Post-Disaster Generation Dispatching for Enhanced Resilience: A Multi-Agent Deep Deterministic Policy Gradient Learning Approach

Michael Abdelmalak
Electrical & Biomedical Eng.
University of Nevada-Reno
Reno, NV, USA
mabdelmalak@nevada.unr.edu

Hadis Hosseinpour
Electrical & Biomedical Eng.
University of Nevada-Reno
Reno, NV, USA
hhosseinpour@unr.edu

Eliza Hotchkiss
Resilience System Design
National Renewable Energy Lab.
Golden, CO, USA
eliza.hotchkiss@nrel.gov

Mohammed Ben-Idris
Electrical & Biomedical Eng.
University of Nevada-Reno
Reno, NV, USA
mbenidris@unr.edu

***Abstract*—This paper proposes a reinforcement learning-based approach for dispatching distributed generators (DGs) to enhance operational resilience of electric distribution systems after a severe outage event. The increased computational complexities and sophisticated modeling procedure of resilience-based enhancement strategies have pushed toward adopting intelligent-based algorithms, specifically for real-time control applications. In this paper, a multi-agent deep deterministic policy gradient learning algorithm is developed to dispatch distributed generators after an extreme event. The proposed approach aims to provide a fast-acting control algorithm for improved resilient operation of islanded distribution power systems. The problem is formulated as an iterative Markov decision process that consists of a system state, action space, and reward function. Each agent is responsible for dispatching a single DG and is trained to increase its cumulative reward value. A system state represents the system topology and characteristics whereas an action refers to DG power supply. A reward is computed based on the power balance mismatch value for each agent. Different failure scenarios are generated and used to train the proposed model. The proposed method is demonstrated on the IEEE 33-node distribution feeder system in the islanded mode. The results show the capability of the proposed algorithm to dispatch DGs for resilience enhancement.**

***Index Terms*—Distribution system, extreme weather events, reinforcement learning, resilience.**

I. Introduction

Modern societies have been increasingly relying on electricity access and availability. The unavailability of electricity due to extreme weather-related events results in significant economic losses and noticeable social harm [1], [2]. During the last seven years, the United States has been exposed to seven wildfires, eight droughts, 75 severe storms, 19 tropical cyclones, 16 floods, five winter storms, and one freeze event with more than one billion-dollar anticipated costs [3]. For example, 4.5 million customers in Texas experienced power outages with an outage time of more than 105 hours due to Winter Storm Uri in February 2021 [4]. These severe events can cause catastrophic impacts on power system equipment, yielding prolonged power outages [5]. Developing fast and efficient operation enhancement strategies and policies for electric distribution systems will be needed to improve the resilience of the power supply [6], [7].

Distributed generators (DGs) provide a potential pathway to improve the load restoration behavior during and after an extreme event [8]. However, determining proper dispatching decisions is a challenging burden due to the tight operational conditions, specifically for large-scale systems. Also, the stochastic behavior of component failures due to an extreme event induces additional complexities for proper generation dispatching. Therefore, implementing a corrective and restorative resilience enhancement strategy that leverages DGs after an extreme event has become important.

Several studies have been conducted for resilience enhancements of distribution power systems via corrective and restorative approaches. This includes microgrid formation [9], network reconfiguration [10], and utilization of DGs [11]. A spectral clustering algorithm has been employed to determine optimal network partitions under tight potential $N-k$ (i.e., $k>1$) contingencies [12]. In [13], we have developed a deterministic proactive generation redispatch strategy to improve the resilience of transmission power systems against hurricanes. Also, we have studied the impact of probabilistic spatiotemporal characteristics of a wildfire event on generation dispatching in [14]. Authors of [15] have developed an integrated framework between unit commitment scheduling and routing of mobile DC de-icing devices during an ice storm. Also, a sequential proactive strategy has been studied in [16] for enhanced resilience. In [7], a proactive microgrid management strategy to control existing DGs has been provided.

Despite the significant contributions of these methods to improve power system resilience, their efficacy depends mainly on the accuracy of system models and the degree of approximations. Also, these methods rely mainly on analytical and optimization techniques, which impose scalability challenges due to the increased modeling and computational complexities. The capabilities of reinforcement learning (RL) approaches to overcome the aforementioned constraints are still under investigation.

