Reinforcement Learning Enabled Microgrid Network Reconfiguration Under Disruptive Events

Jubeyer Rahman*, *Student Member*, *IEEE*, Roshni Anna Jacob*, *Student Member*, *IEEE*,
Steve Paul†, *Student Member*, *IEEE*, Souma Chowdhury†, *Member*, *IEEE*,
and Jie Zhang*, *Senior Member*, *IEEE*
*The University of Texas at Dallas, Richardson, TX 75080, USA
†University at Buffalo, Buffalo, NY 14260, USA
Email: jiezhang@utdallas.edu

***Abstract*—Network reconfiguration is an established technique for improving the performance of distribution network and microgrids. The ever-changing generation, load patterns, and increasing frequency of outages due to extreme events necessitate a more reliable, scalable, and faster reconfiguration algorithm to safeguard the grid assets and ensure smooth operations. A hybrid of value and policy based reinforcement learning (RL) algorithm is proposed to perform the network reconfiguration while considering other network constraints. The OpenAI Gym is used to build the RL-based model with an OpenDSS based environment (in the backend as a power-flow solving tool). A microgrid developed based on the IEEE 34-bus distribution test system is used in case studies with 9 switches (sectionalising and tie switches) under different loading and line contingency conditions. The proposed RL-based algorithm performs satisfactorily in terms of loss and load satisfaction compared with the baseline binary particle swarm optimization algorithm.**

***Index Terms*—Reinforcement learning, microgrid, reconfiguration, resilience, OpenDSS, OpenAI Gym.**

I. Introduction

The widespread deployment of microgrids in recent years has challenged the established premise of safe and economic operation of the power network. The dominating role of distributed generation (DG) along with the significant presence of smart appliances recurrently produce intermittent consumption and generation patterns. This leads to the frequent overloading of some phases of the network along with additional challenges like network congestion, voltage dip, lower reliability, higher power loss, etc. These phenomena often put the protection devices at risk and require more flexibility to be added to the grid [1].

Network reconfiguration is a widely adopted strategy to address these operational challenges in the grid and safeguard valuable grid assets. It is performed by modifying the existing network topology via establishing new power flow paths through changing the status of the sectionalising and tie switches. The switches in the network are mostly operated remotely and dealt with the help of a control scheme. Mathematically, it is a mixed-integer non-linear programming (MINLP) problem thus very challenging to solve. Numerous research efforts have been tried to address the problem by using conventional optimization techniques (e.g., metaheuristic and mathematical programming [2]–[7]). However, the adoption of the developed approaches in providing quick responses, particularly during post-fault load restoration, is contentious, since these methods are computationally slow in producing switching decisions. With the increasing number of controllable devices in the network, the number of decision variables also grows significantly and the obtained solution lacks the scalability and reliability.

Recently, reinforcement learning (RL) approaches have been adopted to address some of the seemingly intractable power system operational problems [8]. RL's high dimensional mapping capability has the potential to tackle the scalability issue in combinatorial optimization problems like network reconfiguration. Moreover, RL is a machine learning approach to learn optimal controllers from examples, which is more suitable for solving an optimization problem compared with other supervised and unsupervised machine learning approaches.

A model free Q-learning based reconfiguration strategy was developed as a part of a operation planning scheme in [9] with an objective to minimize the power dispatch loss. A hybrid multi-agent framework capitalizing Q-learning algorithm for transmission system restoration was proposed in [10]. The agents comprising generator agents, load agents, and switch agents were deployed at different nodes of the power grids, which could exchange information with each other without a centralized controller, thus avoiding a single point of failure. A high fidelity software tool to simulate the self-healing process through network reconfiguration of a distribution network was developed in [11], which used Q-learning to discover action policies to determine the switching actions to restore the lost loads. In [12], a modified Q-learning framework was proposed that offers a package of self-healing workflow including fault location, isolation, and service restoration. The action space consists of either the switch statuses or the percentage amount of shed load. A Q-learning based shipboard network reconfiguration scheme was proposed in [13], where switching operations resulting in a single loss of load were excluded from the action space. In [14], a dynamic distribution network reconfiguration problem was formulated as a Markov decision process. A Gaussian process was applied to learn the estimated values of total network loss and nodal voltage magnitudes along with their uncertainties. To avoid the sampling inefficiency due to the extrapolation error, a batch-constrained soft actor-critic model

