the switch of the bth branch is on; N_{L} is the number of branches; N is the number of buses; and N_{S} is the number of substations.

Additionally, the constraint of the above necessary and sufficient condition 2) is achieved by a virtual network mirroring the original network's topology, which is formulated as follows:

$$\begin{cases} v_{ij} = -v_{ji},\ \forall (i,j) \in S_{\text{E}} \\ \sum_{j \in \Gamma_i} v_{ji} = 1,\ \forall i \in B_{\text{L}} \\ -x_b H \leqslant v_{ij} \leqslant x_b H,\ \forall (i,j) \in S_{\text{E}} \end{cases} \tag{7}$$

where v_{ij} denotes the virtual power flow from bus i to j; H is a large enough constant; S_{E} and B_{L} represent the set of branches and buses in the distribution network, respectively.

III. Loop-based Coding Schemes for Distribution Network Reconfiguration

An appropriate coding method is beneficial to improve the efficiency of solving the DNR model. There are three coding methods for branches: traditional branch coding method, branch group-based coding method and loop-based coding method.

According to the traditional branch-based coding method, each branch state in the network is coded with 0 or 1, where 0 denotes an open state and 1 denotes a closed state. The branch coding approach is straightforward, but it has the drawback of producing a large number of infeasible solutions, which has a negative impact on the efficiency of solving the DNR model.

The second coding method is based on the branch-group according to the network homeomorphism graph theory. Firstly, buses unrelated to the loops are deleted, and then buses with degree two are deleted, and the equivalent branches are merged to form a branch group. Each branch group is a set of optimization variables, and at most one branch is disconnected in each branch group, otherwise an island will be formed. For a network with M loops, only when M branches are open, the network may become radial.

The loop-based coding method regards each loop as an optimization variable, which contains all the remote-controlled switches (RCS) in a loop. A FL is defined as the smallest loop that does not contain other loops. This method ensures the radial constraint through the following two constraints: 1) Each FL at least keeps one branch in an open state, which avoids the formation of loops in the distribution network, formulated as (8); and 2) Only one branch of the common branches (CB) between the two FLs should be open, which ensures that there will be no isolated island in the network, shown in (9).

$$\sum_{b=1}^{M_l} x_{l,b} \leqslant M_l - 1, \ \forall l \in \{1,2,\cdots,L\} \tag{8}$$

$$\sum_{b=1}^{C_{\mathrm{B},lk}} x_{lk,b} \leqslant 1, \ \forall l \neq k \in \{1,2,\cdots,L\} \tag{9}$$

where $x_{l,b}$ represents the state of the bth branch in the lth loop; M_l is the number of RCS in the lth loop; L is the number of FLs of the network; $x_{lk,b}$ represents the state of the bth branch in the CB between the lth loop and the kth loop; $C_{\mathrm{B},lk}$ denotes the number branch of CB between the lth loop and the kth loop.

Through the loop-based coding, the solution space of the DNR model can be effectively reduced without the loss of feasible solutions, and the calculation cost is greatly reduced. For example, the IEEE 33-bus power system is shown in Fig. 1. The FLs and CBs of the distribution network are shown in Table I and Table II. When all switches are closed, there are 5 loops. The branch state of the lth loop of the distribution network can be represented by a vector $\boldsymbol{x}_l$:

$$\boldsymbol{x}_l = (x_{l,1}, x_{l,2}, \cdots, x_{l,b}, \cdots, x_{l,M_l}), \ \forall l \in \{1,2,\cdots,L\} \tag{10}$$

Then, the branch state of the whole network is represented by $\boldsymbol{x}$:

$$\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_l, \cdots, \boldsymbol{x}_L) \tag{11}$$

The number of switch combinations of the three coding methods and computing time are shown in Table III. Compared with the branch-based coding method, the branch-group-based coding and the loop-based coding method can effectively reduce the action combination by 68.67% and 80.22%, respectively. Obviously, the number of combinations of the loop-based coding method is the fewest, which significantly increases the solving efficiency for solving the DNR problem. The computing time is defined as the sum of the time for coding and power flow calculation of all potential DNR schemes of the different coding methods. It can be found from Table III that the branch-based coding time is 15788.73 s due to the large variable dimension. Due to the reduction of variable dimension, the calculation time of the method based on the branch-group-based coding is 4739.99 s. The loop-based coding method has the shortest calculation time, only 1195.81 s, which is 92.43% and 74.77% lower than the previous two methods, respectively. Through the loop-based coding method, the variable dimension and calculation time are greatly reduced. It provides a groundwork for the establishment of MADDPG action space. The neural network constructed by the first two coding methods is huge and difficult to converge, so the loop-based coding method is crucial.

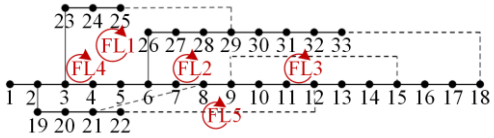


Fig. 1. FLs of IEEE 33-bus system.

TABLE I
FLs of IEEE 33-bus System

FLs	Branches
FL1	{b3, b4, b5, b22, b23, b24, b25, b26, b27, b28, b37}
FL2	{b6, b7, b8, b15, b16, b17, b25, b26, b27, b28, b29, b30, b31, b32, b34, b36}
FL3	{b9, b10, b11, b12, b13, b14, b34}
FL4	{b2, b3, b4, b5, b6, b7, b18, b19, b20, b33}
FL5	{b8, b9, b10, b11, b21, b33, b35}

TABLE II
CBs of the IEEE 33-bus System

CBs	Branches
CB12	{b25, b26, b27, b28}
CB14	{b3, b4, b5}
CB23	{b34}
CB24	{b6, b7}
CB25	{b8}
CB35	{b9, b10, b11}
CB45	{b33}

TABLE III
Number of Switch Combinations and Computing Time

Coding strategy	Number	Computing time (s)
Branch-based coding	435 897 (C_{37}^{5})	15 788.73
Branch-group-based coding	136 548	4739.99
Loop-based coding	86 240	1195.81

IV. Reinforcement Learning Modeling of Distribution Network Reconfiguration

A. Principle of Multi-agent Reinforcement Learning

RL is an algorithm with self-learning ability through the continuous interaction between agents and the environment. The agents aim to obtain the maximum cumulative expected discount return and find the optimal action policy. During the training stage, the agent firstly observes the environment and obtains the state $\boldsymbol{s}_t$. Next, the agent executes the action policy $\boldsymbol{a}_t$ according to its historical experience and the environment state $\boldsymbol{s}_t$ is transferred to a new state $\boldsymbol{s}_{t+1}$, and gets a reward $\boldsymbol{r}_t$ as feedback to the agents. Finally, the agent will update the action policy according to the received reward $\boldsymbol{r}_t$ to take optimal action policy.

Because of the dimensionality disaster and slow convergence speed of single-agent DRL, the MADRL is introduced. The framework of MADRL is shown in Fig. 2.