978-1-6654-9921-7/22/$31.00 ©2022 IEEE

Deep reinforcement learning (DRL) has been used to provide a fast-acting control algorithm for high-dimension stochastic optimization problems [17], [18]. In [19], a multi-agent based DRL model has been developed to control the reactive power of shunt compensators for improved voltage resilience. Authors of [20] have developed an RL-based controller to make fast real-time decisions to dispatch DGs during a hurricane. The proposed framework has shown promising results to outperform classic optimization approaches in terms of operation costs and computation time. A multi-agent DRL approach has been developed in [21] to optimize the control operation of a microgrid after a disaster. A resilience-based protection scheme utilizing reinforcement learning has improved the efficiency of microgrid operation considering market participation and stochastic behavior of renewable energy sources [22]. In [23], a DRL approach is used for an optimal rescheduling plan of generators at transmission level impacted by hurricanes. RL approaches provide a potential pathway to reduce reliance on the sophisticated modeling procedure, specifically in the resilience enhancement domain. Since RL-based methods can be easily integrated into online decision-making process, they can learn from experiences during online operations [24]. The diverse learning methods have pushed toward deeper investigation of DRL methods in controlling and dispatching DGs for enhanced resilience after an extreme event.

This paper proposes a DRL-based approach to dispatch DGs for resilience enhancement after an extreme event. The proposed algorithm aims to reduce the amount of load curtailments through minimizing the power balance mismatch. A multi-agent framework is developed such that each agent controls a specific DG. A multi-agent Deep Deterministic Policy Gradient (MADDPG) approach is adopted to create and train the developed DRL model. Diverse failure scenarios are generated using multiple line outages. In the proposed method, the distribution power grid is assumed to be disconnected from the main feeder yielding islanded microgrids. The developed problem environment takes into consideration the split of the islanded feeder into smaller microgrids due to multiple line failures. A Markov Decision Process (MDP) is formulated to train the DDPG-based model for enhanced resilience. The proposed algorithm is tested on a modified version of the IEEE 33-node distribution feeder with arbitrarily allocated DGs. The proposed algorithm provides a corrective and restorative strategy to improve the resilience of distribution systems after an extreme event.

The rest of the paper is organized as follows. Section II describes the multi-agent deep deterministic policy gradient method. Section III explains the developed post-event dispatching strategy and formulates the MDP environment. Section IV illustrates the implementation procedures on a modified version of the IEEE 33-node distribution feeder and discusses the results. Section V provides concluding remarks.

II. Multi-Agent Deep Deterministic Policy Gradient Approach

MADDPG algorithm is an improved version of the DDPG algorithm for multi-task applications. In a multi-agent system, the agents are not only affected by the environment, but also by other agents where the critic is augmented with extra information about the policies of other agents. The return of a single agent in the multi-agent system is related to both its own actions and the actions of other agents. Markov games are often used to describe multi-agent systems. In MADDPG, a Markov game for N agents is defined by a set of states (S) describing the possible configurations of all agents, actions (a), and observations (o) for each agent. The control law for each agent with a Gaussian noise $\mathcal{N}$ can be expressed as follows.

$$a_i^t = \pi_i\big(o_i^t|\theta_i^\pi\big) \, + \mathcal{N}(0\,,\, \sigma_i^t) \tag{1}$$

where θ_i^π is the weight of the actor for agent i, and σ_i^t is a parameter for exploration. The discount accumulate reward of the i^{th} Actor is as follows.

$$J_i = E_{\mu_i}\, R_i \;, \quad R_i = \sum_{t=1}^{T} (\gamma^{t-1}\, r_i^t) \tag{2}$$

where μ_i is the policy network of the i^{th} Actor, γ is a discount factor, r_i^t is the reward obtained at time step t in an episode, and T is the time horizon. Updating actors using the sampled policy gradient of the (2) is given by,

$$\nabla_{\theta_i^\mu} J_i \;\approx\; \frac{1}{S} \sum_{j=1}^{S} \nabla_{\theta_i^\mu}\; \mu_i\; (o_i^j) \nabla_{a_i}\;\; Q_i^\mu(x^j,\, a^j)\, |_{a_i^j=\mu_i(o_i^j)} \tag{3}$$