978-1-6654-6591-5/22/$31.00

(that uses given historical operation data) was adopted in [15]. However, the aforementioned work did not use detailed network parameters, rather relied on simply the historical data to learn the control policy, which may lead to network constraints violations. In [16], perturbations were introduced to network weights in a modified deep Q-network to relieve from the tuning of exploration parameters thus reducing the training time. A graph-reinforcement learning model was proposed to perform sequential service restoration in [17], where each agent being fed by a concatenated output from preceding layers, encodes the observations by a Multi-Layer Perceptron (MLP) to produce the feature vector.

With a few exceptions, most of the work in the literature utilized value-based methods (e.g., Q-learning, deep Q-learning, etc.) that learn a value function, which maps each state action pair to a value but lack the ability to directly refine the agent's behavior. This often leads to poor performance of the algorithm. The main contribution of this paper is to utilize a hybrid of value and policy based RL method, called Proximal Policy Optimization (PPO), to solve the microgrid network reconfiguration problem for both normal and post-fault operations. Since PPO exploits the advantages of both the value- and policy-based approaches, it can measure the quality of taken action and control, or modify the agent's behavior and action to improve the performance.

The organization of the paper is given as follows. In section II, the microgrid network reconfiguration problem formulation is briefly discussed. Section III describes the details of the learning framework. The case study results of the proposed PPO-based network reconfiguration framework at varying operating conditions are present in Section IV. Finally, conclusions and future work are discussed in Section V.

II. Microgrid Network Reconfiguration Problem Formulation

The network reconfiguration has been formulated based on the AC load flow equations of a microgrid. It has taken into consideration the nodal real and reactive power balance constraints, bus voltage magnitude and angle limit constraints, branch flow constraints, etc. Although this work deals with the post-fault load restoration process via changing the switch statuses, the switching cost is not considered, thereby no constraint on switching count has been taken into account. The baseline binary particle swarm optimization (BPSO) algorithm explicitly utilizes all these constraints, whereas the RL-based algorithm implicitly embeds them in its structure. For brevity, the detailed formulation is excluded from this paper but can be found in [18].

The reconfiguration problem has been reformulated as a Markov Decision Process (MDP) to implement the proposed PPO algorithm. Let $P(s'|s_0, a)\ \forall s_0, s' \in S, \forall a \in A$ is the transition probability from state s_0 to s defined over the state space S at a given action a in the action space A. The action space formation is different in the two considered modes of operation. In the normal operating condition, the action space is reduced to contain only those switching configuration which are feasible. In the post-fault condition, all possible switching combinations are considered due to the prohibitively large feasible search space, considering the unpredictability in branch outages. Then the proposed MDP can be described as $M = (S, A, P(s'|s_0), R, \gamma)$, where R is the reward and γ is the discount factor. The state, action, transition probability, and reward formulations used in the MDP are described as follows.

*1) **State**:* The observation of the microgrid network constitutes the state of the agent, which includes different parameters, system variables, and network topology. So, the state of the network can be described for normal operation and post-fault operation as $S_{normal} = [P_g, Q_g, P_L, Q_L, V, \theta, I, \tau, P_{loss}]$ & $S_{post} = [P_g, Q_g, P_L, Q_L, V, \theta, I, \tau, P_{un}]$, respectively. Here, P_g and Q_g are the supplied real and reactive power, respectively; P_L and Q_L are the real and reactive load, respectively; V and θ represent node voltage magnitude and angle, respectively; line flow is denoted by I, and τ represents network topology. P_{loss} denotes the loss incurred for power distribution, which is included in the state vector while executing the normal operation. In the post-fault operation, P_{un} that represents the 'unserved power' to the loads is included in the state vector, since restoring power is prioritized during disruptions. Based on the given load parameters, the rest of the state vector elements are calculated by leveraging the OpenDSS power flow solver.