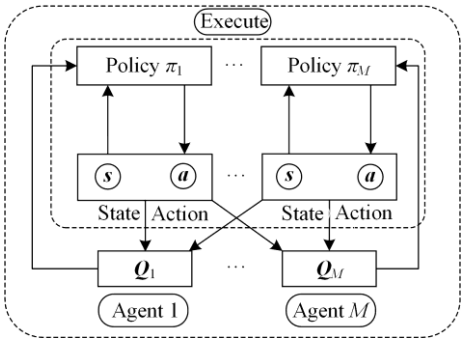


Fig. 2. Framework of MADRL.

An interactive process between agents and the environment in multi-agent DRL can be represented by $<M,\boldsymbol{s},(\boldsymbol{o}_1,\boldsymbol{o}_2,...,\boldsymbol{o}_M),(\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_M),(r_1,r_2,\cdots,r_M),\mathcal{T},\gamma>$ where M is the number of agents; $\boldsymbol{s}$ is the environment state; $\boldsymbol{o}_m:\boldsymbol{s}\to\boldsymbol{o}_m$ denotes the partial observation state of the mth agent; $\boldsymbol{a}_m$ represents the action policy of the mth agent; $r_m:\boldsymbol{s}\times\boldsymbol{a}_m\to\mathbb{R}$ represents the reward value of the mth agent; $\mathcal{T}:\boldsymbol{s}\times\boldsymbol{a}_1\times\boldsymbol{a}_2\times\cdots\boldsymbol{a}_M\to\boldsymbol{s}'$ is the state transition function; γ is the discount factor. The agents aim to maximize the cumulative expected discount return, which is as follows:

$$\max_{\pi_m} \underset{\boldsymbol{a}_{m,t}\sim\pi_m,\ \boldsymbol{s}_{t+1}\sim p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_{m,t})}{E}\left(\sum_{t=0}^{T}\gamma^t r_{m,t}\right) \tag{12}$$

where π_m is the mth agent's policy; $\boldsymbol{a}_{m,t}$ represents the action of the mth agent; $r_{m,t}$ represents the reward of the mth agent at time t; $\boldsymbol{s}_t$ is the environment state at time t; $p(\boldsymbol{s}_{t+1}\mid\boldsymbol{s}_t,\boldsymbol{a}_{m,t})$ represents the state transition probability of $\boldsymbol{s}_{t+1}$ after executing $\boldsymbol{a}_{m,t}$ at state $\boldsymbol{s}_t$.

In the context of DNR, each agent corresponds to a FL. The environment signifies the simulated distribution network. The agents will execute the action policy according to the state of the environment (line and bus information), and the state of the distribution network environment will be updated due to the actions. Then, a new state and a reward value will be fed back to the agents. At the same time, the agents will store the trajectory of interaction with the environment, evaluate the action policy, and guide the neural networks of the agents to update.

B. Partially Observable Markov Decision Processes for Active Distribution Network Reconfiguration

The partially observable Markov decision process (POMDP) is the basis of RL modeling. POMDP is an observation-to-action system. It represents the chances that an action may lead to different results, and estimates the specific actions that are most likely to lead to the best results. In the POMDP, decision makers take measures to get the best results. POMDP of DNR will be constructed as follows, including state space, action space and reward function.

1) State Space

In the POMDP of DNR, the mth agent state space denotes the state of the distribution network of the mth loop, including load, DG output, bus voltage and branch state. The state space of the mth agent at time t can be illustrated as:

$$\boldsymbol{o}_{m,t}=\{\boldsymbol{P}_{m,t}^{\text{load}},\boldsymbol{Q}_{m,t}^{\text{load}},\boldsymbol{P}_{m,t}^{\text{DG}},\boldsymbol{Q}_{m,t}^{\text{DG}},\boldsymbol{V}_{m,t},\boldsymbol{B}_{m,t}\} \tag{13}$$

where $\boldsymbol{P}_{m,t}^{\text{load}}$ and $\boldsymbol{Q}_{m,t}^{\text{load}}$ represent the vector of the active load and the reactive load of every bus in the mth loop at time t, respectively; $\boldsymbol{P}_{m,t}^{\text{DG}}$ and $\boldsymbol{Q}_{m,t}^{\text{DG}}$ represent the vector of the active power and reactive power of every DG in the mth loop at time t; $\boldsymbol{V}_{m,t}$ represents the vector of bus voltage in the mth loop at time t; $\boldsymbol{B}_{m,t}$ is the vector of the status of every branch in the mth loop at time t.

2) Action Space

Because multi-agent DRL is different from the heuristic algorithm, the coding method is improved according to Section III. $\boldsymbol{a}_{m,t}$ is the action policy of the mth agent at time t, where $\boldsymbol{a}_{m,t}$ is a vector of $n_m\times1$ and n_m is the number of switches in the mth loop.

Therefore, the agents' joint action $\boldsymbol{a}_t$ can be obtained by combining all the agent actions together, which is expressed as follows:

$$\boldsymbol{a}_t=(\boldsymbol{a}_{1,t},\boldsymbol{a}_{2,t},\cdots,\boldsymbol{a}_{M,t}) \tag{14}$$

where M represents the number of agents, and satisfies $M=L$, i.e., the number of fundamental loops in the system.

3) Reward Function

The objective of the DNR model formulated in (1) is to minimize the active power loss and voltage offset. To achieve the objective, the objective function needs to be transformed into reward functions. Simultaneously, it is imperative to ensure that the DNR solution satisfies the power flow equation constraints and radial constraints. Since $P_{\text{loss},t}$ and ΔV_t represent the active power loss and voltage offset before DNR, which are constants at any

time t, so deleting $P_{\text{loss},t}$ and ΔV_t from the objective function will not affect the agents' exploration of the agent policy. Consequently, the reward functions can be transformed into as follows:

$$r_{m,t,1} = -P'_{\text{loss},t} / P_{\text{loss},t} \tag{15}$$

where $r_{m,t,1}$ is the opposite of active loss after reconfiguration of the mth agent at time t.

$$r_{m,t,2} = -\Delta V'_t = -\sum_{i=1}^{N} \left| \frac{V_{i,\text{rate}} - V_{i,t}'}{V_{i,\text{rate}}} \right| \tag{16}$$

where $r_{m,t,2}$ is the opposite of the voltage offset after reconfiguration of the mth agent at time t.

Therefore, the reward for the mth agent can be formulated as follows:

$$r_{m,t} = \begin{cases} r_{m,t,1} + r_{m,t,2}, & \text{Meet the constraints} \\ -10, & \text{Do not meet the constraints} \end{cases} \tag{17}$$

C. Multi-agent Reinforcement Learning Algorithm for Distribution Network Reconfiguration

To address the problem of active DNR problem with multi-uncertainties, this paper proposes a MADDPG-based active distribution network dynamic reconfiguration approach, which is a combination of Q-learning and policy gradient. In the DNR problem, each actor network will output the action policy according to the state of the network, that is, opening or closing of the branches in the loop, while the critic will evaluate the reconfiguration scheme of the actor and guide the actor network update policy. By updating the parameters of the neural networks, the optimal policy can be found. In addition, to address the problem of overestimation during the training process, MADDPG employs double networks, namely the current network and the target network. As shown in Fig. 3, the current network is utilized for action selection, while the target network is employed to calculate the target Q-values.