where Q is the action-value function, x^j is state, and S is the sample number of a random mini-batch. o_i^j and a_i^j are the observation and action of the i^{th} Actor, ∇i = 1, 2, …, N, respectively. A Critic's primary task is to predict the discount accumulate reward based on the current observations and actions of all Actors. The i^{th} critic can be updated minimizing the following loss function,

$$L(\theta_i^Q) = \frac{1}{S}\; \sum_j (y^j - Q_i^\mu(x^j,\, a^j))^2 \tag{4}$$

$$\begin{aligned} y^j = r_i^j + \gamma\; Q_i^{\mu'}(x'^j, a'^j)|_{a'_k=\mu'_k(o_k'^j)}\;, \\ a_i'^j \in a'^j, o_i'^j \in x'^j \end{aligned} \tag{5}$$

$$x^j = \big[o_1^j,\, o_2^j,\, \ldots,\, o_N^j\big]\;,\;\; a^j = \big[a_1^j,\, a_2^j,\, \ldots,\, a_N^j\big] \tag{6}$$

$$x'^j = \big[o_1'^j,\, o_2'^j,\, \ldots,\, o_N'^j\big]\;,\;\; a'^j = \big[a_1'^j,\, a_2'^j,\, \ldots,\, a_N'^j\big] \tag{7}$$

$$\theta_i' \;\leftarrow\; \tau\theta_i + (1-\tau)\theta_i' \tag{8}$$

where $(.)'$ donates to the target for Q' and μ' and next for a' and o'. $\theta_i^{(.)}$ shows the weight of parameter. Equation (8) can be used to softly update target network parameters (Q' and μ') for each agent i that τ is a control parameter for updating the target networks. To train all agents, a replay buffer is used as follows.

$$\mathcal{D} \leftarrow \left(s_t, o_t^i, a_t, r_t, s_{t+1}, o_{t+1}, a_{t+1}, d\right) \tag{9}$$

Fig. 1.b shows the complete MADDPG framework with a soft update and random noise. The presented soft actor-critic algorithm is classified as an off-policy maximum entropy algorithm. The difference between the general actor-critic framework and the presented MADDPG framework is the introduction of the entropy of the actor outputs during the training phase, as shown in Fig. 1.

III. The Proposed MADDPG-Dispatch Algorithm

This section describes the proposed RL-based approach to dispatch DGs for resilience enhancement of distribution systems. First, it explains the MADDPG-dispatch environment and then it illustrates the training and execution algorithms.

A. MADDPG-dispatch environment

An MDP is used to formulate the problem where a system state represents specific system conditions. A transition to another state is due to taking certain actions yielding a reward that can be defined as a function of desired outcome. For a multi-agent framework, a system state is decomposed into observations equal to the number of agents. The components of the formulated MDP are defined below.

1) States:

A system state is defined to be the set of parameters that can be used to describe the system conditions and it includes required information to observe system characteristics under specific circumstances. The state set is defined as:

$$s_t = \{G_i^s, \Delta G_i^m, u_j, o_i^n\}, \forall n \in \Omega^N,\ \forall i \in \Omega^G,\ \forall j \in \Omega^B \tag{10}$$

where G_i^s is the DG power, ΔG_i^m is the DG power mismatch, u_j is the line status, o_i is the set of connected nodes to the i^{th} agent, Ω^N is the set of system nodes, Ω^G is the set of DGs, and Ω^B is the set of lines.

2) Actions:

It is required to determine the power supply of each DG to minimize the amount of power balance mismatch. In other words, the amount of power supplied by DGs shall be equal to the load demand within a specific grid. In the proposed problem, a continuous action representing the DG real power needs to be taken by each agent, as follows,

$$a_t^i = \{G_i^s\} \tag{11}$$

where a_t^i represents the action taken by the i^{th} agent.

