Deep Reinforcement Learning-Based Model-Free On-Line Dynamic Multi-Microgrid Formation to Enhance Resilience

Jin Zhao, *Member, IEEE*, Fangxing Li, *Fellow, IEEE*, Srijib Mukherjee, *Senior Member, IEEE*, and Christopher Sticht, *Senior Member, IEEE*

***Abstract*—Multi-microgrid formation (MMGF) is a promising solution for enhancing power system resilience. This paper proposes a new deep reinforcement learning (RL) based model-free on-line dynamic MMGF scheme. The dynamic MMGF problem is formulated as a Markov decision process, and a complete deep RL framework is specially designed for the topology-transformable micro-grids. In order to reduce the large action space caused by flexible switch operations, a topology transformation method is proposed and an action-decoupling Q-value is applied. Then, a convolutional neural network (CNN) based multi-buffer double deep Q-network (CM-DDQN) is developed to further improve the learning ability of the original DQN method. The proposed deep RL method provides real-time computing to support the on-line dynamic MMGF scheme, and the scheme handles a long-term resilience enhancement problem using an adaptive on-line MMGF to defend changeable conditions. The effectiveness of the proposed method is validated using a 7-bus system and the IEEE 123-bus system. The results show strong learning ability, timely response for varying system conditions and convincing resilience enhancement.**

***Index Terms*—Convolutional neural network (CNN), Deep reinforcement learning (DRL), extreme weather, distributed generation (DG), microgrids (MGs), multi-microgrid formation (MMGF), power system resilience.**

I. Introduction

HIGH-IMPACT and low-probability events, such as extreme weather events, are occurring with increasing intensity. The extensive damage and subsequent outages of a power system caused by extreme events indicates the necessity of enhancing power system resilience [1]. Microgrids (MGs), which improve the flexibility of power system operation with both grid-connecting and islanding modes, are promising solutions for power grids to withstand unplanned catastrophic events [2]. With the wide penetration of distributed energy resources (DERs), advanced metering, communication, and automatic control infrastructures, the distribution system (DS) can be easily transformed into self-supported MGs [3]. These self-supported MGs largely benefit power system resilience by improving the restoration capability of the distribution networks [4]-[6] as well as the survival of critical loads [7].

Since the ability of MGs to handle extreme conditions has been proven both by academic index [8] and by practical cases (e.g., islanded MGs successfully survived Hurricane Sandy [9]), resilience-oriented MG formation, resources allocation, and system operation have been widely discussed. Optimal MG formation strategies were proposed in [4] and [10] to divide original DSs into resilient MGs after major faults of the main grid. Allocable distributed generations (DGs) [11] and remotely controlled switches (RCSs) [12] have been highlighted to provide a good planning study of resilient MGs. A transformative architecture for the normal operation and self-healing of multi-MGs was proposed in [13] to improve system self-healing capability, and [14] used a scheduling-horizon-based optimization scheme to reduce load shedding with reasonable operation cost. In addition, helpful resilient control strategies to benefit the operation of islanded MGs were studied in [15].

To fully utilize DERs to enhance power grid resilience, forming multiple MGs using DGs has become a promising solution for handling extreme conditions [3]. The essence of the multi-MGs formation (MMGF) problem is to identify the desired topology subject to various constraints. For the topology determination problem, mathematical programming [3], [4], [10] and heuristic search approaches [16], [17] are widely used methods. A mixed-integer non-linear programming (MINLP) model was built in [3] to sectionalize the outage area into networked MGs. The MMGF problem was formulated as a mixed-integer linear programming (MILP) model in [4], and the model was further improved in [10] by reducing both binary and continuous variables. Based on the graph theory, [16] developed a graph-theoretic search algorithm to identify a post-outage DS topology. Another heuristic approach was proposed in [17] to approximately solve the MG formation problem of large-scale systems with tractable computation. A good summary of existing methods, especially for the radial topology consideration when MGs are being formed, was provided in [18].

Manuscript received October 5, 2021; revised January 14, 2022; accepted March 5, 2022. Date of publication March 17, 2022; date of current version June 21, 2022. This work was supported in part by the National Science Foundation (NSF) under Award ECCS-1809458, and in part by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy under Contract DEAC05-00OR22725. Paper no. TSG-01611-2021. *(Corresponding author: Fangxing Li.)*

Jin Zhao and Fangxing Li are with the Department of EECS, The University of Tennessee, Knoxville, TN 37996 USA (e-mail: fli6@utk.edu).

Srijib Mukherjee and Christopher Sticht are with the Electrification and Energy Infrastructures Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830 USA.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSG.2022.3160387.

Digital Object Identifier 10.1109/TSG.2022.3160387

The aforementioned MG formation strategies are mainly based on observable system conditions and environments with short-term considerations, while conditions under natural disasters might be uncertain and changeable [7]. The uncertain output of RES-DGs and unexpected damage to grids reduces the efficiency or even damages the initially formed MGs. Therefore, an adaptive and dynamic MG formation strategy is needed to further enhance the resiliency under unexpected system conditions. By continuously interacting with the environment and obtaining feedbacks, the deep reinforcement learning (DRL) method [19] promises to help the MMGF scheme obtain adaptability to changeable conditions.

As an efficient solution to handle Markov decision processes (MDPs), DRL methods have become an attractive method for intractable problems in power systems. Reference [20] cast the volt-VAR optimization to a deep Q-network (DQN) framework and finally realized adaptive voltage control under time-varying operating conditions. To achieve real-time service restoration, [21] proposed an imitation learning (IL) framework to improve the training efficiency of DRL methods. In terms of MGs, the DRL method showed satisfying performance in energy management problems [22]-[24]. However, because of the difficulty of ensuring feasible radial topology, few studies have discussed the MMGF problem using DRL methods. For the MMGF problem, the action space of DRL methods has exponential growth with the increase of the number of switches, which deteriorates the learning ability of DRL methods. Therefore, the DRL based dynamic MMGF is a valuable but challenging problem to study.

For the purpose of realizing an on-line dynamic MMGF, a new deep RL-based model-free real-time adaptive scheme is proposed in this paper to enhance grid resilience over a long time-horizon. First, the dynamic MMGF problem is formulated as an MDP, and the deep Q-learning based RL method is introduced as a promising solution. Second, holding the features of a spanning forest, a topology transformation method and an action-decoupling method based on convolutional neural networks (CNN) are developed to reduce the action space and mitigate tricky topology issues. Finally, several techniques, such as double DQN (DDQN), Epsilon-greedy based exploration, and specially designed multi-buffers, are implemented to improve the learning ability of the proposed DRL method.

The contributions of this paper can be summarized as follows: 1) A new DRL supported on-line dynamic MMGF scheme is proposed. A long time-horizon is considered to fully utilize the available DGs under major faults of the main grids. The original problem is reformulated using an MDP and a complete DRL framework is specially designed for the topology-transformable MGs. 2) The problem of large action space when applying DRL methods is mitigated. The topology transformation method and the CNN-based action-decoupling Q-network are developed to efficiently handle the issue of exponentially increasing action numbers. 3) The learning ability of the DDQN method is further improved to become the CNN-based multi-buffer double DQN (CM-DDQN) method. The CM-DDQN method has strong learning ability and satisfactory computational performance to provide a real-time adaptive MMGF strategy according to the newly updated system information.

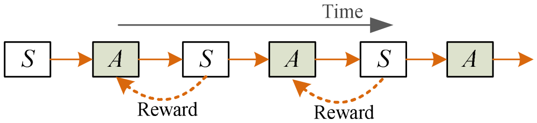


Fig. 1. MDP for the MMGF problem.

The rest of the paper is organized as follows: Section II reformulates the dynamic MMGF problem as a DRL based MDP. The topology transformation and the CNN based action-decoupling Q-network methods are provided in Section III. Section IV shows the detailed designs for the training and on-line application of the CM-DDQN based dynamic MMGF scheme. Section V provides case study results and discussions, followed by the conclusions of this work.

II. MMGF Problem Formation Using DRL Framework

This section introduces the dynamic MMGF problem with a DRL based MDP form. First, the dynamic MMGF problem is formulated to fit into an MDP from. Then, the solution is designed using the deep Q-learning structure with characteristics of the MMGF problem.

