Deep Reinforcement Learning-Aided Pre-Positioning of Mobile Wind Turbines to Enhance Power Distribution System Resilience

Ruotan Zhang*, Jinshun Su†, Payman Dehghanian‡
Department of Electrical and Computer Engineering
The George Washington University
Washington, DC, USA.
{*zhangruotan114, †jsu66, ‡payman}@gwu.edu

Mohannad Alhazmi
Department of Electrical Engineering
King Saud University
Riyadh, Saudi Arabia.
mohalhazmi@ksu.edu.sa

***Abstract*—Compared to stationary wind turbines, mobile wind turbines (MWTs) can move within the transportation system and supply power to electrical microgrids. With the goal to enhance the power distribution system (PDS) resilience, MWTs can help in service restoration and supply power to islands that are separated from the main grid during emergencies. However, MWTs typically start to operate only after faults driven by high-impact low-probability (HILP) events occur. To improve service restoration efficiency with MWTs, we propose a framework based on deep reinforcement learning (DRL) for MWT pre-positioning. This framework uses the DRL agent's training rewards to determine the pre-position of MWTs before HILP emergencies arise. Since the MWT movement is a discrete decision, we applied the Deep Q-learning (DQL) and Double Deep Q-learning (DDQL) algorithms as the DRL agent models. The agent provides MWTs actions on an integrated system comprising an 11-node transportation system (TS) and a 33-bus PDS.**

***Index Terms*—Mobile wind turbine (MWT), pre-positioning, resilience, deep reinforcement learning (DRL), Deep Q-learning (DQL), power distribution system (PDS).**

I. Introduction

In recent years, high-impact low-probability (HILP) incidents such as hurricanes, floods, wildfires, and winter storm have been evidenced to cause widespread, prolonged power outages, significant equipment damage, and severe economic losses [1]. From 1980 to 2023, the frequency of billion-dollar disasters has steadily increased, with many of them resulting in widespread power outages [2]. Power outages driven by extreme weather not only caused significant economic loss but also has threatened human lives [3]. In 2021, a historic winter storm impacted many northwest, central, and eastern states in the US, leading to widespread power outages and the deaths of 226 people due to a shortage of energy for heating [4]. In response to such threatening events, enhancing service resilience has become a primary goal of the energy community.

With the ability to supply power to essential loads and help reduce economic losses, utilizing mobile power sources (MPSs) for service restoration can be crucial [5]. Due to their significant mobility and flexibility advantages over stationary energy sources, MPSs can move to isolated areas and supply power to aid in faster service restoration [6]. For instance, the study in [7] proposes a co-optimization approach for integrating MPSs and dispatching repair crews, formulated as a mixed-integer second-order cone programming model to enhance power distribution systems (PDSs) resilience. The study in [8] proposes a two-stage restoration scheme for PDSs, leveraging the full potential of MPS dispatch alongside dynamic distribution system reconfiguration under various seismic force scenarios. The study in [9] accounts for decision-dependent uncertainty in the availability of MPSs, considering travel and waiting times to provide a more realistic estimation of their contributions to enhancing PDS resilience. By strategically deploying MPSs, the study in [10] proposes a risk-averse model that generates a public-safety power-shutoff plan to balance wildfire risks and power outages.

However, available research in the existing literature [5]–[10] focuses on MPSs that rely on traditional energy sources, such as fossil fuels, which generate greenhouse gas emissions and are harmful to the environment. The development of sustainable energy systems is considered essential to mitigate the impacts of climate change-driven HILP incidents [11]. To address environmental problems, sustainable energy resources are widely integrated into power systems [12]. Mobile wind turbines (MWTs) are small-scale, easily-portable wind turbines commonly used for off-grid power generation or supplying power to remote locations [13]. The study [14] incorporates the joint use of MWTs and electric thermal storage into the microgrids energy portfolio, enabling load profile shifting and preventing additional costs associated with peak demand. In reference [15], MWTs combined with hydrogen storage units are applied in microgrids to minimize expected power outage costs and reduce carbon emissions.