3) Rewards:

A proper reward value, r_t^i, should be defined to assess the effectiveness of the taken actions. Each agent is responsible for controlling the power supply of a specific DG through reducing/eliminating the amount of load curtailment after an extreme event. This can be achieved by minimizing the amount of power balance mismatch—given sufficient availability of generation resources. The reward value increases as the absolute power mismatch approaches zero value. Due to multiple line failures, a system can split into one or more microgrids, M. Therefore, the set of DGs in a specific microgrid should supply sufficient generation for minimal power mismatch. The reward r_t^i for taking a specific action is calculated as:

$$r_t^i = G_i^s - \big[\sum_{n \in \Omega_m^N} L_n\big]/N_G^m \tag{12}$$

where L_n is the load demand of n^{th} node, N_G^m is the number of DGs in the m^{th} microgrid, and Ω_m^N is the set of all connected nodes in the m^{th} microgrid.

B. Training and Execution Algorithms

The training and testing/execution steps for the multi-agent framework are summarized in Algorithm 1 and Algorithm 2.

Algorithm 1 - Training of the MADDPG-dispatch Framework

for Episode = 1 to E_{train} **do**
  Create failure scenario
  Reset the environment to default settings
  Extract observations of all agents (o_t) using current state (s_t)
  while Constraints not fulfilled and step $< N$ **do**
    for i = 1 to N_{agents} **do**
      Generate an action (a_t^i) using (1)
    end for
    Append all actions
    Execute action (a_t) on the environment
    Obtain new state (s_{t+1}), new observations (o_{t+1}), reward (r_t), and terminal conditions (d).
    Store (s_t, o_t, a_t, r_t, s_{t+1}, o_{t+1}, d) using (9)
    if size(Memory) $\geq$ batch size **then**
      Randomly select minibatch
      Update weights of the policies using (3)
      Update the Q-function parameters of each agent using (4)
      Update temperature of networks using (8)
      Update target network weights of each agent using (8)
    else if d is true **then**
      Reset the environment
    end if
  end while
end for

IV. Implementation and Results

The proposed approach is applied on the 33-node distribution feeder for validation. The proposed MADDPG model is formulated to dispatch DGs connected to distribution feeder for enhanced resilience after extreme outage events.

A. System under study

The 33-node distribution test system is a radial distribution system with 33 nodes and 32 branches with a total system