A. Formulate Dynamic MMGF as an MDP

The goal of the MMGF is transforming a DS into several self-supported islanded MGs [4]. Under a changeable environment, such as during extreme weather events, the dynamic MMGF maintains load supply during a time period by adaptively adjusting the topologies of multi-MGs. It is a sequential decision-making problem in a multi-step process. At each step, a topology configuration is determined to form islanded MGs through system reconfiguration and splitting based on the current state and the MMGF action of the last step.

Therefore, the dynamic MMGF problem can be described by an MDP which consists of four essential elements: state S, action A, state transition probability P, and reward r. In the MDP of the MMGF problem, the agent can be the distribution system operator (DSO). As shown in Fig. 1, the agent takes an action At based on the environment's state St at each time step t. Consequently, the agent gains a reward $r(St, At)$ and the state transitions to $St + 1$ according to the state transition probability $P(St+1|St, At)$. This state–action–next-state process is an interaction between the environment and agent, and it continues until the terminal state or the last step of setup [19].

Assume that the original DS totally has n nodes, l lines, w RCSs ($w \leq l$), n_g DG nodes and n_L load nodes and the time horizon is T. The binary variable α denotes the close ($\alpha = 1$) or open ($\alpha = 0$) statues of RCSs. Each MG that is derived from the original DS should be energized by a DG [4], [10].

1) *State:* The state is a part or all of characteristics of the current environment observed by the agent. The state is composed of the current network topology configuration $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_w]$, active and reactive DG output condition

$\boldsymbol{p}_{\mathrm{DG}} = [p_{\mathrm{DG},1}, \ldots, p_{\mathrm{DG},ng}]$ and $\boldsymbol{q}_{\mathrm{DG}} = [q_{\mathrm{DG},1}, \ldots, q_{\mathrm{DG},ng}]$, load amount $\boldsymbol{p}_{\mathrm{L}} = [p_{\mathrm{L},1}, \ldots, p_{\mathrm{L},ng}]$ and $\boldsymbol{q}_{\mathrm{L}} = [q_{\mathrm{L},1}, \ldots, q_{\mathrm{L},ng}]$ and the time *t*. Accordingly, the state at time *t* is defined as $\boldsymbol{S}_t = [\boldsymbol{\alpha}_t, \boldsymbol{p}_{\mathrm{DG},t}, \boldsymbol{p}_{\mathrm{DG},t}, \boldsymbol{p}_{\mathrm{L},t}, \boldsymbol{q}_{\mathrm{L},t}]$.

2) *Action:* The action is the reaction of the agent to the current state. In the dynamic MMGF problem, the action at time t $\boldsymbol{A}_t$ can be represented by changing the configuration of networks in $\boldsymbol{St}$. Therefore, the action space contains all the combinations of topology configuration $\boldsymbol{\alpha}_t$. However, the action space is quite large with $2^{\wedge w}$ combinations. Since the large action space creates problems of non-convergence in DRL methods, this will be further handled in Section III.

3) *Reward:* The reward is the feedback of the environment after the agent takes some action in a state. For the dynamic MMGF process, the action should first maintain the radial network of each individual MG. Further, the security constraints such as voltage and branch flow limits should be considered. The reward function (1) defines the MMGF problem.

$$r_t(\boldsymbol{S}_t, \boldsymbol{A}_t) = \begin{bmatrix} f_{\mathrm{topo}}(\alpha_t) - f_{\mathrm{AC}}\left(\boldsymbol{p}_t, \boldsymbol{q}_t, \alpha_t\right) \\ -f_{\mathrm{swi}}(\alpha_0, \ldots, \alpha_t) - f_{\mathrm{pb}}\left(\boldsymbol{p}_t, \alpha_t\right) \end{bmatrix} \tag{1}$$

The reward function contains the first term as the reward and the last two terms as penalties of the current action. Specifically, the first term f_{topo} (.) represents the reward if switch action $\boldsymbol{\alpha}_{t+1}$ successfully forms multi-MGs with radial networks. The second term function f_{AC} (.) represents the penalty based on AC power flow results. The third function f_{swi} (.) punishes frequent close/open actions of each switch. The last function f_{pb} (.) represents the penalty for insufficient power supply.

Note that the reward provides immediate feedback for step one. However, the dynamic MMGF problem is concerned with long-term feedback with cumulative rewards. Thus, the return (2) in which the accumulation of the current reward and the discounted future rewards is defined.

$$R_t = \sum_{k=0}^{T-1} \gamma^k r_{t+k} \quad \gamma^k \in [0, 1] \tag{2}$$

where γ^k is the discount factor.

B. Dynamic MMGF Using Deep Q-Learning

Different from classic dynamic programming methods, the DRL method does not require either explicit policies and value functions for MDPs or complete knowledge of MDPs. Reference [19]. Therefore, DRL is a promising approach to solving the complex dynamic MMGF problem. The well-trained DRL method can quickly provide an on-line scheme for dynamic MMGF, which helps the original grid give an adaptive reaction under changeable environments. This further enhances system resilience.

As introduced in Section II-A, the dynamic MMGF problem has both is with discrete action space as well as continuous state space. This feature makes it suitable to apply the DQN method [25]. The DQN method is a combination of a deep neutral network (DNN) and Q-learning which updates the action-value function iteratively. For a policy π, define the action-value (Q-value) function as (3),

$$\mathrm{Q}^{\pi}(S_t, A_t) = \mathbb{E}[R_t | S_t, A_t, \pi] \tag{3}$$

where Q-value is the expected discounted reward for executing action *At* at state *St* and following policy π. The objective of Q-learning is to estimate the value for an optimal policy. It has been proven that an optimal policy can be derived from the optimal Q-values $Q^*(St, At) = \max \ Q^{\pi}(St, At)$ by selecting the action with the highest Q-value in each state [26]. Therefore, the agent can decide how to properly perform actions by learning the Q values. For the dynamic MMGF problem, the Q-value can guide proper MMGF decisions in the MDP introduced in Section II-A.

Based on the Bellman equation, (3) can be further represented as a recursive format (4). As a form of temporal difference (TD) learning, Eq. (5) can update the Q-value towards the targeted Q-value with the learning rate η.

$$\mathrm{Q}^{\pi}(S_t, A_t) = \mathbb{E}\left[r_t + \gamma Q^{\pi}\left(S_{t+1}, A_{t+1}\right)\right] \tag{4}$$

$$\begin{aligned} \mathrm{Q}(S_t, A_t) \leftarrow\ & \mathrm{Q}(S_t, A_t) \\ & + \eta\left[r_t + \gamma \max_{A_{t+1}} \mathrm{Q}\left(S_{t+1}, A_{t+1}\right) - \mathrm{Q}(S_t, A_t)\right] \end{aligned} \tag{5}$$

Theoretically, the convergence of the iterative process is guaranteed, which means that Q^* (*St, At*) can be found [26]. The agent can be guided to optimally perform actions using Q^* (*St, At*). However, it is difficult to function the Q-value in the dynamic MMGF problem, and it is hard to provide a reasonable Q-table because of continuous state space. Therefore, the Q-value function is approximated via a deep neural network (DNN) parameterized by θ (6). As such, the original Q-learning method is transformed into the DQN method [25].

$$\mathrm{Q}(S_t, A_t) \approx Q(S_t, A_t | \theta) \tag{6}$$

The DNN based Q-value is updated with a loss function representing the mean-squared TD error, as shown in (7). For the MDP, the first two terms in (7) represent the direct reward of the current action and the potential value of the current action for a future MDP, respectively. Together, they measure the value of the current action. The last term directly generates the value of the current action using the Q-network. By minimizing the loss function, the Q-network gradually learns to generate Q-values guiding proper MMGF schemes.

$$L(\theta) = \left[r_t + \gamma \max_{A_{t+1}} \mathrm{Q}\left(S_{t+1}, A_{t+1} | \theta\right) - \mathrm{Q}(S_t, A_t | \theta)\right]^2 \tag{7}$$

III. Action Generation of Dynamic MMGF Process

In this section, the problem of large action space when applying DRL methods is mitigated. First, a topology transformation method is used to handle the radial topology requirement. Then, the CNN-based action-decoupling Q-value is designed to further handle the large action number.

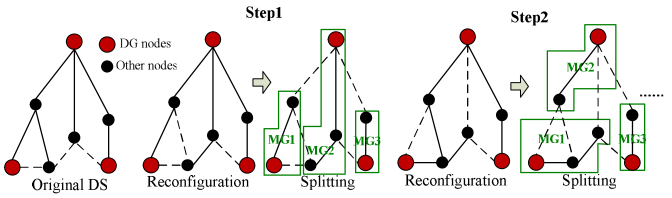


Fig. 2. The process of multi-MGs formation.