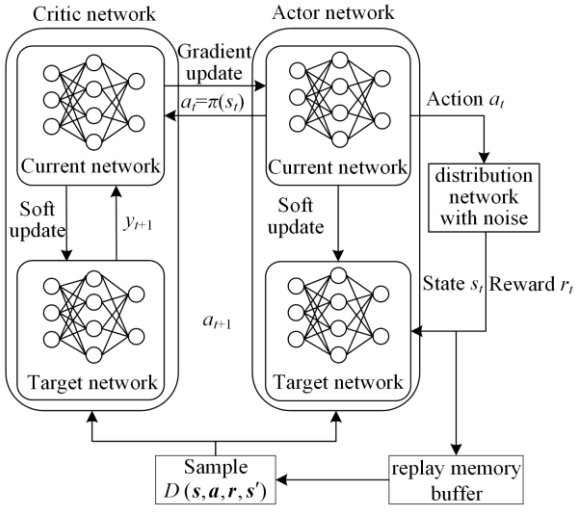


Fig. 3. Neural network structure of MADDPG.

During the training stage, the policy gradient of the ith actor network is represented as follows:

$$\nabla_{\theta_m} \boldsymbol{J}(\boldsymbol{\theta}_m) = E[\nabla_{\theta_m} \log \pi_m(\boldsymbol{a}_{m,t} \mid \boldsymbol{o}_{m,t}) \boldsymbol{Q}_m(\boldsymbol{s}_t, \boldsymbol{a}_t)] \tag{18}$$

where $\boldsymbol{Q}_m(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is the state-action value function of the mth critic network; ∇_{θ_m} represents the operation of obtaining the gradient of parameter θ_m.

For critic networks, the parameters can be updated by the following Berman error:

$$L(\boldsymbol{\theta}_m) = E[(\boldsymbol{Q}_m(\boldsymbol{s}_t, \boldsymbol{a}_t) - y)^2] \tag{19}$$

$$y = r_m + \gamma \boldsymbol{Q}_m\left[\boldsymbol{s}_{t+1}, \boldsymbol{a}_{m,t+1} \mid \boldsymbol{a}_{m,t+1} = \boldsymbol{u}'_m(\boldsymbol{s}_{t+1})\right] \tag{20}$$

where $\boldsymbol{u}'$ is the parameter of the target network.

The pseudo code of the MADDPG-based active DNR approach is shown as Table IV.

TABLE IV
MADDPG-BASED ACTIVE DISTRIBUTION NETWORK DYNAMIC RECONFIGURATION FOR M AGENTS

MADDPG-based distribution network reconfiguration for M agents

Initialize MADDPG network parameters
for e = 1 to E do:
  for t =1 to T do:
  Receive initial state $\boldsymbol{o}_{m,t} = \{\boldsymbol{P}_{m,t}^{\text{load}}, \boldsymbol{Q}_{m,t}^{\text{load}}, \boldsymbol{P}_{m,t}^{\text{DG}}, \boldsymbol{Q}_{m,t}^{\text{DG}}, \boldsymbol{V}_{m,t}, \boldsymbol{B}_{m,t}\}$
    for agent m = 1 to M do:
      Agent m selects an action $\boldsymbol{a}_{m,t} = \boldsymbol{u}_{\theta_m}(\boldsymbol{o}_{m,t})$
      Execute action $\boldsymbol{a}_{m,t}$ in the environment
      Update the distribution network state and observe the reward $r_{m,t}$
      if m=M do:
        Update load rate, DG output, and let $\boldsymbol{o}_{1,t+1} \leftarrow \boldsymbol{o}_{M,t}$
        Store $(\boldsymbol{o}_{M,t}, \boldsymbol{a}_{m,t}, r_{m,t}, \boldsymbol{o}_{1,t+1})$ in memory buffer D
      else do:
        $\boldsymbol{o}_{m+1,t} \leftarrow \boldsymbol{o}_{m,t}$
        Store $(\boldsymbol{o}_{m,t}, a_t^m, r_t^m, \boldsymbol{o}_{m+1,t})$ in memory buffer D
    end for
    for agent m = 1 to M do:
      Sample a random minibatch sample from D
      Update critic by minimizing the loss
      $$L(\boldsymbol{\theta}_m) = E[(Q_m(\boldsymbol{s}_t, \boldsymbol{a}_t) - y)^2]$$
      Update actor using sampled policy gradient:
      $$\nabla_{\theta_m} \boldsymbol{J}(\boldsymbol{\theta}_m) = E[\nabla_{\theta_m} \log \pi_m(\boldsymbol{a}_{m,t} \mid \boldsymbol{o}_{m,t}) Q_m(\boldsymbol{s}_t, \boldsymbol{a}_t)]$$
    end for
    Update target networks for each agent m:
    $$\boldsymbol{\theta}_m' = r\boldsymbol{\theta}_m + (1-r)\boldsymbol{\theta}_m'$$
    $$\boldsymbol{u}_m' = r\boldsymbol{u}_m + (1-r)\boldsymbol{u}_m'$$
  end for
end for

V. CASE STUDY

A. Environment and Parameter Setting

In the case study, an improved IEEE 33-bus power system is adopted, as displayed in Fig. 4. It contains 1 substation and 37 branches, and the operating voltage ranges from 0.90 to 1.05. Branch and bus information can be found in [21]. Based on the IEEE 33-bus system, wind turbines with rated power of 120, 220 and 200 kW are integrated at buses 10, 18 and 21, respectively, with a power factor of 0.9. The 24-hour power curves of the load and DGs with 5% fluctuation are illustrated in

Figs. 5 and 6, which come from the Xihe energy big data platform [22]. The FLs and branches with RCS are shown in Fig. 7 and Table V.

All DRL methods employed in this paper run on the Windows 11 operating system, and the framework of DRL is Pytorch 11.3. The computer hardware configuration is i7-12500h CPU@2.50 GHz, NVIDIA GeForce RTX 2050 4 GB GPU and 16 GB memory.

The non-DRL algorithms for comparison, namely GA and mixed-integer second-order cone programming (MISOCP), run on Matlab 2021b, and GA algorithm toolbox and Yalmip+Gurobi are used to solve the model respectively.

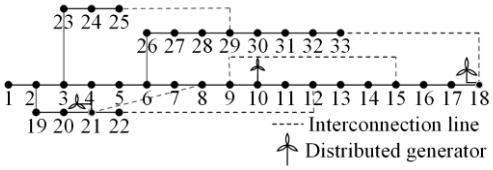


Fig. 4. Improved IEEE 33-bus power system.