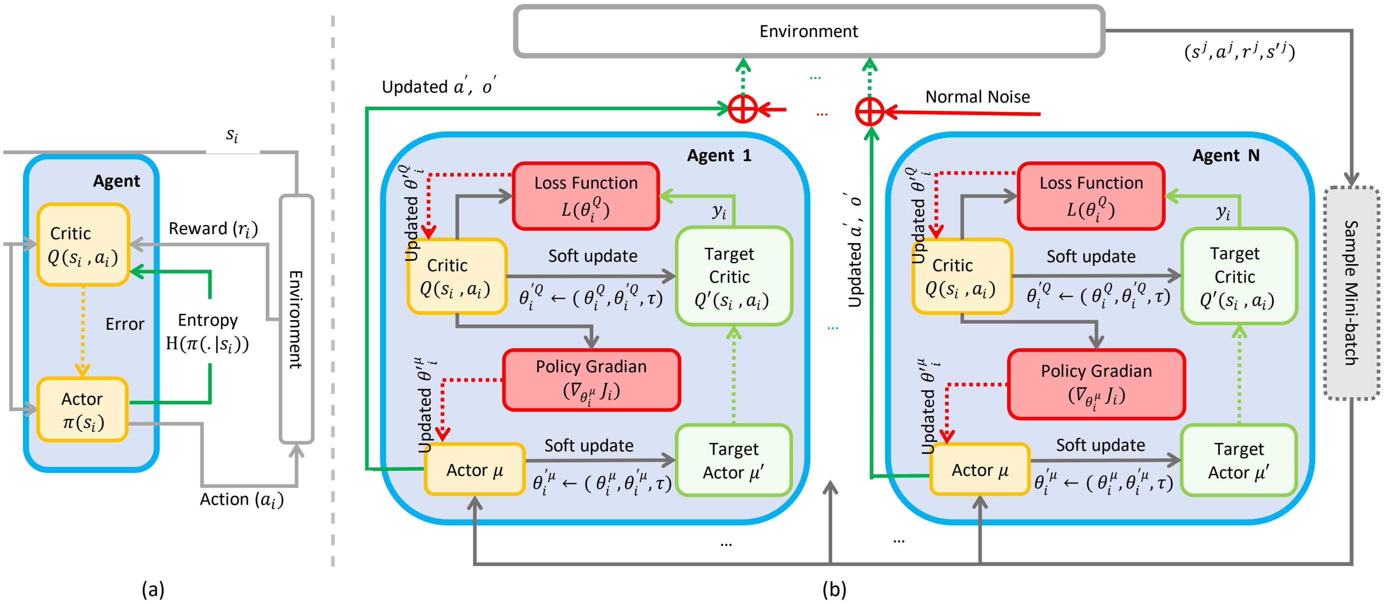

Fig. 1. (a) A general actor critic framework (the green path shows the difference of soft actor critic framework with entropy term). (b) The MADDPG framework.

Algorithm 2 - Testing of the MADDPG-dispatch Framework

for episode = 1 to E_{test} **do**
  Create failure scenario
  Reset the environment to default settings
  for i = 1 to N_{agents} **do**
    Generate an action (a_t^i) using actor network
  end for
  Execute action (a_t) on the environment
  Observe s_{t+1}, r_t, and d
end for

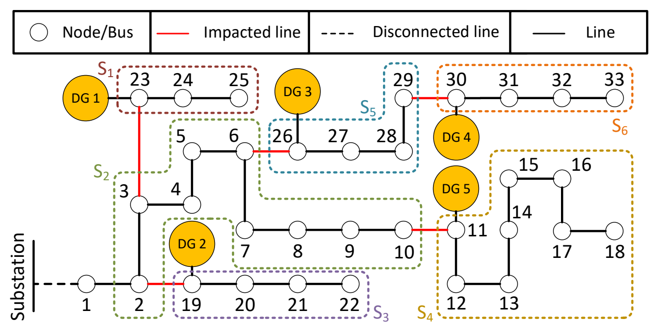

Fig. 2. IEEE 33-bus distribution feeder

TABLE I
LIST OF POTENTIAL MICROGRIDS

Index	Connecting nodes	Power (kW)
S_1	23, 24, 25	930
S_2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	855
S_3	19, 20, 21, 22	360
S_4	11, 12, 13, 14, 15, 16, 17, 18	555
S_5	26, 27, 28, 28, 29	400
S_6	30, 31, 32, 33	620

load of 3.72 MW [25]. Five DGs are connected to the feeder at arbitrarily chosen locations, as shown in Fig. 2, where each DG is represented by a single agent. Although the locations of DGs play a vital role to improve the resilience of the system, this work focuses on leveraging RL-based approaches to control predefined DGs after an extreme event. The proposed algorithm can be applied on any set of potential line failures; however, the list of vulnerable lines, in this work, includes (2–19), (3–23), (6–26), (29–30), and (10–11), as shown in Fig. 2. To induce further operating conditions, the connection to the main feeder is disconnected, which islands the feeder from the system and acts as an islanded microgrid. The failure of any vulnerable line results in splitting the main feeder into smaller microgrids. The list of all possible microgrids is summarized in Table I with their corresponding load demand.

B. Training

The proposed MADDPG algorithm is implemented for a fixed number of episodes (failure scenarios). A total of 20,000 episodes are used for training with a maximum of 20 iterations per episode. In each failure scenario, single or multiple lines are selected from the vulnerable lines to be disconnected. The hyper-parameter settings of the actor and critic networks of the proposed framework are shown in Table II. The recursive MDP process provided in algorithm 1 is used to train the MADDPG model. For evaluation, the running mean of the episodic rewards and the number of iterations per episode are calculated using a window of 100 episodes.

Fig. 3 shows the running mean and number of iterations per episode for the MADDPG model. The average reward value increases as the number of training episodes increases, as anticipated. The average reward value reaches a saturation

TABLE II
HYPER-PARAMETER SETTINGS OF THE MADDPG FOR 33-NODE SYSTEM

Hyper-parameter	Value
Number of hidden layers	3
No. of neurons in hidden layers	64
Learning rate	10^{-3}
Learning episodes	every 10
Temperature rate (τ)	0.01
Reward discount factor	0.99
Batch size	512
Activation function of output layer	Sigmoid
Activation function of hidden layers	ReLU
Optimizer	Adam

level in less than 8,000 episodes. At 5,000 episodes, a rapid increase in the reward value is noticed due to the decentralized structure of the MADDPG, where each agent is trained independently. Also, the random selection of mini-batch plays a vital role to provide a set of scenarios where an exploratory feature is achieved. On the other hand, the average number of iterations decreases dramatically after 5,000 episodes, reaching a value of five iterations per episode. This shows the capability of the proposed algorithm to determine an optimal solution in five trials.