With the advancement of artificial intelligence, machine learning methods are increasingly utilized to enhance the service resilience in power systems [16]. Unlike classical optimization methods, deep reinforcement learning (DRL) does not require a precise objective function and can handle higher-dimensional data than convex optimization methods [17]. In addition, DRL can make decisions based on the state of the current time step, facilitating real-time and online

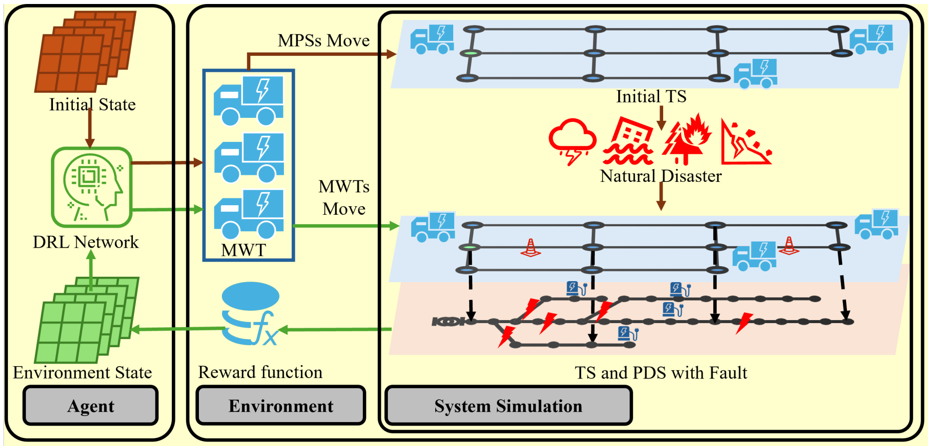


Fig. 1. The proposed DRL-based MWT pre-positioning framework for PDS restoration.

decision-making [18]. With proven advantages, the DRL was also applied to identify the optimal strategy for restoring the PDS with MPSs. In [19], a single-agent DRL approach was introduced to optimize the dispatch decisions of MPSs for the restoration of critical loads, considering the uncertainties in electrical consumption. To simplify the complex actions of different repair crews and various types of MPSs during service restoration, the authors in [20] applied a multi-agent DRL (MADRL) approach to reduce the action dimension for each agent. To integrate the discrete MPS action of routing with the continuous MPS action of charge/discharge schedules, reference [21] employs different agents within a MADRL framework, which addresses the hybrid discrete-continuous action space for MPSs and derives the restoration scheme.

In previous research on service restoration policies for MPSs based on DRL, as referenced in [19]–[21], all MPSs were stationed at the same depot and began moving only after the disaster occurs. However, a research gap is found in pre-positioning MWTs using DRL. The DRL is better suited for natural signals without a clear mathematical formula, as the actual data distribution is analytically unknown [22]. For MWTs, wind power generation depends on the local, unpredictable wind speed. Therefore, applying DRL to pre-position MWTs is crucial for identifying suitable positions for allocating MWTs. To address the issue of MWTs pre-positioning, we developed a framework for pre-positioning of MWTs based on a single-agent DRL algorithm. We use episode rewards of DRL training to evaluate the pre-positioning of MWTs. This framework is validated in an environment consisting of an 11-node transportation system and an IEEE 33-bus test system.

The remainder of the paper is organized as follows: Section II presents the MWT pre-positioning framework and the DRL algorithm. Section III discusses the training results and compares the DRL model with and without pre-positioning decisions. Section IV summarizes the research findings.

II. Problem Description

A. DRL-Based Framework

The general idea of the proposed MWT pre-positioning framework is depicted in Fig. 1. The framework comprises two main components: the reinforcement learning agent and the environment. The environment is responsible for managing MWT actions (including movement and power delivery), simulating the system (involving the PDS and the transportation system (TS)), and evaluating actions based on system information and the reward function. The DRL agent comprises a DRL network that determines the actions of the MWTs based on the state, which includes environment information. The agent is updated with the reward for the action that it provided. In our framework, the agent must provide two types of actions: one for the pre-positioning of the MWTs, depicted by the brown arrow in the framework, and another for the MWTs restoration of the PDS, depicted by the green arrow in the framework. Initially, the agent needs to provide the pre-positioning action before a natural disaster occurs. The agent receives the initial state of the environment without faults and provides the pre-positioning information to the MWTs. The environment then moves the MWTs to the pre-positioned locations and simulates the faults caused by the natural disaster in the TS and PDS. Once the disaster occurs, MWTs begin to operate. The environment provides information on the MWTs and the system as the state and calculates the reward for the agent. The agent updates and provides restoration actions to MWTs, guiding the restoration work at each time step. The loop of MWT restoration actions, represented by the green arrow in the framework, continues until the restoration process is complete. At each time step, the framework provides a reward, and the total reward accumulated throughout the process is used to evaluate the pre-positioning action. In the context of DRL training, each process is considered one episode, with the occurrence of faults due to natural disasters being randomly generated in each episode.