TABLE V
BRANCHES WITH RCS OF IMPROVED IEEE 33-BUS POWER SYSTEM

FLs	RCS
FL1	{b3, b23, b27, b37}
FL2	{b7, b8, b27, b31, b34, b36}
FL3	{b9, b13, b34}
FL4	{b3, b7, b18, b33}
FL5	{b8, b9, b33, b35}

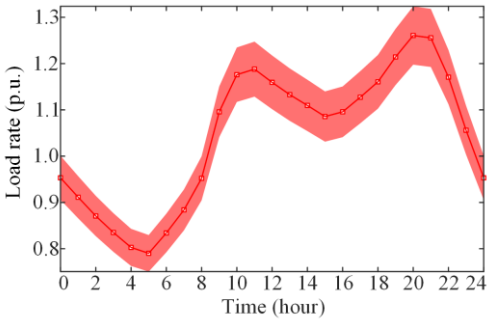


Fig. 5. Typical daily load rate of power system.

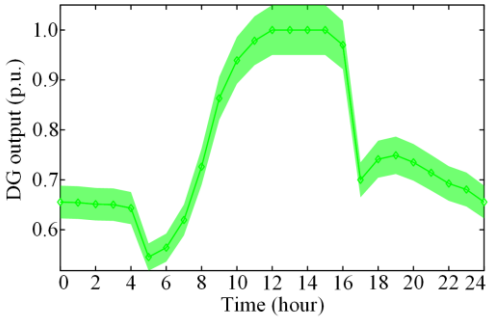


Fig. 6. Typical daily DG output in the power system.

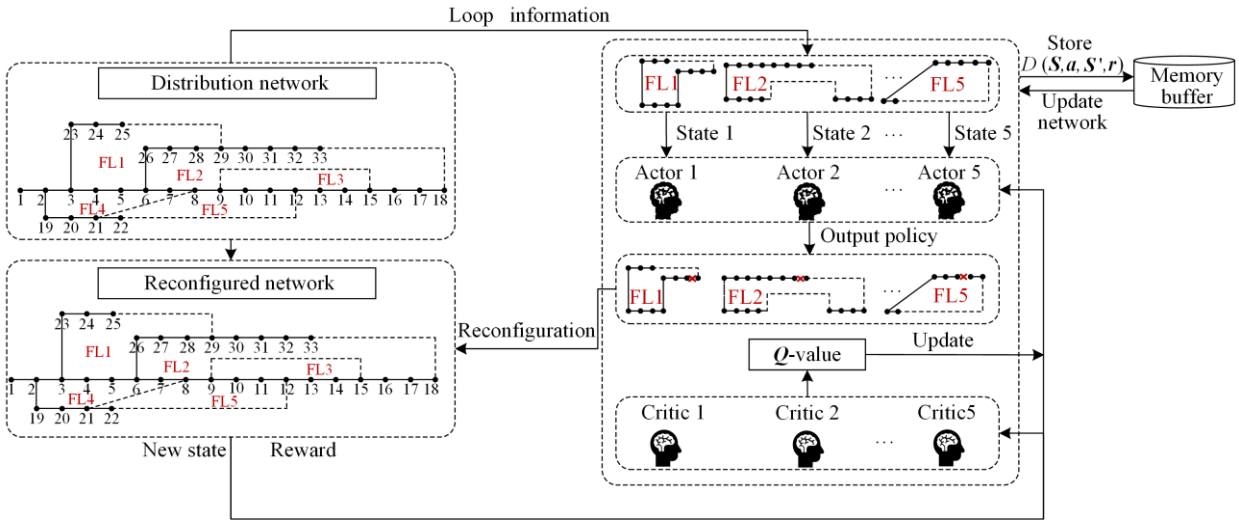


Fig. 7. Structure diagram of DNR based on MADDPG.

There are many hyperparameters involved in the MADDPG algorithm. The parameters of MADDPG and neural networks are set in Tables VI and VII.

TABLE VI
HYPERPARAMETERS OF MADDPG

Hyperparameter	value
learning rate α	0.00001
discount factor γ	0.95
training batch size	64
memory capacity	10000
max episode	1000
random exploration episode	300

TABLE VII
ACTOR AND CRITIC NETWORK STRUCTURES OF MADDPG

Layer	Actor network	Critic network
1	$\lvert o_{m,t}\rvert$ /512 (ReLu)	$\lvert s_t\rvert$ /256 (ReLu)
2	512/64 (ReLu)	(256+ M_l) /128 (ReLu)
3	64/M_l (softmax)	128/64 (ReLu)
4		64/1 (ReLu)

B. MADDPG Hyperparameters Determination

In order to evaluate the impact of different hyperparameters, it is necessary to adjust different learning rates and discount factors of MADDPG. Figures 8–10 plot the training process of the MADDPG network with

different learning rates. It can be found from Fig. 8 that the reward of the three networks all exhibit an initial decrease followed by an increase during training. The larger the learning rate, the faster the convergence of MADDPG. The convergence speed of MADDPG network with the learning rate 1×10^{-4} and 1×10^{-3} is similar, both of which are faster than that with the learning rate 1×10^{-5}. However, both of them have a downward trend after the reward reaches the maximum, and the trend of network learning rate 1×10^{-3} is more significant, which is because the network is over-fitted due to the high learning rate, making the training effect decline. Therefore, it can be concluded that the network with a learning rate of 1×10^{-5} is the most stable. Besides, when the learning rate is 1×10^{-5}, the reward is the highest, which is 5.19% and 32.97% higher than that when the learning rate is 1×10^{-4} and 1×10^{-3}, respectively. It can be seen from Figs. 9 and 10, the convergence speed of loss value of actor and critic with learning rate of 1×10^{-5} is slower than that of the other two networks, but the converged loss value is the smallest, and the training effect is better than that of the other two networks. Thus, the learning rate is set as 1×10^{-5}.

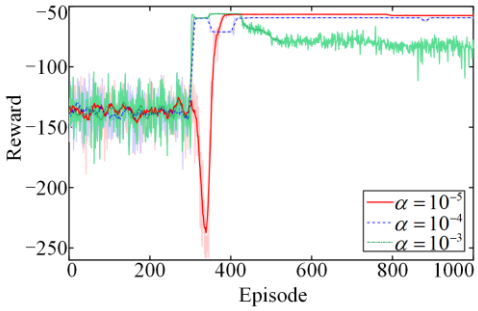


Fig. 8. The reward of MADDPG with different learning rates.

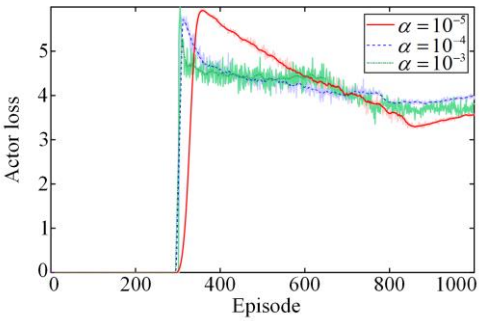


Fig. 9. The actor loss of MADDPG with different learning rates.