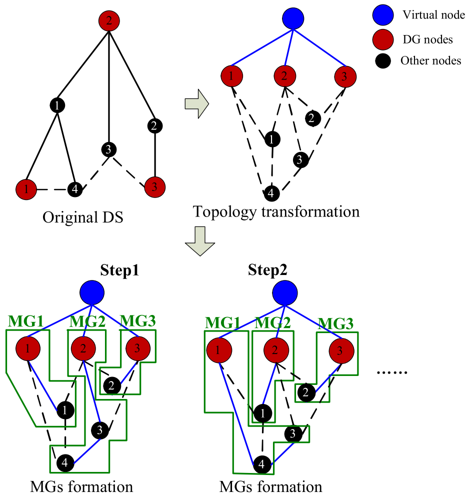


Fig. 3. Topology transformation for MMGF.

A. Search Space Reduction of Spanning Forest

Since the action space contains all the combinations of topology configuration $\boldsymbol{\alpha}_t$, the original scale of the action space is $2^{\wedge w}$. This exponentially increasing action space creates problems for the convergence of (5), and it contains tremendous infeasible network configurations because the radial network of each MG needs to be maintained [4], [18]. Moreover, the infeasible topology and the computation burden make power flow calculation-based environment interaction difficult to perform. Therefore, topological issues need be addressed.

From the point of changing topology, splitting a tree leads to several trees. Therefore, the MMGF problem includes the reconfiguration and splitting of the original DS. As shown in Fig. 2, all the reasonable radial MGs can be found by: 1) reconfiguring the original radial DS via switching operations, and 2) splitting the reconfigured DS by opening any closed switches.

The essence of the reconfiguration and splitting process is to find all the spanning forests of a network topology. Since it is intractable to directly trace all the spanning forests, a topology transformation method, as shown in Fig. 3, is applied to simplify the problem.

First, the DG nodes are picked up and connected using a virtual node (the blue node in Fig. 3). Accordingly, the spanning forest problem can be simplified as the spanning tree problem. In this way, finding MMGFs is equivalently transformed into the problem of finding radial networks of the new topology. The radial topology of the new topology can be ensured by two conditions [18]: 1) $n-n_{\mathrm{g}}$ RCSs switch on (the virtual node is not included in n) and 2) all the nodes must be connected.

The first condition reduces the action space from the exponential form $(2^{\wedge w})$ to the polynomial form $\mathrm{C}^{w}_{n-n_{\mathrm{g}}}$. This is how it works: the original DS has n nodes, n_{g} DG nodes and w RCSs in total. So, the number of original possible combinations is 2^{w}. After applying the proposed topology transformation method, the new topology has $n+1$ nodes (including the virtual node) and $w+n_{\mathrm{g}}$ lines (including the virtual lines between the virtual node and DG nodes). To ensure radial topology, the new network should have n lines in operation. Because n_{g} virtual lines have already been in operation, $n-n_{\mathrm{g}}$ RCSs should be switched on. Therefore, the problem is transformed into picking up $n-n_{\mathrm{g}}$ RCSs in w RCSs, which has $C^{w}_{n-n_{\mathrm{g}}}$ possible combinations. This is significantly less than the original $2^{\wedge w}$ combinations.

Based on the first condition, the second condition can be used to check the feasibility of the network.

B. CNN-Based Action-Decoupling Q-Value

The key purpose of the DQN method is to learn Q values by building and training the neural network. Regarding power related applications, the DQN method is normally based on the structures of an artificial neural network (ANN) [27] or DNN [18], [28]. However, as defined in (3), the Q-value is the expected accumulation of the discounted reward functions. The reward function for the MMGF problem heavily depends on the AC power flow calculation which has a sparse function relationship [29], [30], while the neighboring branches have a stronger topology relationship. Therefore, the neural network is organized using a CNN which has a strong automatic feature learning ability for processing data with a grid-like topology with sparsely connected features. The CNN promises to better learn the Q values of the MMGF problem.

The data preprocessing for the deep CNN of the Q-value is based on the new topology in Fig. 3. The input data contains node active and reactive power injection vectors $\boldsymbol{P}^{\mathbf{0}}$ and $\boldsymbol{Q}^{\mathbf{0}}$, branch resistance and reactance elements $\boldsymbol{R}^{\mathbf{0}}$ and $\boldsymbol{X}^{\mathbf{0}}$ elements, switch open/close statues $\boldsymbol{W}$ and frequencies of switching operation $\boldsymbol{F}$. In order to ensure the consistent dimension, $\boldsymbol{P}^{\mathbf{0}}$ and $\boldsymbol{Q}^{\mathbf{0}}$ are extended with l-n zeros (if $l > n$) to become $\boldsymbol{P}$ and $\boldsymbol{Q}$, and $\boldsymbol{R}$ and $\boldsymbol{X}$ are formed by considering the original branch parameters $\boldsymbol{R}^{\mathbf{0}}$ and $\boldsymbol{X}^{\mathbf{0}}$, as well as the switch status $\boldsymbol{W}$. The input is organized as $[\boldsymbol{P}; \boldsymbol{Q}; \boldsymbol{R}; \boldsymbol{X}; \boldsymbol{F}]$.

Fig. 4 shows the process of generating a Q-value using the CNN structure. Assuming the first convolutional layer has filters of the size [3, 3, 1, 12] where the first three numbers are the height, width and depth of one filter and the last number is the number of filters. Zero-padding is applied to maintain the original size of the input data. The filter of *Conv2* has the size [3, 3, 12, 24]. Hence, the output of *Conv2* has the size $[5, w, 24]$ and it is further flattened as a vector with the size $[1, 5 \times w \times 24]$ and goes through a *FC1* layer. Using a matrix of the weight parameters with the size $[5 \times w \times 24, w]$ and a $[1, w]$ parameter, the output will become a vector with the size $[1, w]$.

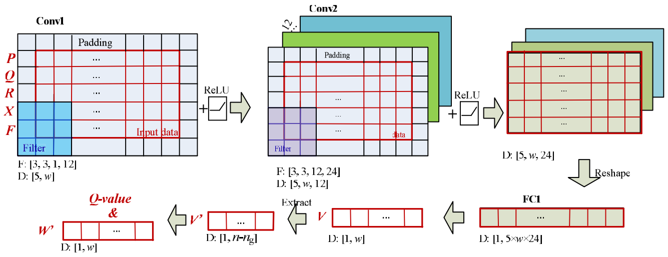


Fig. 4. Structure of CNN for Q-value.

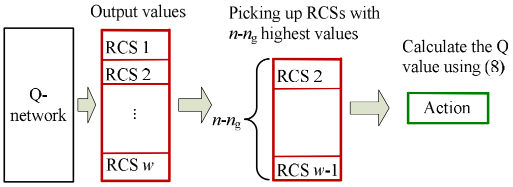


Fig. 5. Design of action-decoupling Q-value.

Although the action number has been reduced by the topology transformation method, the amount of CNN output data can still be large if each action is considered as an output. Therefore, instead of taking each output data as an action, an action-decoupling method is designed, as shown in Fig. 5, by setting the CNN output $\boldsymbol{V}$ as the values of switches. Accordingly, the Q-value does not take a specific value of the DQN output; instead, it is set to the average value of a selected subset in the CNN output data $\boldsymbol{V}$. That is, top n-n_{g} values in $\boldsymbol{V}$ are selected and extracted as $\boldsymbol{V}'$ and the Q-value (8) is the average of values in $\boldsymbol{V}'$. In general, the action-decoupling method takes the outputs $\boldsymbol{V}$ of CNN as values for each switch, selects the closing switches, and then calculates the Q-value. When transforming $\boldsymbol{V}$ to $\boldsymbol{V}'$, we may record the switch index numbers of the top n-n_{g} values, set 1 (closed) to these switches and 0 (open) to the remaining switches. Then, the new switch statues in $\boldsymbol{W}'$ are obtained by closing the selected switches and opening the remaining ones.

$$\mathrm{Q}\big(S_{t+1}, A_{t+1}|\theta\big) = {}^{\mathrm{sum}(V')}\!/_{n\,-\,n_g} \tag{8}$$

A *min_max_scaler* transformation is applied to normalize the input data $\boldsymbol{P}$; $\boldsymbol{Q}$; $\boldsymbol{R}$; $\boldsymbol{X}$ and output data $\boldsymbol{V}$. The frequencies of switching operations in $\boldsymbol{F}$ are normalized by dividing the total step number. Switch open/close statues in $\boldsymbol{W}$ are originally binary. Through normalization, the values of the data are within the range [0, 1] which helps create a more regular search region for faster convergence of the algorithm. The loss function used to train the CNN is given by (7). The reward function for the MMGF problem is formulated considering the topology requirements, power balance, voltage, branch flow and switch operation times which will be further functioned in Section IV-B.