B. DRL Algorithm

The dispatch of MWTs for service restoration focuses solely on the destination node for each MWT as they route through the transportation system, making it possible to model this process as a Markov decision process (MDP) with a discrete action space. The MDP is fundamental in RL and is defined as a four-dimensional tuple, which is used as the experience for the DRL learning process. The MDP tuple is (S, A, R, P), in which [23]:

- S represents the environment state, with s_t indicating the state of the environment at time slot t.
- A denotes the action taken by the agent, with a_t representing the agent's action at time t corresponding to the state s_t.
- R represents the reward received by the agent for its action in a given state, with r_t indicating the reward obtained by the agent when taking action a_t in state s_t.
- P denotes the transient stability function, which evaluates the stability of state transitions using s_t, a_t, and s_{t+1}. $P(s_t, a_t)$ represents the transition where the agent takes action a_t in state s_t, resulting in a transfer to state s_{t+1}.

In our framework, we utilize a single agent to provide actions for all MWTs. This agent is a DRL model, allowing it to learn from experience, which consists of the MDP tuple, directly from the environment through exploration and exploitation [24]. For the DRL model in this paper, we employed Deep Q-Learning (DQL) and Double Deep Q-learning (DDQL) as the agent model. In DQL, the agent calculates a Q-value using S, A, and R, representing the expected reward when the agent chooses action A in state S. However, the DQL architecture always selects the action that maximizes the Q-value, which can result in an overestimation in the training process. To address this issue, DDQL employs a different structure by using two neural networks: one network selects the action, while the other evaluates the selected action. The details of DQL and the DDQL are provided in Algorithm 1:

Algorithm 1 DQL & DDQL Algorithm for MWT Agent

1: Initialize replay memory D
2: Initialize action-value function $Q(s, a; \theta)$ and target function $\hat{Q}$ with random weights $\theta^- = \theta$
3: **for** episode $= 1, M$ **do**
4:     Initialize state s_t
5:     Set fault and input to the environment
6:     **for** time step $t = 1, T$ **do**
7:         **if** $t = 1$ **then**
8:             Generate pre-position of MWTs as a_t
9:         **else if** $t \neq 0$ **then**
10:             Trigger the fault
11:             Generate MWTs restoration action for as a_t
12:         **end if**
13:         Getting new State information s'_t and reward r_t
14:         Save (s_t, a_t, r_t, s'_t) into D
15:         Randomly samples (s_j, a_j, r_j, s'_j) from D
16:         Set $y_j = r_j$ if episode terminates at Step $j + 1$
17:         Otherwise
18:         **if** DQL **then**
19:             $y_j = r_j + \gamma \max_{a'} \hat{Q}(s'_j, a'; \theta^-)$
20:         **else if** DDQL **then**
21:             $y_j = r_j + \gamma \hat{Q}(s'_j, \text{argmax}(Q(s'_j, a')))$
22:         **end if**
23:         $L_i(\theta_i) = (y_j - Q(s_j, a_j; \theta))^2$
24:         Update θ with loss function
25:         Update the state $s = s'$
26:     **end for**
27: **end for**

In line 1, the replay memory space D is created to store the experiences from training. In line 2, two Q-networks are established. One Q-network is used to determine the action for MWTs, while the other, known as the target network, is used to evaluate the actions generated by the first Q-network. From line 3, the loop of the training episode starts. In lines 4 and 5, the state s_t should be initialized and the fault information in the current episode is generated. From line 6, the episode starts. From line 7 to 11, the DRL agent provides the actions to MWTs for pre-positioning and restoration action a_t for the current time step based on the state information s_t. In line 13, the environment will change due to a_t, giving the new state s'_t and reward r_t. In line 14, the tuple (s_t, a_t, r_t, s'_t) will be saved in the memory D as the experience. Once the experience is sufficiently accumulated, from the line 15 to 25, the model starts training by using the loss function to update the weights of the Q-networks. In line 15, the agent randomly selects an experience sample from replay memory D and calculates the target value y_i based on the current time step. If the episode has ended, the target value y_i equals the reward value for the last time step. If the agent uses DQL, the target y_i is calculated using the single Q-value from the target Q-network and the reward value. Conversely, if the agent uses DDQL, the target y_i is determined by selecting the lower value from the two Q-networks and calculating y_i with the reward value. Then, in line 23, the agent computes the loss function based on the target value and the Q-value. In line 24, the agent updates θ based on the loss function. In line 25, the state information is updated as the next input for the agent.