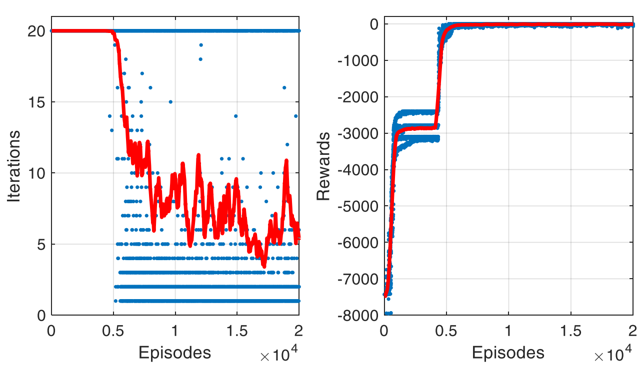

Fig. 3. Rewards and iterations per episode

To visualize the internal learning behavior the MADDPG, the actor and critic losses of all agents are plotted, as shown in Fig. 4. It is worth noting that training episodes differ from running episodes since training the model is executed every ten episodes, after having sufficient stored learned lessons in the memory buffer. As a result, the instant of sudden behavioral change of the agent losses in Fig. 4 at 8,000 training episodes does not match the rapid learning rate at 4,000 episodes in Fig. 3. The actor losses of A_1, A_4, and A_5 converge faster than A_2 and A_3. Also, A_3 takes more time to learn due to the unique location of DG_3 in the middle of distribution feeder where it has more possible island connections. For instance, DG_3 is responsible to supply S_5 only in case lines (29–30) and (6–26) fail; however, DG_3 and DG_4 will supply S_5 and S_6 if only line (6–26) fails. All agents converge to almost zero losses after 14,000 training episodes. On the other hand, the critic losses show much faster convergence rate. All agents provide pre-mature convergence after 5,000 training episodes. The sudden improved learning behavior of A_2 and A_3 at 9,000 training episode instant is reflected in the critic losses curve.

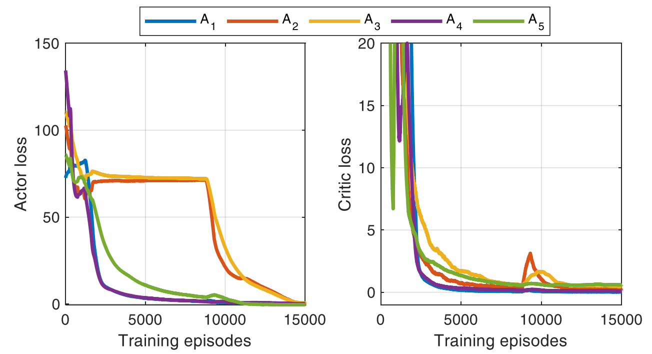

Fig. 4. Actor and critic losses for each agent

C. Testing and Validation

To validate the efficiency of the trained models, a total of 1,000 failure scenarios are tested. Algorithm 2 is used to test the proposed dispatch algorithm. For each episode, the model determines the required power supplied by each DG in the system. The trained model achieves 99.1% success rate of all the simulated cases. A successful decision is counted if the power supply mismatch of each DG does not exceed 15 kW. To validate the accuracy of the calculated DG power outputs for the successful cases, the average power mismatch is 8.5 kW. For non-successful cases, the average power mismatch is 22 kW. This implies that the non-successful cases have relatively close values to the predefined threshold. Further tuning of MADDPG hyper-parameters can results in enhanced accuracy.