IV. Learning and Application of CM-DDQN

The learning and application of the proposed CM-DDQN method are discussed in this section. First, techniques for better learning are incorporated in the DRL method. Then, detailed designs of the reward function and the RL process for a dynamic MMGF are presented. Finally, the entire method as well as the on-line application framework is shared.

A. Techniques for Better Learning

The key to realizing the DRL based dynamic MMGF is to let the Q-network learn the proper reactions in the MDP. Many techniques have been studied for the efficient DQN learning. The experience replay, Epsilon-greedy based exploration, and fixed network are the three most efficient techniques.

1) Experience Replay and Multiple Buffers: The DQN learns the Q-value based on previous experiences. However, for the MDP, the previous experiences are overwritten with new experiences. This largely reduces data efficiency. Therefore, the experience replay method [25] is applied to memorize the experiences and re-train the Q-network. Accordingly, each experience can be used repeatedly, and bias due to correlation between training samples can be eliminated.

The experience replay consists of a *memory* part and a *replay* part. The *memory* contains a list of previous experiences and observations to re-train the Q-network. In the MDP, state $\boldsymbol{S}_t$, action $\boldsymbol{A}_t$, reward $\boldsymbol{r}_t$,, next state $\boldsymbol{S}_{t+1}$ and topology surviving condition $\boldsymbol{D}$ are appended to the *memory*1. As long as the memory stores enough experiences, the replay part is activated.

$$memory1 = [\ldots, (S_t, A_t, r_t, S_{t+1}, D_t), \ldots] \tag{9}$$

Since the learning of the Q-network largely depends on the experience, reserving a part of the good experience helps the Q-network learn proper reactions. For the MMGF problem, good experiences have high reward values and feasible topology. Accordingly, multi-buffers are constructed with extra *memories* to reserve good experience extracted from the original *memory*. The replay part randomly extracts some experiences from buffers to organize a *minibatch* to train the Q-network.

$$memory2 = \left[\ldots, \left(S_t, A_t, r_t^*, S_{t+1}, D_t\right), \ldots\right] \tag{10}$$

$$memory3 = \left[\ldots, \left(S_t, A_t, r, S_{t+1}, D_t^*\right), \ldots\right] \tag{11}$$

The training of the Q-network is enhanced using the *minibatch* in the MDP.

$$minibatch = \begin{bmatrix} \text{random.sample}(memory1, batch\_size1), \\ \text{random.sample}(memory2, batch\_size2), \\ \text{random.sample}(memory3, batch\_size3) \end{bmatrix} \tag{12}$$

2) Epsilon-Greedy Based Exploration: Since the CNN based Q-network is initialized with random weights and biases, it's difficult for its performance to be satisfying in the early stages. Therefore, instead of selecting actions that directly use the not well-trained Q network, it is better to try all possibilities before the pattern starts to. The random selection of actions is called 'exploration', while the prediction using DQN is 'exploitation'. The *Epsilon* represents the exploration rate which is a certain percentage at which the agent randomly selects its actions. The Epsilon-greedy method uses an annealing ε value to guide 'exploration' and 'exploitation'. As shown in (13), with the constant k controlling the annealing speed, the

ε value gradually decreases after the DQN training begins. In each MDP step, the agent randomly extracts a value of [0, 1]. Then, the agent selects the action with the largest Q value if the random value is less than ε; otherwise, a random action will be selected.

$$\varepsilon = \begin{cases} \mathrm{k}\varepsilon, & \varepsilon > \varepsilon_{\min} \\ \varepsilon_{\min}, & \varepsilon \le \varepsilon_{\min} \end{cases} \tag{13}$$

3) Fixed Q Network (Double DQN): Because the update process (5) picks up the maximum Q value of the next state, the overestimation becomes a long-standing problem for all Q-learning based algorithms. In order to address this issue, a DDQN is proposed [30] with better results on ATARI 2600 games than other Q-learning based methods. Therefore, it is applied in the deep Q learning based MMGF scheme.

The DDQN has two separate neutral networks: the original Q network and the target Q (T-Q) network, which decouples the action selection and action evaluation. The original Q network is used to select the action with maximum Q value while the T-Q network evaluates the Q value of the selected action. The T-Q network is a fixed network which is not updated in the Q network updating process. The fixed features enhance the efficiency and stability in the learning process. The loss function (7) is adjusted into (14) accordingly.

$$L_t(\theta) = \begin{cases} [r_t - Q(S_t, \mathrm{A_t}|\theta)]^2 & (t = \mathrm{T}) \\ \begin{bmatrix} \mathrm{r_t} + \gamma \max\limits_{A_{t+1}} T - Q\big(S_{t+1}, A_{t+1}\big|\theta^{Tar}\big) \\ -Q(S_t, A_t|\theta) \end{bmatrix}^2 & (t \ne \mathrm{T}) \end{cases} \tag{14}$$

Based on the original deep Q learning structure, the extra designs 1-3) give the DQN good performance in dealing with the overestimation issue and provide better learning processes. At this point, all the designs for the CM-DDQN method are presented.

B. CM-DDQN Learning Process

Defining a suitable reward function is an indispensable part to completing the learning process of DRL methods. The detailed reward function (1) is shown in (15) to help determine the Q network of the dynamic MMGF problem.

$$r_t(\boldsymbol{S}_t, \boldsymbol{A}_t) = \begin{cases} \begin{bmatrix} f_{\mathrm{topo}}(\alpha_t) - \sum_{i \in M_g} f_{\mathrm{AC},i}\big(\boldsymbol{P}_t, \boldsymbol{Q}_t, \alpha_t\big) \\ -f_{\mathrm{swi}}(\alpha_0, \ldots, \alpha_t) - f_{\mathrm{pb},i}(\boldsymbol{P}_t, \alpha_t) \end{bmatrix} & \mathrm{S_{topo}} = 1 \\ f_{\mathrm{topo}}(\alpha_t) & \mathrm{S_{topo}} = 0 \end{cases} \tag{15}$$

where $\mathrm{S_{topo}}$ is the signal to show whether the switch action $\boldsymbol{\alpha}_t$ successfully forms multi-MGs with radial networks, and M_{g} is the set of newly formed MGs. As shown in (16), if $\mathrm{S_{topo}} = 1$, $f_{\mathrm{topo}}(.)$ provides the reward w; otherwise, f_{topo} (.) gives punishment –w and the 'game over' signal of an MDP is triggered because of the infeasible topology. Functions related to AC power flow $f_{\mathrm{AC,i}}(.)$, switch status $f_{\mathrm{swi}}(.)$ and power balance $f_{\mathrm{pb}}(.)$ are further explained in (17), (18) and (19), respectively.

$$f_{\mathrm{topo}}(\alpha_t) = \begin{cases} \mathrm{w} & S_{\mathrm{topo}} = 1 \\ -\mathrm{w} & S_{\mathrm{topo}} = 0 \end{cases} \tag{16}$$

$$f_{\mathrm{AC},i}\big(\boldsymbol{P}_t, \boldsymbol{Q}_t, \alpha_t\big) = \sum_{j \in i} p_{\mathrm{vol},j} + p_{\mathrm{loss},i} + \sum_{l \in i} p_{\mathrm{bran},l} \tag{17}$$

$$f_{\mathrm{swi}}(\alpha_0, \ldots, \alpha_t) = \sum_{j \in w} p_j \tag{18}$$

$$f_{\mathrm{pb},i}(\boldsymbol{P}_t, \alpha_t) = \begin{cases} 0, & M \le 0 \\ M, & M > 0 \end{cases} \tag{19}$$

$$M = \sum_{j \in i} p_{\mathrm{L},j} + p_{\mathrm{loss},i} - \sum_{j \in i} p_{\mathrm{DG},j} \tag{20}$$