III. Numerical Case Studies

A. DRL Training Environment

The DRL environment simulates the problem that needs to be solved, allowing the agent to make decisions based on the current state of the environment. The environment of the MWT pre-positioning consists of three parts: (i) a system simulation of the TS and PDS; (ii) a wind speed generation function and MWTs output power estimation; (iii) an action reward function that calculates rewards based on the power supplied by the MWTs and the load cost at PDS load points.

1) The Integrated Power-Transport Network: In this paper, the natural disaster causes damage not only to the PDS but also to the TS, requiring MWTs to move within the available TS and aid in restoration on the PDS. Therefore, establishing an integrated Power-Transport network is essential for the DRL environment. The TS and PDS are illustrated in Fig. 2. In this paper, we use the IEEE 33-bus test system [25] as the PDS coupled with an 11-node transportation system [20]. Each node in the TS is coupled with a bus in the PDS. The fault configurations within the PDS involve randomly selecting several power distribution lines to fail at different times during

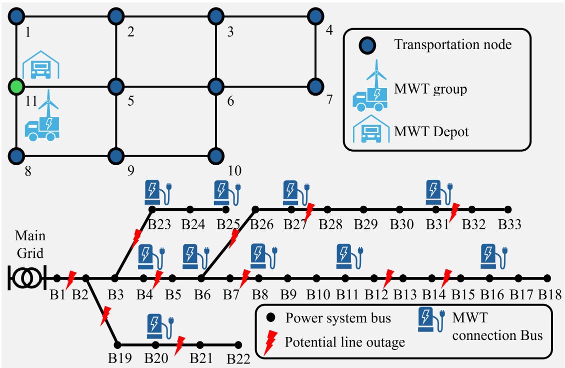


Fig. 2. The integrated power-transport network of the DRL environment

the process. This results in the segmentation of the PDS into multiple disconnected islands. The TS fault is modeled as road damage caused by a disaster. Throughout the simulation, these damaged roads will not be repaired, preserving the existing road conditions to restrict MWT transport. In each case, up to three roads in the TS will be randomly selected to be unavailable during the disaster. To provide PDS and TS information to the DRL agent's input state, we use the number of damaged power lines and the power supply status of the load points to represent the PDS information. For TS information, we use the MWTs' locations and the shortest path matrix of the TS, considering road damage.

2) MWTs Power Generation: In our paper, we applied three MWTs in the framework. The power output of a wind turbine follows a truncated cubic relationship with wind speed [26]. We set the cut in speed to 0 mph. The MWT's power output can be expressed as follows:

$$P = \begin{cases} \frac{v^3}{v_{max}^3} P_{max} & 0 < v < v_{max} \\ P_{max} & v > v_{max} \\ 0 & v > v_{out} \end{cases}$$

v_{max} represents the rated wind speed of the wind turbine, and P_{max} denotes its rated power output. v_{out} indicates the cut-out wind speed of the turbine, while v refers to the wind speed at the current time step. In this study, the rated wind power output, P_{max}, for each MWT group is set at 1,000 kW. The wind speed, v, varies between 30 and 40 mph and follows a Weibull distribution that is randomly generated for each traffic node at every time step. The v_{max} is 45 mph, and the v_{out} is 60 mph. The MWTs' power information includes two parts: the status of the MWT (whether it is supplying power to the PDS or not) and the wind power generation. Both pieces of information will be part of the agent's input state.