*2) **Action**:* At a certain step the agent takes an action a from the action space A. The action space has been designed depending on the mode of operation. During normal operating conditions, the action is chosen from a pool of feasible switching configurations. On the other hand, during post-fault conditions, all possible switching configurations are considered and a feasible search space is not predefined. There exists another major difference in the design of the action adopted during the two different conditions. In the normal operation, a configuration which is a vector of all switching status is picked up from the action space, $A_{normal} = [a_{comb_1}, a_{comb_2}, ..., a_{comb_n}]$, where $a_{comb_i} = [s_1, s_2, ..., s_N]$. Whereas for the post-fault operation, each switch status is determined by the policy network individually, thus the action vector is $\mathcal{A}_{post} = [s_1, s_2, ..., s_N]$ and the action space is composed of these action vectors, $A_{post} = [\mathcal{A}^1_{post}, \mathcal{A}^2_{post}, ..., \mathcal{A}^i_{post}, ...]$, where s_i is a binary variable representing the i_{th} switching status (1 for ON and 0 for OFF).

*3) **Transition Probability**:* The system moves from state s_0 to s' at a certain step according to the transition probability $P(s'|s_0, a)$ while action a is taken. In the normal operation, it is only the transition probability from one complete configuration to another one, and can be represented with a single probability term in Eq. 1, where f_θ represents the transition probability from one configuration to another.

$$P_{normal}(s'|s_0, a) = f_\theta \tag{1}$$

But for the post-fault operation, the probability is calculated from the cumulative probability of all the switches, which is the multiplication of the transition probability of all the

switches together since one switch's state is independent of others. The joint probability of a state transition can then be described in Eq. 2, where $f_{\theta(.)}$ represents the transition probability of individual switches from one state to another.

$$P_{post}(s'|s_0, a) = f_{\theta_1} \cdot f_{\theta_2} \cdot ... \cdot f_{\theta_N} \tag{2}$$

*4) **Reward**:* Given a specific operating condition, the reward function differs to suit the operational requirement. At the normal operating condition, the network loss reduction is given the utmost priority. So, the reward function aims to minimize the loss.

$$R_{norm} = -P_{loss} \tag{3}$$

At the post-fault operation, the priority is given to the maximum demand satisfaction of the consumers, so the reward function takes the form in Eq. 4.

$$\begin{aligned} R_{post} = -w_1 \cdot P_{un} - w_2 \cdot C_{V\,viol} - w_3 \cdot C_{I\,viol} \\ -w_4 \cdot (1 - F_{conv}) \end{aligned} \tag{4}$$

where w_1 ,w_2, w_3, and w_4 are the assigned weights (determined via empirical trials) to the reward components. Determining the values of these weights is critical, since it cannot be readily obtained that which of the reward terms represent the characteristics of what magnitude in the RL model. $C_{V\,viol}$ denotes the magnitude of voltage limit violation, which is determined by Eq. 5.

$$\begin{aligned} C_{V\,viol} = \sum_{i=1}^{\mathcal{N}} \sum_{k=1}^{\phi} \left[|v_{ik} - v_{max}| + |v_{min} - v_{ik}| \right], \\ \forall v_{ik} > v_{max} \vee v_{ik} < v_{min} \end{aligned} \tag{5}$$

where v_{ik} is the voltage magnitude at the k_{th} phase (of ϕ phases) of the i_{th} node (of $\mathcal{N}$ nodes), whereas v_{max} and v_{min} denote the maximum and minimum limit for the node voltages in the microgrid, respectively. Current violation is also penalized and has been integrated in the reward formulation. $C_{I\,viol}$ represents the current violation magnitude, which is determined similarly like the voltage violation magnitude from Eq. 6.