The AC power flow related function $f_{\mathrm{AC},i}(.)$ provides the punishment value of forming MG i. It contains penalties of voltage violation $p_{\mathrm{vol},j}$ (21), system power loss of $p_{\mathrm{loss},i}$, and branch overflow $p_{\mathrm{bran},l}$ (22). The $f_{swi}(.)$ function punishes frequent close/open actions of each switch using (23) which works if switch j exceeds the allowed number of operations in the whole dynamic MMGF process. As shown in (19) and (20), $f_{\mathrm{pb}}(.)$ gives punishes to MG i if it has power deficiency.

$$p_{vol,j} = \begin{cases} 0, & 0.95 \le V_j \le 1.05 \\ \mathrm{p}_{vol}, & otherwise \end{cases} \tag{21}$$

$$p_{\mathrm{bran},l} = \begin{cases} \big(L_l - L_{\mathrm{upp},l}\big)/\mathrm{B_{base}}, & L_l > L_{\mathrm{upp},l} \\ 0, \ L_l & \le L_{\mathrm{upp},l} \end{cases} \tag{22}$$

$$p_j = \begin{cases} 0, & \sum_{s=1}^{t} \big|\alpha_{j,s} - \alpha_{j,s-1}\big| \le S_{\mathrm{upp},j} \\ \sum_{s=1}^{t} \big|\alpha_{j,s} - \alpha_{j,s-1}\big|/\mathrm{N_{step}}, & otherwise \end{cases} \tag{23}$$

where V_j is the voltage amplitude of node j, L_l and $\mathrm{L}_{\mathrm{upp},l}$ are respectively the absolute value and upper bound of branch power of line l, $\mathrm{B_{base}}$ is the base value to standardize the branch flow penalty, $\mathrm{S}_{\mathrm{upp},j}$ is the allowed operation number of switch j, and $\mathrm{N_{step}}$ is the required step of the dynamic MMGF process.

The CM-DDQN learning process for the MMGF problem is provided in the above algorithm description. Therein, there are hard constraints and soft constraints. A "game over" is triggered if any hard constraint is violated, while soft constraint violations lead to certain consequences instead of an immediate "game over". In the dynamic MMGF problem, the hard constraint is the feasible topology requirement. If the switch actions cannot ensure a radial network for each MG, there will be a 'game over' and the MDP is directly ended. The voltage limit, branch flow limit, switching number limit and power balance limit belong to soft constraints which form penalties to organize the reward value of the switch on/off decisions in the current step.

C. Deep RL Based Dynamic Model-Free MMGF Scheme

The whole DRL based on-line dynamic MMGF process is shown in Fig. 6. Since the feasible topology is hard to learn quickly, a pre-training part is prepared to make the Q-network capture some topology and power flow related features of the original system. This helps the DRL scheme to be used directly under emergencies or major fault conditions of the main grid. In a major fault event process, the pre-trained Q-network will quickly provide the MMGF scheme to make full use of the current available DERs. Meanwhile, new experience will be recorded in the buffer and the Q-network can

Algorithm 1 CM-DDQN Learning Process

```
Input: DG generation data set $\{\boldsymbol{p}_{DG}^1, \ldots, \boldsymbol{p}_{DG}^T\}$ and $\{\boldsymbol{q}_{DG}^1, \ldots, \boldsymbol{q}_{DG}^T\}$, load data set $\{\boldsymbol{p}_L^1, \ldots, \boldsymbol{p}_L^T\}$ and $\{\boldsymbol{q}_L^1, \ldots, \boldsymbol{q}_L^T\}$. Initial topology of the original DS $\boldsymbol{\alpha}^0 = [\alpha_1^0, \ldots, \alpha_w^0]$. Apply topology transformation in Fig. 3.
Output: well-trained action-value Q network
S1: Initialization. Initialize Q network and T-Q network with same random weights and bias. Initial replay memory1-3 with capacity maxlen. Set $D^{\text{step}} = 0$. Set batch size, Episode M, step number T and Epsilon-greedy parameters.
S2: for Episode from 0 to M do
      Initialize state $\boldsymbol{S}^0 = [\boldsymbol{P}^0; \boldsymbol{Q}^0; \boldsymbol{R}^0; \boldsymbol{X}^0; \boldsymbol{F}^0]$
      for Step from 1 to T do
         Perform Epsilon-greedy, and randomly select an action $\boldsymbol{\alpha}^{\text{step}}$ or $\boldsymbol{\alpha}^{\text{step}} = \arg\max\,[Q(St, At)]$.
         Calculate reward value (15). If topology infeasible, set $D^{\text{step}} = 1$.
         Organize new state $\boldsymbol{S}^{\text{step}+1}$. Note that $\boldsymbol{R}$, $\boldsymbol{X}$ and $\boldsymbol{F}$ are updated according to $\boldsymbol{\alpha}^{\text{step}+1}$ while $\boldsymbol{P}$ and $\boldsymbol{Q}$ follow DG generation and load data sets.
         Add record $[\boldsymbol{S}^{\text{step}}, \boldsymbol{\alpha}^{\text{step}+1}, r^{\text{step}+1}, \boldsymbol{S}^{\text{step}+1}, D^{\text{step}},]$ in memory. Add record to memory2 if $r^{\text{step}+1} \geq r_{\text{std}}$. Add record to memory3 if $\boldsymbol{D}^{\text{step}} = 0$.
         If topology is infeasible/$\boldsymbol{D}^{\text{step}} == 1$ do
            Update T-Q-CNN as Q-CNN
            Break;
          End if
         If conditions for replay are satisfied do
            Randomly select batch size records from memory. Train Q network (Q-CNN) using loss function (14).
            If Step = T do
               Update T-Q-CNN as Q-CNN
            End if
         End if
      End for
   End for
S3: Obtain the Q-CNN.
```

be further updated when the training condition is trigged. Specifically, in the on-line application process, a ‘do-nothing’ module [32] can be added to ensure the topology feasibility. The actions produced by the Q-network will be re-checked using the topology check module in the environment. The ‘do-nothing’ module is triggered if the re-organized network is infeasible, and the produced action will not be implemented to maintain the feasible radial network of the last step.

As such, the DRL based dynamic MMGF scheme can satisfy the real-time computation requirement of the on-line application because the RL agents obtain switch on/off decisions very quickly through simple numerical calculations. Without the actual power system modeling or power flow equations, the on-line application can be performed in a model-free way. On the other hand, the agent can keep on learning new experiences according to newly updated system conditions and resilient reactions, which improves the system's adaptability to handle changeable event conditions. The essence of the proposed dynamic MMGF scheme is to enhance system resilience by flexibly changing topology to form different self-supported MGs according to the newly updated system conditions.

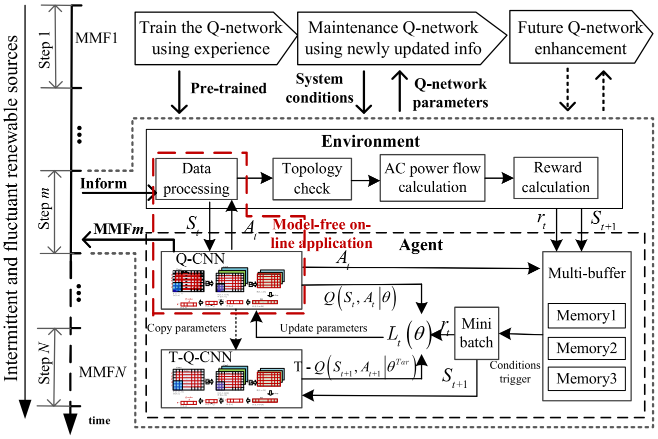


Fig. 6. CM-DDQN based dynamic MMGF process.

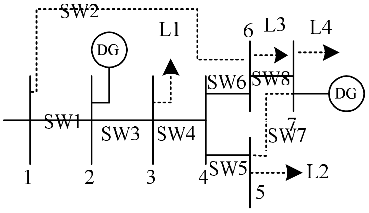


Fig. 7. The 7-bus system.

V. Case Study

In this section, the training and application performance of the proposed CM-DDQN based dynamic MMGF scheme is demonstrated. The dynamic MMGF scheme is compared with the conventional initially-formed MGs schemes [4], [10]. The proposed CM-DDQN is compared with the DDQN [28].