3) Reward Function: The reward for each timestep depends on the cost of loads supplied by the MWTs and the wind power generation of the MWTs. To minimize the wind spillage, if the MWTs' generated power exceeds the load demand on the same island, the wasted wind power will incur a penalty based on the average load cost of the island's power demand. The reward equation for each MWTs is:

$$R_{MWT} = c_{load} \frac{p_{MWT}}{p_{total} N_f} - c_{avg}(p_{total} - P_{load}) \frac{p_{MWT}}{p_{total} N_f} \quad (1)$$

R_{MWT} represents the reward value for each agent, where c_{load} denotes the cost of the load supplied by the MWT in the current island of the PDS. c_{avg} is the average load cost of the island. p_{MWT} signifies the wind power output of MWTs, while p_{total} indicates the total power supplied by MWTs in the current section of the PDS. N_f represents the transfer coefficient, which reduces the magnitude of the reward. The agent's reward is the sum of the rewards from the three MWTs at each time step. Each training episode consists of 24 timesteps, with each timestep representing half an hour. The evaluation of the MWTs pre-positioning is based on the total reward accumulated over all timesteps in a single episode.

B. The Deep Learning Network and the Action in DRL

The MWTs actions consist of selecting a destination node in the TS, giving the agent a discrete action space. In this framework, we use a Convolutional Neural Network (CNN) as the deep learning model for the DRL to perform the classification tasks. The size of the action space corresponds to the number of output labels for the CNN. In each time step, the CNN generates a label that is used to stand for a specific action combination. The action combination consists of the target transportation node in TS for each MWT. In the integrated power-transport network, we have three identical MWTs and ten transportation nodes capable of supplying power to the PDS. Thus, excluding combinations with the same MWT positions but different vehicle positions, the number of output labels for this convolutional network should be 120 (i.e., the action space size is 120). The input state will be reshaped to a group of $12x12$ matrix. and the structure of the CNN is: **Input(3, 12×12) – Conv1(64, 10×10) – Pool1(64, 5×5) – Conv2(128, 4×4) – Pool2(128, 2×2) – FC1((64×2)×2×2, 144) – FC2(144, 72) – FC2(72, 120)**. The learning rate η is 1e-4, and the minimum exploration rate ϵ is 5e-4.

C. Training for MWT Pre-positioning

To compare the agent models base on DQL and DDQL, we conducted two separate training sessions with different models. Since the pre-positioning should be applicable to various fault settings in the PDS and TS, we use the moving reward, calculated from the latest 500 episode rewards, to evaluate the performance of the agent model. Figure 3 shows the moving reward curve for the DQL and DDQL models, as well as the difference between the actual episode reward and the moving reward. During the training, we applied the ϵ-greedy method to make the agent do exploration and exploitation and get experience [23].

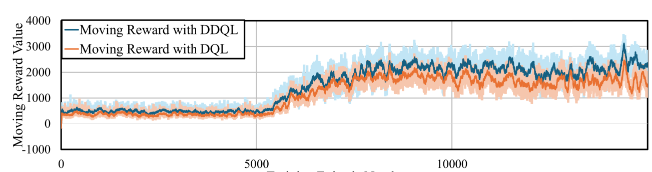
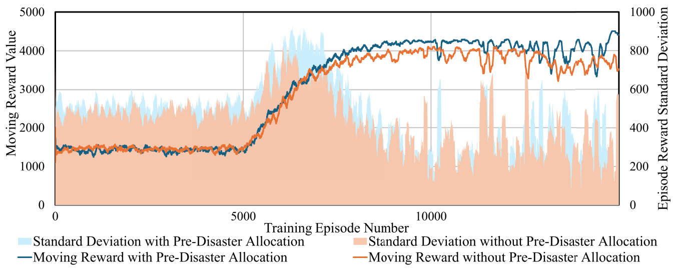
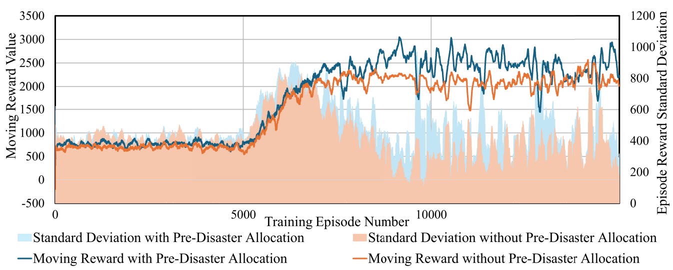


Fig. 3. Training moving reward for different networks.