$$C_{I\,viol} = \sum_{j=1}^{\mathcal{L}} \left[|I_j| - I_{max_j} \right], \ \forall I_j > I_{max_j} \vee I_j < I_{min_j} \tag{6}$$

where I_j is the current magnitude at the j_{th} branch, whereas I_{max_j} and I_{min_j}, respectively, denote the maximum and minimum limit for the current in the j_{th} branch of the microgrid. Usually the upper limit and lower limit of the current are of same magnitude but with opposite signs so, the violation is calculated by taking the difference between the absolute current flow and the maximum limit, and each individual violation is summed over all the lines ($\mathcal{L}$) of the microgrid. F_{conv} is the convergence flag which indicates the convergence status of the case under consideration (1 for converged, and 0 for not converged), which has been incorporated in the reward function to encourage the convergence. To reduce the sparseness in the reward function values generated from each episode, all the terms in the reward function are expressed in per unit (p.u.). For normal operations, the actual values of network loss have been used, since it gives better learning performance.

III. The Learning Framework: Proximal Policy Optimization Algorithm

PPO belongs to the family of the policy gradient based RL algorithm, which offers the benefit of trust-region policy optimization (TRPO) [19] and performs the policy update through multiple epochs of gradient descent [20]. There are two available variants of PPO for policy update: PPO-penalty and PPO-clip. In this work, the PPO-clip version has been used, which updates the policy by maximizing a "surrogate" objective given in Eq. 7.

$$\phi_{k+1} = arg \max_{\phi} \mathop{\mathbb{E}}_{s,a \sim \pi_{\phi_k}} [L(s, a, \phi_k, \phi)] \tag{7}$$

where $L(.)$ denotes the loss function, ϕ is the vector of policy parameters, and π_{ϕ_k} is the stochastic policy.

$$\begin{aligned} L(s, a, \phi_k, \phi) = min(\frac{\pi_\phi}{\pi_{\phi_k}} A^{\pi_{\phi_k}}(s, a), \\ clip(\frac{\pi_\phi(a|s)}{\pi_{\phi_k}(a|s)}, 1 - \epsilon, 1 + \epsilon) A^{\pi_{\phi_k}}(s, a))) \end{aligned} \tag{8}$$

Here, ϵ is a hyperparameter which sets the upper limit on the policy update, and $A^{\pi_{\phi_k}}$ represents the advantage estimate. The policy network for the PPO considered here is based on a multi-layer perceptron (MLP) with a hyperbolic tangent as activation function. The input to the policy network is a vector that is formed by the concatenation of all the state variables, and the output of this network is a learned feature vector ($F_{feature}$) of size h (considered as 128 here). The input vector is of size 1,521 (computed by concatenating all the state variables). The log probability of the action space is computed using Eq. 9.

$$LogProb^i_{Action} = F_{feature}.W_{Action} \tag{9}$$

where W_{Action} is a learnable weight matrix of size $h \times N$. Since each switch can have just two states (ON or OFF), the probability distribution corresponding to the log probabilities computed by Eq. 9 can be considered as a Bernoulli distribution. This distribution is being used to sample the action for the Rollout operation in the RL training process. The learning model has been developed on python 3.9 using the Stable-Baselines3 library. The model parameters were determined through empirical processes but inspired from our previous work [18]. The model parameters for both operation schemes are summarized in Table I.

TABLE I
SETTINGS FOR MODEL TRAINING

DETAILS	VALUES
Algorithm	*PPO*
Total steps	80,000
Rollout buffer size	200
Batch size	100
Optimizer	*Adam*
Learning step size	10^{-5}
Entropy coefficient	0.1 $(normal)$ & 0.05 $(fault)$
Value function coefficient	0.5
Epochs	100

IV. RESULTS AND DISCUSSION

The proposed learning algorithm is tested on a microgrid developed based on the IEEE 34-bus test system. The original 34-bus network does not contain any switches, and modifications have been made in the test system to include nine switches, among which five are normally closed sectionalising switches and four are normally opened tie switches. A three phase distributed energy resource (DER) of 50 kW nominal capacity (a generator) has been placed at node 814 whose nominal operating voltage is 24.9 kV. The nominal real power load of this test system is 2.04 MW. The switch placement positions along with the DER location are illustrated in Fig. 1.