Two systems are used: the 7-bus system with 2 DGs and the IEEE 123-bus system with 12 integrated DGs. The time horizon is set as 200 minutes, and each MG formation transformation step is set as 10 minutes. The switch limit for each RCS is set as 4. The initial value of ε is 1 and $\varepsilon_{\min} = 0.1$. Filters of the CNN are with the size [5, 5, 1, 12] and [5, 5, 1, 24]. The uncertain data of DG outputs follow a 3 sigmas normal distribution with 20% forecast error from the expected values. The DRL codes and the corresponding environment are written and compiled in Python 3.7 while the CNN is built using TensorFlow 2.2. and Keras 2.4. Pypower 5.1 is applied to solve the power flow calculation in the environment. All simulation studies were conducted on a computer with Intel Core i7-8550U CPU and 16 GB RAM.

A. DRL Based Dynamic MMGF Process Using 7-Bus System

1) Pre-Training of Deep RL: For the 7-bus system, the episode number for training is 500, and the DG output values

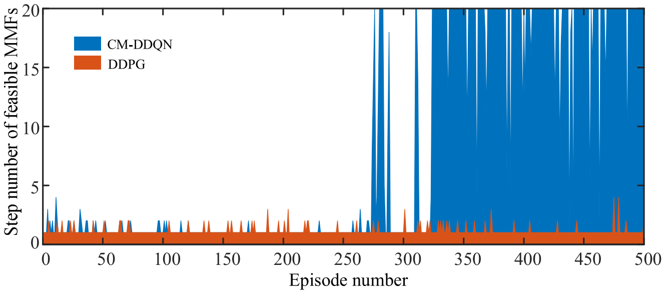


Fig. 8. Successful MMGF steps in the training process.

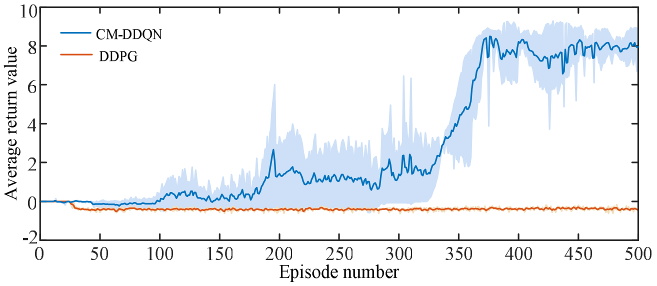


Fig. 9. Convergence progression of the return of 7-bus system.

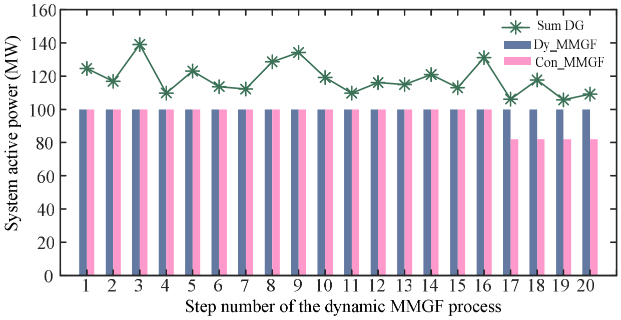


Fig. 10. DG output and reserved load amount in 20 steps.

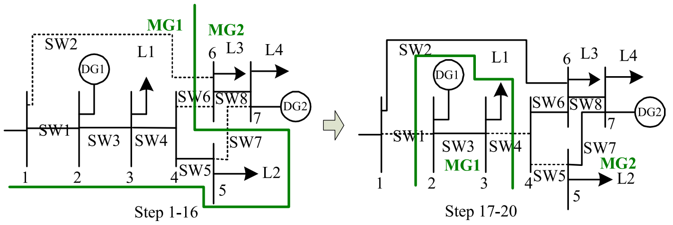


Fig. 11. Dynamic Multi-MG formation.

for each step of the 500 episodes are randomly generated. The numbers of input and output data are 5×8 and 8. Since the feasible topology is regarded as the hard constraint, the topology condition is the primary concern throughout the entire process. Note that the switch on/off decisions are obtained from the Q-network output value of each switch. Using the method introduced in Section III-B, the switches with the top 5 highest values according to the CNN output data $\boldsymbol{V}$, are regarded as switched on, and the rest are switched off.

The deep deterministic policy gradient (DDPG) method is applied in this case to compare with the proposed CM-DDQN. The actor-network of DDPG has 8 outputs representing 8 switches, and the output of the actor-network is limited to 0∼1. If the output value is larger than 0.5, then, switch on; otherwise, switch off.

The number of steps with radial networks of each MG is shown in Fig. 8. A 20-step feasible topology condition means a successful dynamic MMGF process, while any infeasible topologies in the process directly lead to a 'game over' which means the end of an episode. As can be observed, after almost 350 episodes, the proposed DRL method successfully learned how to form feasible topologies by providing switch on/off decisions in the dynamic multi-step process.

After the switch statues are determined, the reward function (16) can be calculated according to the topology check and AC power flow calculation in the environment. The convergence process of the return, which is obtained using the Q value and the reward, is shown in Fig. 9. The values of the return are organized according to ten separate trainings. The maximum and minimum return values are extracted from the ten separate trainings and become the upper and lower bounds, which form the light blue area in Fig. 9. As shown in the figure, from 0 to 330, the return first goes through an exploration process with low values; then, it increased rapidly with an episode increasing from 330 to 370. After that, the return value becomes relatively stable with small oscillations. Since the return value contains a comprehensive consideration of the topology condition, voltage violation, branch overflow, switch number limits and power balance, the convergence means the Q-network can reasonably judge the performance of an action. As the comparison shows, the output values of the DDPG critic-network stay below zero, meaning it is an unsuccessful learning process.

2) Comparison of Performances of Different Schemes: After the Q-network learned to properly judge the performance of actions, a series of uncertain output of DGs in the MDP is randomly extracted to test the performance of the proposed method. The total DG output in 20 steps is shown with the green line in Fig. 10. With the same DG output conditions, the proposed DRL based dynamic MMGF scheme (Dy_MMGF) is compared with the conventional mathematical programming based MMGF scheme (Con_MMGF) [4], [10]. In steps 1-16, both schemes hold all the loads. However, the conventional MMGF scheme (Con_MMGF) sheds loads in steps 17-20, while the dynamic MMGF scheme (Dy_MMGF) still holds all the loads. With the whole dynamic process considered, the DRL based scheme properly dispatches RCS actions, and adjusts the topology based on newly updated system conditions. According, it shows better load-supplying ability.

As shown in Fig. 11, the proposed deep RL method changed the formation of MGs, and this action avoided load shedding. The uncertain output of DG1 is reduced to 45MW ∼ 49MW in steps 17-20. If the original MG formation is reserved, the MG1 will have to shed load L2 to ensure power balance and voltage security. Then, the Con_MMGF scheme has an 18 MW load reduction while the proposed Dy_MMGF scheme successfully holds all the load in the entire system.

B. Performance of CM-DQN in IEEE 123-Bus System

The basic data of the IEEE 123-bus system is obtained from [33]. The modified IEEE 123-bus system has 124 buses

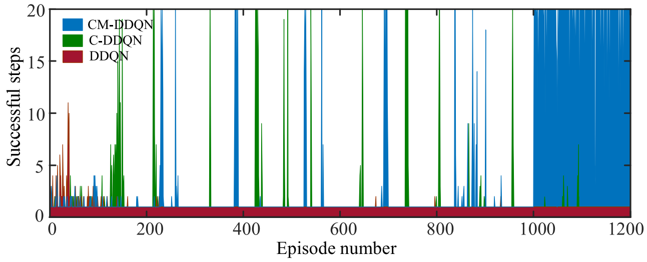


Fig. 12. Comparison of successful MMGF steps of two methods.

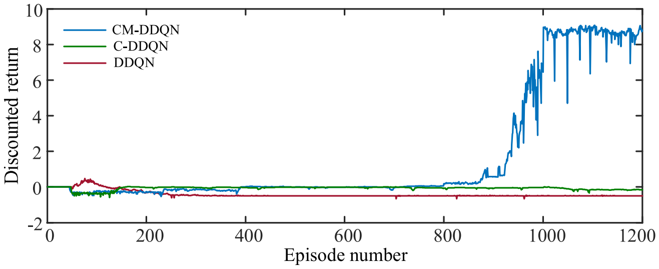


Fig. 13. Convergence progression of the return of IEEE123-bus system.

(including the main substation), 11 DGs, 125 lines and 13 RCSs. The original possible topology is 8192, and the number is reduced to 3432 using the topology transformation method in Section III-A. The size of input data is 125 × 5, while the output size is 13 × 1.