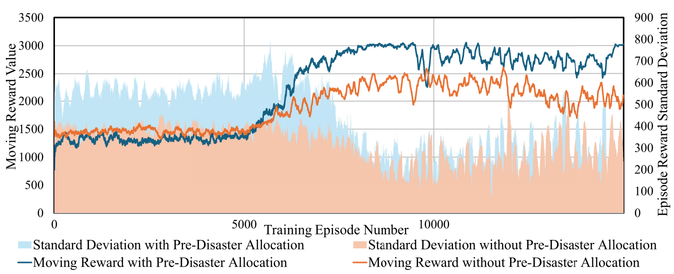
At the start of training, both agents take random actions, resulting in poor performance for both, with no significant differences between the two moving reward curves. After 5000 episodes, as the ϵ value decreases, both agents begin to exploit more, leveraging the experience gained from previous actions at each timestep. The initial moving reward of the agent model using the DQL algorithm is lower than that of the agent model using the DDQL algorithm. This difference is caused by the overestimate of the DQL algorithm.



(a) Case 1



(b) Case 2



(c) Case 3

Fig. 4. The moving reward curve of training in different cases.

Since the damage caused by natural disasters is unpredictable, the PDS and TS remain undamaged, and all MWTs are parked at the depot before the pre-positioning command. As a result, the input state of the agent cannot provide sufficient information for MWTs to move to specific locations based on the specific damage information in the PDS and TS in each episode. Therefore, in each episode, the agent will provide actions for the first timestep based on the same state input matrix. For the agent model with DQL, due to overestimation, the agent will provide pre-positioning for MWTs that yield high episode rewards in certain fault settings in the PDS and TS. This is because these actions have higher Q values in the Q table, leading the agent to believe they have the best expectations while ignoring some poor reward values associated with the same MWTs' pre-position actions. This will lead the agent to provide suboptimal actions. However, with the DDQL agent model having separate networks for estimation and action selection, the agent does not solely rely on actions with high Q values from the action network. This separation encourages the agent to explore a wider range of actions, rising the likelihood of discovering the optimal action.

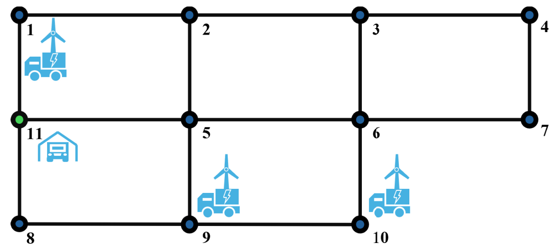


Fig. 6. The pre-positioning decisions of MWTs in the TS.

The pre-positions of the three MWTs are shown in Fig. 6. The selected locations in the TS are Node 1, Node 9, and Node 10. However, as illustrated in Fig. 3, there is a significant discrepancy between the real episode reward and the moving reward. This is because the damage in the PDS and TS is randomly generated in each episode, causing variations in faults that require changes in the MWTs' strategy, making the training difficult to converge. To test the pre-positions of the MWTs and ensure that the framework provides convergent results, we applied three cases with different fault settings in the PDS and TS.

To demonstrate the efficiency of MWT pre-positioning approach, we used two DDQL agent models on three different cases. In one model, the MWTs' starting points are the pre-positions shown in Fig. 6. In the other model, all MWTs start at the depot node of the TS. The training result is shown in Fig. 4. The three moving reward curves for training based on the three different cases show that the performance of the agent model with MWTs' pre-position is better than the agent model without MWTs' pre-position. Additionally, all standard deviations decrease to a lower level. However, due to the unpredictability of wind speed, the moving reward still exhibits fluctuations.

Figure 5 illustrates the restoration scenario based on the PDS and TS faults in case 3. R1 represents the restoration

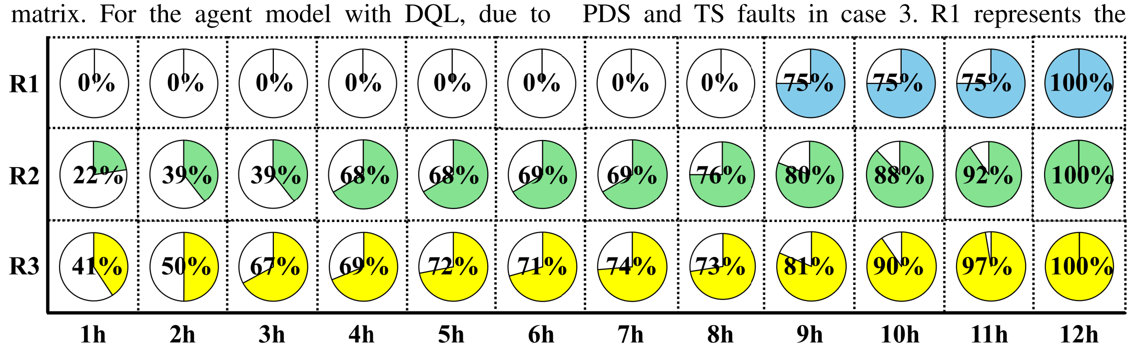


Fig. 5. Percentage of the total restored demand over time with the fault setting in Case 3.