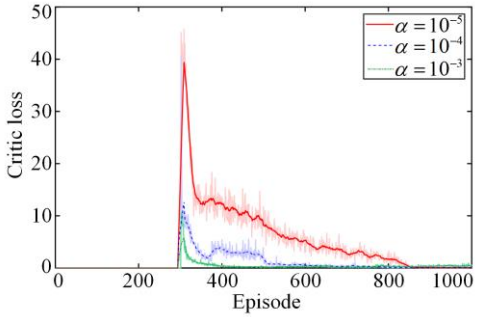


Fig. 10. The critic loss of MADDPG with different learning rates.

Figures 11–13 plot the training process of the MADDPG network with different discount factors. As shown in Fig. 11, different discount factors have little influence on the reward, and the neural network with a discount factor of 0.95 converges slightly faster than the other two networks. Similarly, from Figs. 12 and 13, both actor and critic loss exhibit an initial increase followed by a subsequent decrease in the training stage, and the loss value is the fewest when the discount factor is 0.95, so the discount factor is selected as 0.95.

To sum up, the learning rate of the MADDPG network proposed in this paper is set as 1×10^{-5}, and the discount factor is 0.95.

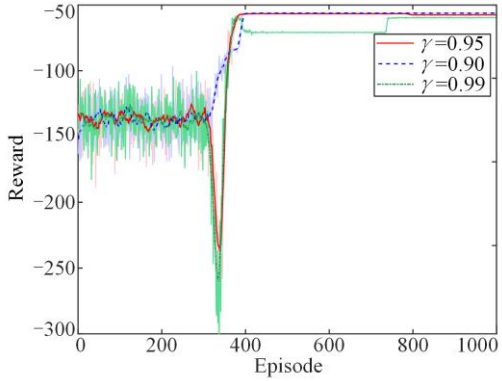


Fig. 11. The reward of MADDPG with different discount factors.

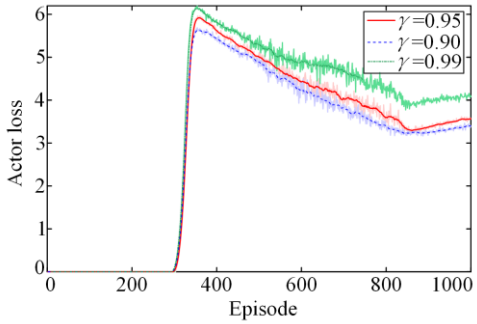


Fig. 12. The actor loss of MADDPG with different discount factors.

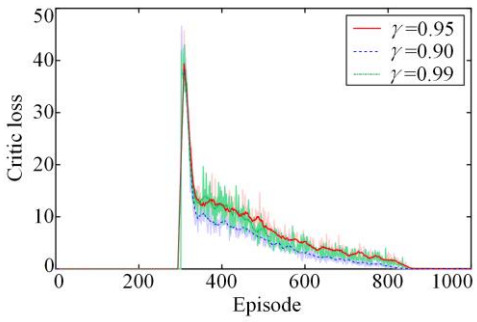


Fig. 13. The critic loss of MADDPG with different discount factors.

C. Simulations and Analysis

Figures 14–16 plot the training results under different algorithms. As shown in Fig. 14, the first 300 episodes are in the exploring stage, and the rewards of DDQN and MADDPG fluctuate due to random exploration. After the number of episodes reaches 300, the agents

begin to train, and the reward of the DDQN network is still hard to converge. However, MADDPG agents cooperate to control the topology of the distribution network, and attain good results. When the number of episodes reaches 400, the reward value reaches the maximum. The gray dotted and black dotted line at the top of Fig. 14 are the objective values of GA and MISOCP, respectively. After the algorithm is stable, MISOCP has the best effect and the highest reward value, but the calculation time is the longest, and MADDPG has the second-highest reward and the shortest calculation time. The reward of GA is slightly lower than MADDPG, and the calculation time is longer. Convergence of DDQN is challenging, and it has the lowest average reward value. Figures 15 and 16 show the loss of actors and critics, where the first 300 episodes are exploring and have not yet started training. After 300 episodes, actor and critic losses first increase and then decrease, the reason for the initial increase is that the network deviation is large, and consequently the loss value is high. After training, the Q value of critic networks is close to the true value, and actor networks also converge to the optimal policy, so actor loss and critic loss begin to decline and gradually converge.

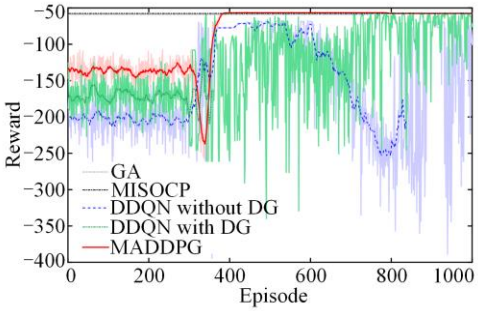


Fig. 14. The reward of different algorithms.

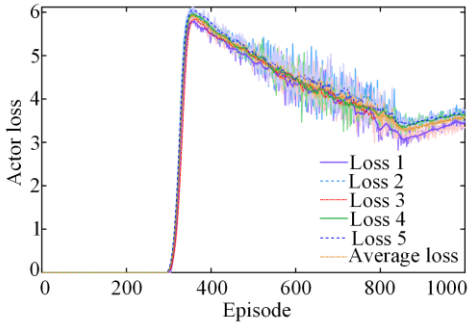


Fig. 15. The actor loss of MADDPG agents.

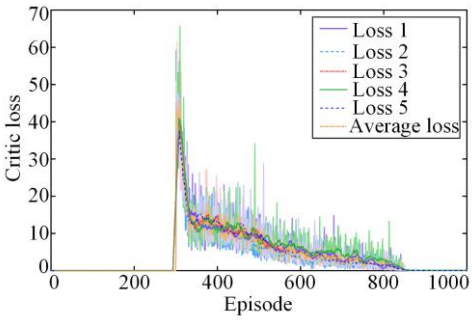


Fig. 16. The critic loss of MADDPG agents.

D. Effectiveness Analysis of MADDPG

To demonstrate the superiority of MADDPG, other three algorithms, i.e., DDQN, GA and MISOCP are utilized to compare the active loss and bus voltage of 24-hour power grid [23]–[25].