1) Comparison of Learning Abilities of DRL Methods: Using the modified IEEE 123-bus system, the learning ability of the proposed CM-DDQN is compared with the DDQN [28] method and the CNN based DDQN (C-DDQN) without the multi-buffer part.

As shown in Fig. 12, the large action space creates problems for the DDQN. Although it has the tendency to learn proper behavior with an increasingly successful MMGF step in the early period (about 1-50 episodes), the features are lost in the following training episodes, and finally lead to a failed training process. Without the multi-buffer part, the learning ability of the C-DDQN is unstable. Although it has learned proper actions with 20 successful MMGF steps in some episodes (e.g., about 430 and 740 episodes), it quickly lost the features and finally led to an unstable training process. With an improved CNN structure and a multi-buffer design, the proposed CM-DDQN successfully captures the feasible topology feature after about 1000 training episodes.

The corresponding return values of the three methods are demonstrated in Fig. 13. The DDQN and C-DDQN method fail to provide proper evaluation for actions, while the return of the CM-DDQN method reaches to a relatively stable condition. That is because the CNN has strong automatic feature learning ability in processing data with a grid-like topology with sparse connectivity, while the designed reward value is based on the AC power flow in the sparsely connected power system. In addition, the multi-buffer design provides stable and satisfying experiences for the learning process, which avoids losing good results explored previously. Note that the large fluctuation in steps 1000-1200 is caused by the exploration design (random action generation) with the lowest 1% probability. Even the Q-network is already well-trained in the training process, so the surviving of feasible topology may be reduced by following a randomly generated infeasible action.

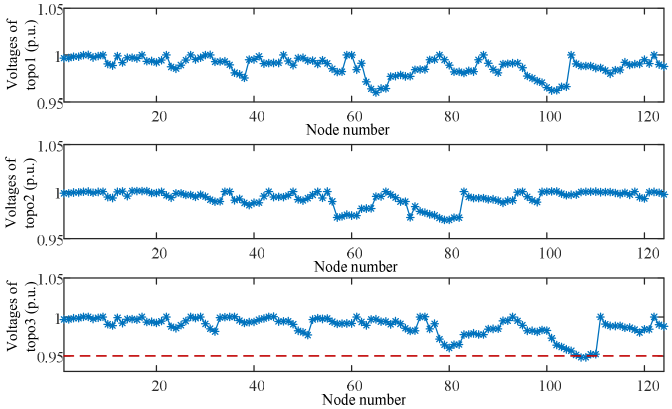


Fig. 14. Voltage conditions of three MMGF topologies.

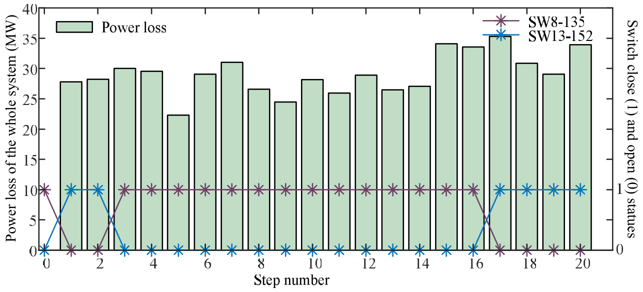


Fig. 15. Power loss and switch conditions of the whole process.

2) Computation Performance of the CM-DDQN: Taking a set of test data with a series of uncertain DG outputs as an example, the proposed CM-DDQN ensures radial networks with three topology forms (topo1, topo2 and topo3) in the 20 steps. Fig. 14 shows the worst voltage condition of the three topology forms. Because the voltage belongs to the soft constraint, there are few violations in the whole process with the proposed CM-DDQN. However, these slight violations are less than 1% of rated bus voltages and are easy to eliminate by local compensates. The corresponding power losses and most frequently operated switches 8-135 and 13-152 are presented in Fig. 15. The topology transformation happened at step 1, step 3 and step 17. The most frequently operated switch has 3 actions in the whole process, which is within the limit of 4 actions.

To verify the on-line application performance of the CM-DDQN method, 100 episodes of complete MDPs (2000 steps) are randomly extracted from the test data. The result is listed in Table I. For the two systems, there are no hard constraint violations, which means that the proposed method successfully ensures topology feasibility. For the soft constraints, the small-scale systems all have satisfaction performance without any violations. The large-scale system has 4 steps of voltage violations of 5 buses. However, the violations are all within 0.005 p.u., since serious voltage violations bring high penalties for the reward function. In the 100 episodes of MDP, there are 3 occurrences of switch violations with 5, 5 and 6 instances of switch operations, respectively. This phenomenon can be mitigated by increasing the corresponding penalty in the reward

TABLE I
RESULTS OF 100 EPISODES OF MDP

Case	Training time (h)	On-line computation time	Hard cons violation	Voltage violation	Switch violation	Brach flow violation
7-bus	5.27h	0.06s	0%	0%	0%	0%
123-bus	58.08h	0.104	0%	0.2%	3%	0%

function. However, it is not suggested to enhance the consideration of switching actions because it will not lead to security problems, and over-focus on soft constraints will influence the performance regarding hard constraints.

In on-line applications, the CM-DDQN provides a feasible MMGF strategy with about 0.1 s computational time even for the 123-bus system. Although the system scale is extended from the 7-bus to 123-bus, the computation time only increases slightly because the proposed method derives results from the model-free DRL structure rather than via the actual power-system model or power flow equation. This feature supports the dynamic on-line MMGF scheme by providing a timely topology-adjusting strategy according to newly updated system conditions.

VI. CONCLUSION

The changeable conditions caused by extreme events reduce the efficiency or even damage initially-formed MGs. In order to improve the adaptability of the MMGF scheme, this paper proposes a new DRL based dynamic on-line MMGF scheme. A DRL based MDP is designed to provide a solution for the transformable MMGF problem over a long time-horizon. A topology transformation as well as a CNN based action-decoupling Q-value is developed to handle the large action space problem. The DDQN is improved to formulate the CM-DDQN which enhances the learning ability for large-scale systems. The case study results demonstrate that the proposed dynamic on-line MMGF scheme enhances system resilience by holding all of the loads using feasible topology adjustment. The proposed CM-DDQN has strong learning ability, distinguished computation speed in real-time, and a satisfactory security guarantee.

ACKNOWLEDGMENT

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

REFERENCES

[1] J. Wang and H. Gharavi, “Power grid resilience,” *Proc. IEEE*, vol. 105, no. 7, pp. 1199–1201, Jul. 2017.

[2] Z. Li, M. Shahidehpour, F. Aminifar, A. Alabdulwahab, and Y. Al-Turki, “Networked microgrids for enhancing the power system resilience,” *Proc. IEEE*, vol. 105, no. 7, pp. 1289–1310, Jul. 2017.

[3] Z. Wang and J. Wang, “Self-healing resilient distribution systems based on sectionalization into microgrids,” *IEEE Trans. Power Syst.*, vol. 30, no. 6, pp. 3139–3149, Nov. 2015.

[4] C. Chen, J. Wang, F. Qiu, and D. Zhao, “Resilient distribution system by microgrids formation after natural disasters,” *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 958–966, Mar. 2016.

[5] Q. Shi *et al.*, “Post-extreme-event restoration using linear topological constraints and DER scheduling to enhance distribution system resilience,” *Int. J. Elect. Power Energy Syst.*, vol. 131, Oct. 2021, Art. no. 107029.

[6] Q. Shi *et al.*, “Network reconfiguration and distributed energy resource scheduling for improved distribution system resilience,” *Int. J. Elect. Power Energy Syst.*, vol. 124, Jan. 2021, Art. no. 106355.

[7] L. Che and M. Shahidehpour, “Adaptive formation of microgrids with mobile emergency resources for critical service restoration in extreme conditions,” *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 742–753, Jan. 2019.

[8] X. Liu, M. Shahidehpour, Z. Li, X. Liu, Y. Cao, and Z. Bie, “Microgrids for enhancing the power grid resilience in extreme conditions,” *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 589–597, Mar. 2017.

[9] C. Abbey *et al.*, “Powering through the storm: Microgrids operation for more efficient disaster recovery,” *IEEE Power Energy Mag.*, vol. 12, no. 3, pp. 67–76, May–Jun. 2014.

[10] T. Ding, Y. Lin, G. Li, and Z. Bie, “A new model for resilient distribution systems by microgrids formation,” *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 4145–4147, Sep. 2017.