TABLE I
COST SAVING WITH MWTS CONTRIBUTION

Case	Cost Saving ($)	Load Cost Saving Ratio (%)
R1	0	0
R2	17,604k	32.35
R3	22,517k	41.37

process without MWTs, R2 represents the restoration process with MWTs but without pre-positioning, and R3 represents the restoration process with both MWTs and pre-positioning decisions. It is evident that the MWTs help reduce the loss of the load in the PDS. Comparing the load restoration values over the first three hours, MWT pre-positioning significantly improves the efficiency of the restoration process as MWTs start supplying power earlier. From the fourth to the eleventh hour, the load restoration between R2 and R3 shows no significant difference. This difference is caused by the MWTs' power generation. Table I shows the load cost saved by MWT restoration. The Cost Saving refers to the load cost supplied by MWTs during the restoration process. The Load Cost Saving ratio is the proportion of the MWT restoration load cost relative to the total load cost of the PDS. MWT pre-positioning helps the PDS save 9.02% of the total load cost. Thus, MWT pre-positioning can enhance the efficiency of PDS restoration following natural disasters and reduce economic losses.

IV. CONCLUSION

In this paper, we present a novel restoration framework designed to enhance the resilience of PDSs impacted by natural disasters. The proposed DRL framework provides optimal pre-positions and discrete actions for MWTs, enabling effective scheduling and restoration strategies for PDSs using MWTs, as well as ensuring power supply to loads isolated from the main grid. This framework is tested on an integrated power-transport network with one IEEE 33-bus test system and an 11-node TS. The training results demonstrated that the proposed framework enhances system resilience by determining pre-positions and movement actions for MWTs in anticipation of emergencies.

V. ACKNOWLEDGMENT

This work was supported in part by the U.S. National Science Foundation (NSF) under Grants RISE-2220626 and ECCS-2114100.

REFERENCES

[1] J. Su, P. Dehghanian, M. Nazemi, and B. Wang, "Distributed wind power resources for enhanced power grid resilience," in *2019 North American Power Symposium (NAPS)*. IEEE, 2019, pp. 1–6.

[2] A. B. Smith, "2023: A historic year of U.S. billion-dollar weather and climate disasters," 2023, available at https://www.climate.gov/news-features/blogs/beyond-data/2023-historic-year-us-billion-dollar-weather-and-climate-disasters.

[3] D. Anokhin, P. Dehghanian, M. A. Lejeune, and J. Su, "Mobility-as-a-service for resilience delivery in power distribution systems," *Production and Operations Management*, vol. 30, no. 8, pp. 2492–2521, 2021.

[4] J. W. Busby, K. Baker, M. D. Bazilian, A. Q. Gilbert, E. Grubert, V. Rai, J. D. Rhodes, S. Shidore, C. A. Smith, and M. E. Webber, "Cascading risks: Understanding the 2021 winter blackout in texas," *Energy Research & Social Science*, vol. 77, p. 102106, 2021.

[5] S. Lei, C. Chen, H. Zhou, and Y. Hou, "Routing and scheduling of mobile power sources for distribution system resilience enhancement," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5650–5662, 2019.

[6] M. Nazemi, P. Dehghanian, X. Lu, and C. Chen, "Uncertainty-aware deployment of mobile energy storage systems for distribution grid resilience," *IEEE Trans Smart Grid*, vol. 12, no. 4, pp. 3200–3214, 2021.

[7] S. Lei, C. Chen, Y. Li, and Y. Hou, "Resilient disaster recovery logistics of distribution systems: Co-optimize service restoration with repair crew and mobile power source dispatch," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6187–6202, 2019.

[8] Z. Yang, P. Dehghanian, and M. Nazemi, "Seismic-resilient electric power distribution systems: Harnessing the mobility of power sources," *IEEE Trans. Ind. Appl.*, vol. 56, no. 3, pp. 2304–2313, 2020.