It can be found from Fig. 17, before DGs are integrated into the network, the daily average network loss and voltage offset are significant, which are 236.21 kW and 1.74, respectively. After DGs are connected, the daily average network loss and voltage offset are 188.46 kW and 1.48 respectively, which are 20.21% and 14.94% lower than those before DGs connected to the distributed network. This demonstrates that the moderate integration of DG into the power system can decrease network loss and voltage offset. The daily average network loss and voltage offset of DDQN are 143.59 kW and 1.13, respectively. Compared with the scheme of simply connecting DG without DNR, the reconfiguration scheme of DDQN reduces the daily average active power loss and the daily average voltage offset by 23.81% and 23.65%, respectively. However, those are still inferior to MADDPG. Under the proposed MADDPG-based active DNR approach, the daily average network loss and voltage offset are 140.90 kW and 0.97. Compared with those of DDQN, the proposed approach can greatly reduce the network loss and voltage offset by 25.24% and 34.46%. These demonstrate that in contrast to DDQN, MADDPG agents can better adapt to the uncertainty of load and DG, and possess better generalization capability. Besides, the daily average loss of GA is lower than that of MADDPG, which is 136.03 kW, but the voltage offset is higher than that of MADDPG, which is 1.10, so the reward of GA is lower than that of MADDPG. As a mathematical optimization algorithm, MISOCP has the best result, the network loss and voltage offset are 137.38 kW and 1.03, respectively. However, the MISOCP algorithm has the longest computation time and cannot achieve real-time optimization of DNR. As the size of the distribution network grows, solving the DNR model with MISOCP becomes increasingly challenging. Additionally, the results obtained by MISOCP are ideal solutions, because they only be applied when the stochastic information of DG and load of the power system are known in advance. The uncertainties in load and DG output make it difficult to apply MISOCP in practical distribution network scenarios. Therefore, MISOCP is only suitable for theoretical research as a reference experiment.

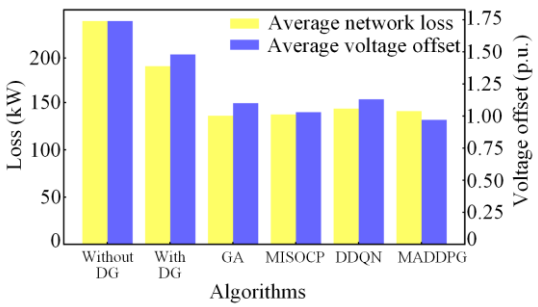


Fig. 17. Daily network loss and voltage offset of different algorithms.

Figures 18 and 19 display the voltage curve of different algorithms. It can be found from Fig. 18 that the result of MADDPG is close to that of MISOCP, the voltages of MADDPG at most buses are higher than that of DDQN and GA, and the voltage distribution is more balanced. This demonstrates that solving the DNR with the MADDPG can more effectively ensure the safety and economy of power systems.

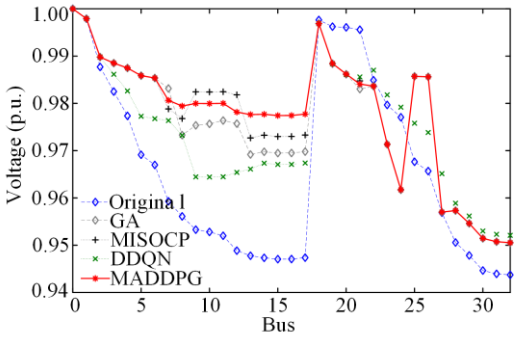
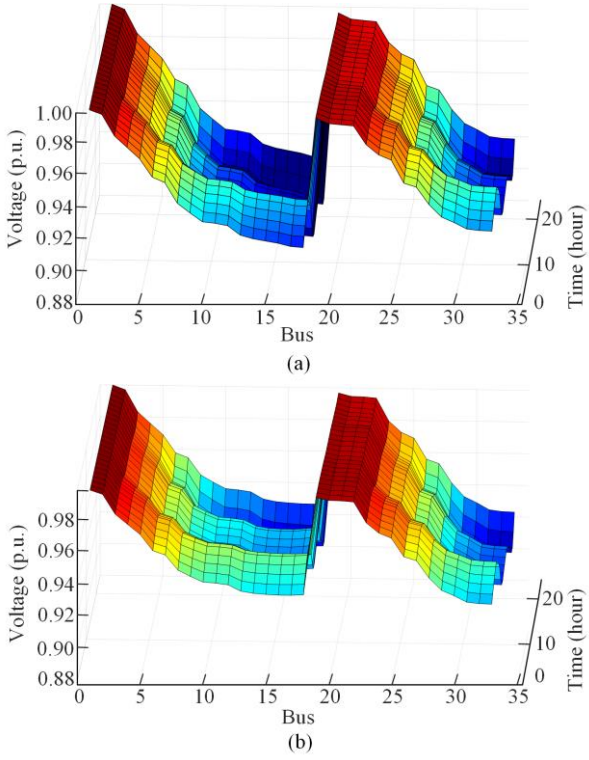


Fig. 18. Bus voltage of different algorithms at 06: 00.

Figure 19 displays the bus voltage of the distribution network at different times with different algorithms. From Fig. 19 (a) and (b), it can be found that DG can effectively improve the bus voltage. For example, the average voltage of the bus 18 has increased from 0.9060 to 0.9304. Similarly, from Figs. 19 (c), (e) and (f), it can also be found that DDQN, GA and MADDPG algorithms have also improved the bus voltage of the network, such as bus 18. The average voltage of bus 18 of the three algorithms is 0.9578, 0.9409 and 0.9713, and the improvement effect of MADDPG is the most obvious. Consequently, the voltage of MADDPG is the most stable.