[11] K. S. A. Sedzro, A. J. Lamadrid, and L. F. Zuluaga, “Allocation of resources using a microgrid formation approach for resilient electric grids,” *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 2633–2643, May 2018.

[12] S. Lei, J. Wang, and Y. Hou, “Remote-controlled switch allocation Enabling prompt restoration of distribution systems,” *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3129–3142, May 2018.

[13] Z. Wang, B. Chen, J. Wang, and C. Chen, “Networked microgrids for self-healing power systems,” *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 310–319, Jan. 2016.

[14] A. Hussain, V.-H. Bui, and H.-M. Kim, “Resilience-oriented optimal operation of networked hybrid microgrids,” *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 204–215, Jan. 2019.

[15] Q. Zhou, M. Shahidehpour, A. Alabdulwahab, and A. Abusorrah, “A cyber-attack resilient distributed control strategy in islanded microgrids,” *IEEE Trans. Smart Grid*, vol. 11, no. 5, pp. 3690–3701, Sep. 2020.

[16] J. Li, X.-Y. Ma, C.-C. Liu, and K. P. Schneider, “Distribution system restoration with microgrids using spanning tree search,” *IEEE Trans. Power Syst.*, vol. 29, no. 6, pp. 3021–3029, Nov. 2014.

[17] K. S. A. Sedzro, X. Shi, A. J. Lamadrid, and L. F. Zuluaga, “A heuristic approach to the post-disturbance and stochastic pre-disturbance microgrid formation problem,” *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5574–5586, Sep. 2019.

[18] S. Lei, C. Chen, Y. Song, and Y. Hou, “Radiality constraints for resilient reconfiguration of distribution systems: Formulation and application to microgrid formation,” *IEEE Trans. Smart Grid*, vol. 11, no. 5, pp. 3944–3956, Sep. 2020.

[19] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[20] Y. Zhang, X. Wang, J. Wang, and Y. Zhang, “Deep reinforcement learning based Volt-VAR optimization in smart distribution systems,”*IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 361–371, Jan. 2021.

[21] Y. Zhang, F. Qiu, T. Hong, Z. Wang, and F. Li, “Hybrid imitation learning for real-time service restoration in resilient distribution systems,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 2089–2099, Mar. 2022.

[22] Y. Du and F. Li, “Intelligent multi-microgrid energy management based on deep neural network and model-free reinforcement learning,” *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1066–1076, Mar. 2020.

[23] Q. Zhang, K. Dehghanpour, Z. Wang, F. Qiu, and D. Zhao, “Multi-agent safe policy learning for power management of networked microgrids,” *IEEE Trans. Smart Grid*, vol. 12, no. 2, pp. 1048–1062, Mar. 2021.

[24] J. Hao, D. W. Gao, and J. J. Zhang, “Reinforcement learning for building energy optimization through controlling of central HVAC system,” *IEEE Open Access J. Power Energy*, vol. 7, pp. 320–328, 2020.

[25] V. Mnih *et al.*, “Human-level control through deep reinforcementlearning,” *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[26] C. J. C. H. Watkins and P. Dayan, “Q-learning,” *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, May 1992.

[27] R. Rocchetta, L. Bellani, M. Compare, E. Zio, and E. Patelli, "A reinforcement learning framework for optimal operation and maintenance of power grids," *Appl. Energy*, vol. 241, pp. 291–301, May 2019.

[28] S. H. Oh, Y. T. Yoon, and S. W. Kim, "Online reconfiguration scheme of self-sufficient distribution network based on a reinforcement learning approach," *Appl. Energy*, vol. 280, pp. 2–15, Dec. 2020.

[29] Y. Du, F. Li, J. Li, and T. Zheng, "Achieving 100x acceleration for N-1 contingency screening with uncertain scenarios using deep convolutional neural network," *IEEE Trans. Power Syst.*, vol. 34, no. 4, pp. 3303–3305, Jul. 2019.

[30] J. Zhao, F. F. Li, X. Chen, and Q. Wu, "Deep learning based model-free robust load restoration to enhance bulk system resilience with wind power penetration," *IEEE Trans. Power Syst.*, early access, Sep. 24, 2021, doi: 10.1109/TPWRS.2021.3115399.

[31] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 2094–2100.

[32] T. Lan *et al.*, "AI-based autonomous line flow control via topology adjustment for maximizing time-series ATCs," in *Proc. IEEE Power Energy Soc. General Meeting (PESGM)*, 2020, pp. 1–5.

[33] "123-Bus Feeder." [Online]. Available: https://cmte.ieee.org/pes-testfeeders/resources/ (Accessed: Jan. 12, 2021).

Jin Zhao (Member, IEEE) received the B.E. and Ph.D. degrees in electrical engineering from Shandong University, Jinan, China, in 2015 and 2020, respectively.

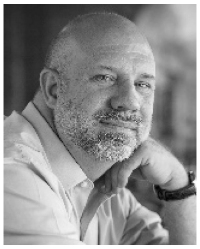
She is currently a Research Scientist with the University of Tennessee, Knoxville, TN, USA. She was a Research Assistant with the University of Hong Kong (HKU), China. She was also a visiting Ph.D. student and a Postdoctoral Researcher with the Technical University of Denmark (DTU), Denmark. Her research interests include power system resilience, transmission and distribution system restoration, renewable energy integration, power system optimization, microgrid and machine-learning. She is a Outstanding Reviewer of IEEE TRANSACTIONS ON POWER SYSTEM in 2021 and OPEN ACCESS JOURNAL OF POWER AND ENERGY 2021.

Fangxing Li (Fellow, IEEE) is also known as Fran Li. He received the B.S.E.E. and M.S.E.E. degrees from Southeast University, Nanjing, China, in 1994 and 1997, respectively, and the Ph.D. degree from Virginia Tech, Blacksburg, VA, USA, in 2001.

He is currently the James W. McConnell Professor of Electrical Engineering and the Campus Director of CURENT, University of Tennessee, Knoxville, TN, USA. His current research interests include renewable energy integration, demand response, distributed generation and microgrid, energy markets, and power system computing. He has received a number of awards and honors, including the Research and Development 100 Award in 2020, the IEEE PES Technical Committee Prize Paper Award in 2019, three best paper awards at international journals, and six best papers/posters at international conferences. From 2020 to 2021, he served as the Chair of IEEE Power System Operation, Planning and Economics Committee. Since 2020, he has been serving as the Editor-in-Chief of IEEE OPEN ACCESS JOURNAL OF POWER AND ENERGY.

Srijib Mukherjee (Senior Member, IEEE) received the B.E. degree in electrical power engineering from Manipal Institute Technology, Manipal, India, in 1989, and the M.S.E.E. and Ph.D. degrees in electrical engineering from the University of Miami, Coral Gables, FL, USA, in 1991 and 1994, respectively, and the M.B.A. degree in finance from the Duke University's Fuqua School of Business in 2003.

He is currently the Senior Staff Research and Development Member of Oak Ridge National Laboratory's Power System Resilience Group with a Joint Faculty appointment with the University of Tennessee's Bredesen Center. His current research interests include renewable energy integration, bulk power operations, energy storage, distributed generation and microgrid, energy markets, and power system computing. He has received several awards and honors, including the IEEE Outstanding Engineer Award in 2001 and the IEEE PICA System Restoration Contest Winner in 1999. From 2001 to 2003, he was the Past Chair of the IEEE PES and IAS, Eastern North Carolina chapter. He is also a Registered and Licensed Professional Engineer with the states of North Carolina, California, and Nevada. Additionally, he is an NERC Certified Grid Operator. He is a Past Education Board Member of NC State University's FREEDM Center and a distinguished alumnus at his alma mater.



Christopher Sticht (Senior Member, IEEE) received the B.S.E.E. degree with a concentration in power systems from Georgia Tech and the M.S.E.E. degree with a concentration in power systems from the University of Washington.

He is a Consulting and Research and Development Engineer. His work focuses on electric power systems, mathematics programming, and machine learning. He invented ATILDA, a revolutionary technique for load forecasting, weather adjustment, and load growth algorithms. He is an Authority in utility system planning, load analysis, planning software, renewables integration, and smart grid. During his tenure in the electric power industry, he has been involved with numerous aspects of power delivery. His experience includes government research, consulting, contracting, work at two power flow software companies, and at three major U.S. utilities, as well as contracting/consulting with numerous other utilities.