[9] J. Su, D. Anokhin, P. Dehghanian, and M. A. Lejeune, "On the use of mobile power sources in distribution networks under endogenous uncertainty," *IEEE Trans. Control*, vol. 10, no. 4, pp. 1937–1949, 2023.

[10] J. Su, S. Mehrani, P. Dehghanian, and M. A. Lejeune, "Quasi second-order stochastic dominance model for balancing wildfire risks and power outages due to proactive public safety de-energizations," *IEEE Transactions on Power Systems*, vol. 39, no. 2, pp. 2528–2542, 2024.

[11] N. Shi, R. Cheng, L. Liu, Z. Wang, Q. Zhang, and M. J. Reno, "Data-driven affinely adjustable robust volt/var control," *IEEE Transactions on Smart Grid*, vol. 15, no. 1, pp. 247–259, 2024.

[12] J. Su, R. Zhang, P. Dehghanian, and M. H. Kapourchali, "Pre-disaster allocation of mobile renewable-powered resilience-delivery sources in power distribution networks," in *2023 North American Power Symposium (NAPS)*, 2023, pp. 1–6.

[13] J. Su and P. Dehghanian, "Post-disaster dispatch of transportable wind turbines for enhancing resilience of power distribution systems," in *2024 IEEE Texas Power and Energy Conference (TPEC)*, 2024, pp. 1–6.

[14] J. Su, P. Dehghanian, B. Vergara, and M. H. Kapourchali, "An energy management system for joint operation of small-scale wind turbines and electric thermal storage in isolated microgrids," in *2021 North American Power Symposium (NAPS)*, 2021, pp. 1–6.

[15] J. Su, R. Zhang, P. Dehghanian, M. H. Kapourchali, S. Choi, and Z. Ding, "Renewable-dominated mobility-as-a-service framework for resilience delivery in hydrogen-accommodated microgrids," *Int. J. Electr. Power Energy Syst.*, vol. 159, p. 110047, 2024.

[16] R. Zhang, Y. Li, M. Hijazi, J. Su, P. Dehghanian, and A. H. Etemadi, "Machine learning-aided enhancement of power grid resilience to electromagnetic pulse strikes," in *2022 North American Power Symposium (NAPS)*, 2022, pp. 1–6.

[17] S. Lee, J. Seon, Y. G. Sun, S. H. Kim, C. Kyeong, D. I. Kim, and J. Y. Kim, "Novel architecture of energy management systems based on deep reinforcement learning in microgrid," *IEEE Transactions on Smart Grid*, vol. 15, no. 2, pp. 1646–1658, 2024.

[18] B. Wang, Z. Zha, L. Zhang, L. Liu, and H. Fan, "Deep reinforcement learning-based security-constrained battery scheduling in home energy system," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 3548–3561, 2024.

[19] S. Yao, J. Gu, H. Zhang, P. Wang, X. Liu, and T. Zhao, "Resilient load restoration in microgrids considering mobile energy storage fleets: A deep reinforcement learning approach," in *2020 IEEE Power & Energy Society General Meeting (PESGM)*, 2020, pp. 1–5.

[20] Y. Wang, D. Qiu, F. Teng, and G. Strbac, "Towards microgrid resilience enhancement via mobile power sources and repair crews: A multi-agent reinforcement learning approach," *IEEE Transactions on Power Systems*, vol. 39, no. 1, pp. 1329–1345, 2024.

[21] Y. Wang, D. Qiu, and G. Strbac, "Multi-agent deep reinforcement learning for resilience-driven routing and scheduling of mobile energy storage systems," *Applied Energy*, vol. 310, p. 118575, 2022.

[22] Q. Wang and C. Tang, "Deep reinforcement learning for transportation network combinatorial optimization: A survey," *Knowledge-Based Systems*, vol. 233, p. 107526, 2021.

[23] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *Robotica*, vol. 17, no. 2, pp. 229–235, 1999.

[24] Y. Mei, H. Zhou, T. Lan, G. Venkataramani, and P. Wei, "MAC-PO: Multi-Agent Experience Replay via Collective Priority Optimization," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 2023, pp. 466–475.

[25] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Power Engineering Review*, vol. 9, no. 4, pp. 101–102, 1989.

[26] A. T. Abolude and W. Zhou, "Assessment and performance evaluation of a wind turbine power output," *Energies*, vol. 11, no. 8, p. 1992, 2018.