The MADDPG is trained to determine the power supply of each DG to avoid load shedding. Table III provides the resulting outcome of the trained MADDPG model for ten failure scenarios. It is worth noting that the proposed algorithm computes the power supply based on the number of connected DGs in each microgrid. In other words, the required load demand of each microgrid is divided equally among the connected DGs within the same grid. In F_1, two microgrids are formed such that the first one includes S_5 and S_6 with total demand of 915 and the second one includes the rest of the feeder with total demand of 2805 kW. The total supplied power by DG_3 and DG_4 for the first microgrid is 922 and other DGs have total supply of 2784 kW. This shows the capability of the trained algorithm to provide relatively close values from the first trial. The same behavior is observed in F_2, F_3, F_4, and F_5. In case of two line failures, the distribution feeder is split into three smaller microgrids. Each set of DGs has a total power supply equal to the load demand of their corresponding microgrid. For instance, F_8 shows that DG_1 and DG_5 supply S_1, S_2, and S_4; DG_2 supplies S_3; and DG_3 and DG_4 supply S_5 and S_6; respectively. With three line failures, the operating conditions become more severe, and a higher power supply might be required from each independent DG. For example,

DG_1 and DG_3 supply higher power compared to DG_2 in F_{10} since they are connected to S_1, S_2, and S_5 forming the majority of the distribution feeder load. In general, the proposed algorithm shows the capability to dispatch DGs against severe situations under single and multiple line failures.

TABLE III
DG DISPATCH FOR SELECTED FAILURE SCENARIOS

	Index	Impacted lines	DG power (kW)				
			1	2	3	4	5
One	F_1	6-26	919	933	459	463	932
	F_2	2-19	832	358	831	848	840
	F_3	3-23	922	698	689	702	699
	F_4	29-30	764	774	771	625	772
	F_5	10-11	791	795	783	799	558
Two	F_6	2-19, 10-11	935	358	941	928	555
	F_7	3-23, 29-30	930	722	720	613	724
	F_8	2-19, 6-26	1218	364	460	453	1222
Three	F_9	3-23, 29-30, 10-11	931	809	807	615	554
	F_{10}	29-30, 2-19, 10-11	1092	358	1088	615	552

V. Conclusion

This paper has proposed a multi-agent reinforcement learning approach to enhance the operational resilience of islanded distribution systems after an extreme event. The proposed method computes the required power supply of DGs to maintain a minimal amount of load curtailments. A MADDPG framework was developed to dispatch DGs under multiple line failures. An MDP was formulated and used to train the MADDPG model such that the actor networks take an action upon which a reward value is computed and assigned to each agent at each iteration. In the training phase, diverse failure scenarios were simulated to create diverse stochastic conditions of the system under study. A modified IEEE 33-node distribution feeder was adopted to validate the effectiveness of the proposed algorithm. The results showed that the trained MADDPG model could provide proper decisions to maintain the reliable operation of an islanded power grid under additional multiple line failures. The trained model showed an accuracy exceeding 99%. The proposed method provides a corrective and restorative strategy to enhance the resilience of distribution systems leveraging the capabilities of reinforcement learning techniques. An extension of this work will include the integration of network reconfiguration strategies and the role of energy storage systems for further improvements.

Acknowledgement

This work was supported by the U.S. National Science Foundation (NSF) under Grant NSF 1847578.

References

[1] N. Bhusal, M. Abdelmalak, M. Kamruzzaman, and M. Benidris, "Power system resilience: Current practices, challenges, and future directions," *IEEE Access*, vol. 8, pp. 18 064–18 086, 2020.

[2] R. J. Campbell, "Weather-related power outages and electric system resiliency," Congressional Research Service, Tech. Rep., 2012.

[3] Billion-dollar weather and climate disasters. [Online]. Available: https://www.ncdc.noaa.gov/billions/events/US/2013-2021

[4] C. King, J. D. Rhodes, J. Zarnikau, N. Lin, E. Kutanoglu, B. Leibowicz, D. Niyogi, V. Rai, S. Santoso, D. Spence *et al.*, "The timeline and events of the february 2021 texas electric grid blackouts," *The University of Texas Energy Institute*, 2021.

[5] A. Khodaei, "Resiliency-oriented microgrid optimal scheduling," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1584–1591, 2014.