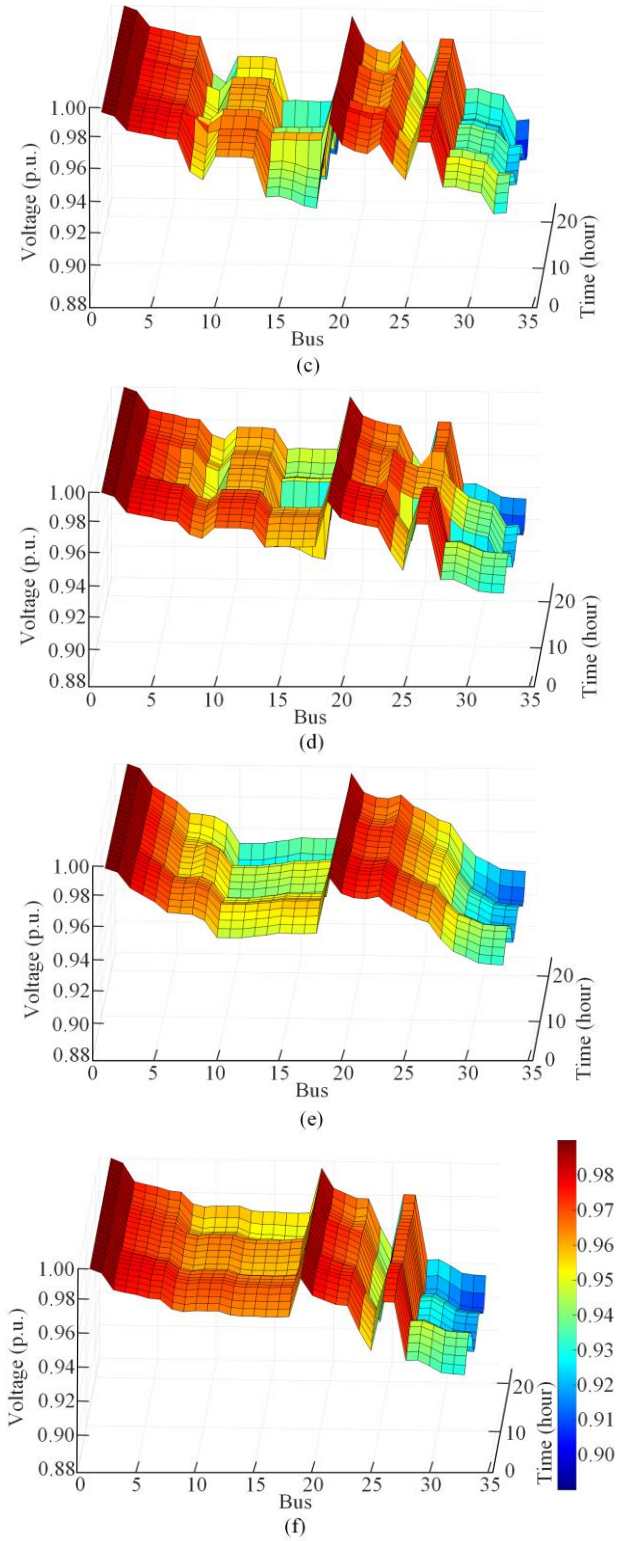


Fig. 19. Voltage in 24 hour under different algorithms. (a) Bus voltage of original network without DG. (b) Bus voltage of original network with DG. (c) Bus voltage of GA. (d) Bus voltage of MISOCP. (e) Bus voltage of DDQN. (f) Bus voltage of MADDPG.

In brief, DG access can improve the voltage of the power system, and DNR schemes provided by DDQN, GA, MISOCP and MADDPG can further improve the

voltage of the network. MADDPG exhibits superior adaptability to non-stationary environments than DDQN. Therefore, the optimization results of MADDPG are superior to DDQN. Compared to MADDPG, the heuristic algorithm GA is sensitive to parameter selection and initial conditions, leading to poor stability and a tendency to get stuck in local optima. Moreover, when dealing with high-dimensional model, GA may incur high computational costs. This implies that more computational resources and time are needed for large-scale problems. While MISOCP, as a mathematical optimization algorithm, can theoretically obtain global optimal solutions, it faces challenges when dealing with high-dimensional mathematical problems and may encounter NP-hard problems or even fail to find solutions. Additionally, MISOCP requires knowing the stochastic variables in advance, making it unable to consider the uncertainty in distribution networks. Thus, MISOCP is mostly applicable to theoretical research and encounters challenges when attempting to implement it in actual engineering. Consequently, among the mentioned algorithms, MADDPG is the most appropriate for solving the practical problem of dynamic DNR with multiple stochastic factors.

E. Numerical Comparison Among Different Algorithms

To quantitatively demonstrate the superiority of MADDPG, the current benchmarks for comparing the performance of different algorithms on DNR tasks include DDQN, GA, and MISOCP. Table VIII lists the reward, training time and decision-making time of different methods.

From Table VIII, it can be observed that:

1) The reward value of MISOCP is the highest, which is −57.5062, and that of MADDPG is −57.5349, which is 4.2218 and 1.4654 higher than that of DDQN and GA, respectively. This is because MADDPG can effectively use the bus and line information of the distribution network, so that it can find a network topology structure more appropriate for the current state. As for GA, due to the complexity and high dimension of the distribution network structure, it is not conducive to the optimization speed and accuracy of GA, resulting in a slightly inferior performance than that of a trained MADDPG.

2) In terms of computational speed, the training times for MADDPG and DDQN are 0.89 hour and 2.08 hour. The training time of MADDPG is reduced by 57.21% compared with DDQN. The decision-making times for MADDPG, DDQN, GA, and MISOCP are 0.10 s, 0.12 s, 28.34 s and 34.94 s, respectively. The decision-making time of MADDPG is reduced by 20%, 99.65% and 99.71% compared with DDQN, GA and MISOCP, respectively. Traditional heuristic methods and mathematical optimization algorithms require optimization based on the precise parameters of the power grid, which is time-consuming. On the other hand, although data-driven approaches require offline training, they do not rely on the power grid model parameters during online operation. These methods utilize a neural network to output the policy based on learned experience, resulting in significantly faster performance compared to other algorithms.

TABLE VIII
COMPARISON OF EVALUATION INDEXES OF DIFFERENT ALGORITHMS

Algorithm	Reward	Training time (hour)	Decision-making time (s)
GA	−59.0003		28.34
MISOCP	−57.5062		34.94
DDQN	−61.7567	2.08	0.12
MADDPG	57.5349	0.89	0.10

Therefore, for solving the DNR model, MADDPG not only effectively minimizes the network active loss and voltage offset, but also significantly reduces the calculation time, demonstrating the remarkable superiority of MADDPG.

VI. CONCLUSION

To address the dynamic DNR model with multiple uncertainties, an approach for DNR based on MADDPG is proposed. It can adapt to the uncertainty of DGs and loads in the power system and provide the optimal DNR scheme. The conclusions are the following.

1) A loop-based coding method is adopted, which greatly reduces the solution space of DNR. Compared with the branch-based and branch group-based coding methods, the proposed coding method reduces the calculation time by 92.43% and 74.77%, respectively.

2) The MADDPG-based active DNR approach proposed in this paper can significantly reduce the network loss and improve the voltage offset.

3) Compared to other DRL methods, the proposed approach is more adaptable to the non-stationary distribution network environment when solving the DNR problem. It can converge neural network parameters faster and achieve higher reward.

4) The proposed approach demonstrates the ability to adapt to the uncertainty of DG and load, and has lower computational costs. The decision-making time of the proposed approach is reduced by 20%, 99.65% and 99.71% compared with that of DDQN, GA and MISOCP, respectively.

In the future, our work will focus on constructing more complex DNR models and developing better DRL algorithms to achieve more flexible control and better results of DNR. For instance, the graph structure information of the distribution network can be fully taken into account in the DRL algorithm to obtain better DNR schemes.

ACKNOWLEDGMENT

Not applicable.

AUTHORS' CONTRIBUTIONS

Changxu Jiang and Zheng Lin: writing original draft, methodology, software, investigation, formal analysis, writing review and editing. Chenxi Liu and Feixiong Chen: writing review and editing. Zhenguo Shao: conceptualization, methodology, and funding acquisition. All authors read and approved the final manuscript.

FUNDING

This work is supported by the Natural Science Foundation of Fujian Province (No. 2022J0512 and No. 2021J05134), and the National Natural Science Foundation of China (No. 52377087).

AVAILABILITY OF DATA AND MATERIALS

Please contact the corresponding author for data material request.