[6] A. Kavousi-Fard, M. Wang, and W. Su, "Stochastic resilient post-hurricane power system recovery based on mobile emergency resources and reconfigurable networked microgrids," *IEEE Access*, vol. 6, pp. 72 311–72 326, 2018.

[7] A. Gholami, T. Shekari, and S. Grijalva, "Proactive management of microgrids for resiliency enhancement: An adaptive robust approach," *IEEE Trans. on Sust. Energy*, vol. 10, no. 1, pp. 470–480, Jan 2019.

[8] J. Wu, X. Chen, S. Badakhshan, J. Zhang, and P. Wang, "Spectral graph clustering for intentional islanding operations in resilient hybrid energy systems," *arXiv preprint arXiv:2203.06579*, 2022.

[9] B. Chen, J. Wang, X. Lu, C. Chen, and S. Zhao, "Networked microgrids for grid resilience, robustness, and efficiency: A review," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 18–32, 2021.

[10] J. Liu, Y. Yu, and C. Qin, "Unified two-stage reconfiguration method for resilience enhancement of distribution systems," *IET Generation, Transmission & Distribution*, vol. 13, no. 9, pp. 1734–1745, 2019.

[11] S. Dharmasena, T. O. Olowu, and A. I. Sarwat, "Algorithmic formulation for network resilience enhancement by optimal der hosting and placement," *IEEE Access*, vol. 10, pp. 23 477–23 488, 2022.

[12] R. Rocchetta, "Enhancing the resilience of critical infrastructures: Statistical analysis of power grid spectral clustering and post-contingency vulnerability metrics," *Renewable and Sustainable Energy Reviews*, vol. 159, p. 112185, 2022.

[13] M. Abdelmalak and B.-I. Mohammed, "Proactive generation redispatch to enhance power system operation resilience during hurricanes," in *North American Power Symposium*, Tempe, AZ, USA, April 2021.

[14] M. Abdelmalak and M. Benidris, "Proactive generation redispatch to enhance power system resilience during hurricanes considering unavailability of renewable energy sources," *IEEE Transactions on Industry Applications*, vol. 58, no. 3, pp. 3044–3053, 2022.

[15] M. Yan, X. Ai, M. Shahidehpour, Z. Li, J. Wen, S. Bahramira, and A. Paaso, "Enhancing the transmission grid resilience in ice storms by optimal coordination of power system schedule with pre-positioning and routing of mobile dc de-icing devices," *IEEE Trans. Power Syst.*, vol. 34, no. 4, pp. 2663–2674, 2019.

[16] C. Wang, Y. Hou, F. Qiu, S. Lei, and K. Liu, "Resilience enhancement with sequentially proactive operation strategies," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2847–2857, 2017.

[17] J. Xie, I. Alvarez-Fernandez, and W. Sun, "A review of machine learning applications in power system resilience," in *2020 IEEE Power Energy Society General Meeting (PESGM)*, 2020, pp. 1–5.

[18] M. M. Hosseini and M. Parvania, "Artificial intelligence for resilience enhancement of power distribution systems," *The Electricity Journal*, vol. 34, no. 1, p. 106880, 2021.

[19] M. Kamruzzaman, J. Duan, D. Shi, and M. Benidris, "A deep reinforcement learning-based multi-agent framework to enhance power system resilience using shunt resources," *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 5525–5536, 2021.

[20] M. M. Hosseini and M. Parvania, "Resilient operation of distribution grids using deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 2100–2109, 2022.

[21] H. Nie, Y. Chen, Y. Xia, S. Huang, and B. Liu, "Optimizing the post-disaster control of islanded microgrid: A multi-agent deep reinforcement learning approach," *IEEE Access*, vol. 8, pp. 153 455–153 469, 2020.

[22] L. Tightiz and H. Yang, "Resilience microgrid as power system integrity protection scheme element with reinforcement learning based management," *IEEE Access*, vol. 9, pp. 83 963–83 975, 2021.

[23] Z.-c. Zhou, Z. Wu, and T. Jin, "Deep reinforcement learning framework for resilience enhancement of distribution systems under extreme weather events," *International Journal of Electrical Power & Energy Systems*, vol. 128, p. 106676, 2021.

[24] A. Zai and B. Brown, *Deep reinforcement learning in action*. Manning Publications, 2020.

[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.