DECLARATIONS

Competing interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

AUTHORS' INFORMATION

Changxu Jiang received the Ph.D. degree in electric power system and automation from the School of Electric Power Engineering, South China University of Technology, Guangzhou, China, in 2020. Since 2020, he works at the School of Electrical Engineering and Automation, Fuzhou University. His research interests include electric vehicle, deep reinforcement learning, smart grid, optimal power system/microgrid scheduling and decision making, stochastic optimization considering large-scale integration of renewable energy into the power system.

Zheng Lin is currently studying at the School of Electrical Engineering and Automation, Fuzhou University, Fuzhou, Fujian province, China. His research interests include smart grid, deep reinforcement learning, and graph neural network.

Chenxi Liu is currently studying at the School of Electrical Engineering and Automation, Fuzhou University, Fuzhou, Fujian province, China. His research interests include smart grid, deep reinforcement learning, and power system operation optimization.

Feixiong Chen works at the School of Electrical Engineering and Automation, Fuzhou University, Fuzhou, Fujian province, China. His research interest includes collaborative optimal control of integrated energy system and analysis and mining of power big data.

Zhenguo Shao works at the School of Electrical Engineering and Automation, Fuzhou University, Fuzhou, Fujian province, China. His research interest includes power grid uncertainty theory and method, power quality analysis and management, stable operation and control of power system, power big data theory and method.

REFERENCES

[1] V. Gurugubelli, A. Ghosh, and A. K. Panda, "Parallel inverter control using different conventional control methods and an improved virtual oscillator control method in a standalone microgrid," *Protection and Control of Modern Power Systems*, vol. 7, no. 3, pp.1-13, Jul. 2022.

[2] J. Zheng, W. Xiao, and C.Wu *et al*., "A gradient descent direction based-cumulants method for probabilistic energy flow analysis of individual-based integrated energy systems," *Energy*, vol. 265, pp. 1-13, Feb. 2023.

[3] H. Yang, X. Li, and Z. Cao *et al*., "A two-stage dynamic reconfiguration method for distribution networks considering wind and solar power," *Power System Protection and Control*, vol. 51, no. 21, pp. 12-21, Nov. 2023. (in Chinese)

[4] D. Xiao, Z. Lin, and H. Chen *et al*., "Windfall profit-aware stochastic scheduling strategy for industrial virtual power plant with integrated risk-seeking/averse preferences," *Applied Energy*, vol. 357, pp. 1-13, Mar. 2024.

[5] Z. Yi , Z. Chen, and K. Yin *et al*., "Sensing as the key to the safety and sustainability of new energy storage devices," *Protection and Control of Modern Power Systems*, vol. 8, no. 2, pp. 1-22, Apr. 2023.

[6] M. Mahdavi, H. H. Alhelou, and N. D. Hatziargyriou *et al*., "Reconfiguration of electric power distribution systems: comprehensive review and classification," *IEEE Access*, vol. 9, pp. 118502-118527, Aug. 2024.

[7] Z. Guo, Z. Zhou, and Y. Zhou *et al*., "Impacts of integrating topology reconfiguration and vehicle-to-grid technologies on distribution system operation," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 2, pp. 1023-1032, Apr. 2020.

[8] M. R. Dorostkar-Ghamsari, M. Fotuhi-Firuzabad, and M. Lehtonen *et al*., "Value of distribution network reconfiguration in presence of renewable energy resources," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 1879-1888, May 2016.

[9] Y. Fu and H. Chiang, "Toward optimal multiperiod network reconfiguration for increasing the hosting capacity of distribution networks," *IEEE Transactions on Power Delivery*, vol. 33, no. 5, pp. 2294-2304, Oct. 2018.

[10] M. Abdelaziz, "Distribution network reconfiguration using a genetic algorithm with varying population size," *Electric Power Systems Research*, vol. 142, pp. 9-11, Jan. 2017.

[11] Y. Liu, L. Wang, and D. Li *et al*., "State-of-health estimation of lithium-ion batteries based on electrochemical impedance spectroscopy: a review," *Protection and Control of Modern Power Systems*, vol. 8, no. 3, pp. 1-17, Jul. 2023.

[12] W. Huang, W. Zheng, and D.J. Hill, "Distribution network reconfiguration for short-term voltage stability enhancement: an efficient deep learning approach," *IEEE Transactions on Smart Grid*, vol.12, no. 6, pp. 5385-5395, Nov. 2021.

[13] H. Song, Y. Liu, and J. Zhao, "Prioritized replay dueling DDQN based grid-edge control of community energy storage system," *IEEE Transactions on Smart Grid*, vol. 12, no. 6, pp. 4950-4961, Nov. 2021.

[14] W. Lei, H. Wen, and J. Wu, "MADDPG-based security situational awareness for smart grid with intelligent edge," *Applied Sciences*, vol. 11, no. 7, Mar. 2021.

[15] H. Xu, Z. Yu, and Q. Zheng *et al*., "Deep reinforcement learning-based tie-line power adjustment method for power system operation state calculation," *IEEE Access*, vol. 7, pp. 156160-156174, Oct. 2019.

[16] B. Wang, H. Zhu, and H. Xu *et al*., "Distribution network reconfiguration based on noisy net deep Q-learning network," *IEEE Access*, vol. 9, pp. 90358-90365, Jun. 2021.

[17] D. Cao, W. Hu, and X. Xu *et al*., "Deep reinforcement learning based approach for optimal power flow of distribution networks embedded with renewable energy and storage devices," J*ournal of Modern Power Systems and Clean Energy*, vol. 9, no. 5, pp. 1101-1110, Sept. 2021.

[18] R. Shen, S. Zhong, and X. Wen *et al*., "Multi-agent deep reinforcement learning optimization framework for building energy system with renewable energy," *Applied Energy*, vol. 312, Apr. 2022.

[19] J. Zheng, Z. Liang, and Y. Li *et al*., "Multi-agent reinforcement learning with privacy preservation for continuous double auction based p2p energy trading," *IEEE Transactions on Industrial Informatics,* vol. 20, no. 4, Apr. 2024.

[20] R. Lowe, Y. Wu, and A. Tamar *et al*., "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems*, Long Beach, USA, Dec. 2017, pp. 6379-6390.

[21] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Transactions on Power Delivery*, vol. 4, no. 2, pp. 1401-1407, Apr. 1989.

[22] *Xihe Energy Big Data Platform*, [Online]. Available: https://xihe-energy.com/

[23] N. Gholizadeh, N. Kazemi, and P. Musilek, "A comparative study of reinforcement learning algorithms for distribution network reconfiguration with deep q-learning-based action sampling," *IEEE Access*, vol. 11, pp. 13714-13723, Feb. 2023.

[24] P. Dziwiński and Ł. Bartczuk, "A new hybrid particle swarm optimization and genetic algorithm method controlled by fuzzy logic," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 6, pp. 1140-1154, Jun. 2020.

[25] J. C. López, M. Lavorato, and M. J. Rider, "Optimal reconfiguration of electrical distribution systems considering reliability indices improvement," *International Journal of Electrical Power & Energy Systems*, vol. 78, pp. 837-845, Jun. 2016.