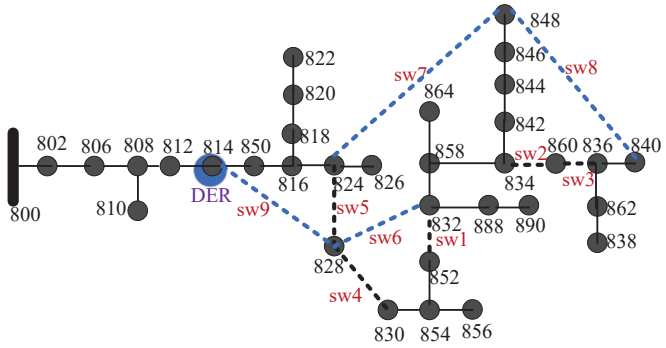

Fig. 1: Microgrid modified based on the IEEE 34-bus distribution test network at the normal condition. The dashed lines represent switches. The tie lines are marked in blue and sectionalising switches are marked in black.

A. Normal Operation

During the normal operating condition, only the loading condition of the test network is changed via a randomly generated load factor multiplier. During training, the microgrid load ranges from 0.5 to 2 times of the nominal rating. Prior to the training, all possible switching combinations are created and tested with the OpenDSS power flow solver. Switching combinations that result in infeasible solutions were excluded from the action space. This action space reduction is reasonable in the sense that the operator should at least have the knowledge of the switch positions. Instead of calculating the transition probability of individual switches, the RL algorithm calculates the transitional probability from one configuration to the other feasible configuration. Fig. 2 shows the training performance of the RL model during the normal operation, and the pattern is highly dependent on the hyperparameter selection. Table II compares the configuration performance between the proposed RL algorithm and the baseline BPSO algorithm. The switching statuses are represented in order by either 0 (switch open) or 1 (switch closed). It is observed that the proposed PPO-based RL performs relatively well with a little higher network losses, which is expected to decrease further after proper hyperparameter tuning. The proposed RL-based algorithm shows an acceptable performance while reducing the solution time significantly. It also accommodates and reflects the impact of different loading conditions by providing different switching combinations.

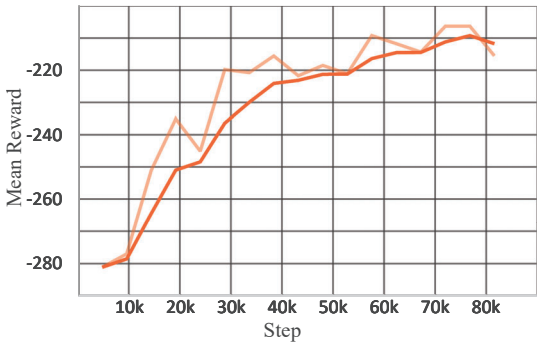

Fig. 2: Mean reward per episode during training in the normal operation (the brighter line represents the smoothed average value).

B. Post-fault Operation

To simulate the post-fault operation, an outage is created for an individual line at a time from a pool of candidate lines. The selection of the line for outage is random. The pool of candidate lines is created with an assumption that the remedial switching action can restore the shed load to some extent. The modified microgrid network tested for this post-fault reconfiguration study is illustrated in Fig. 3.

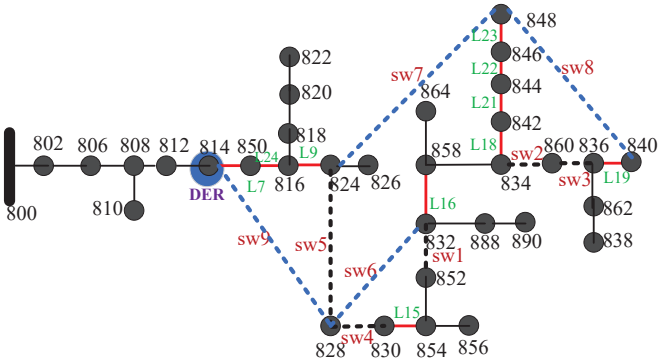

Fig. 3: Microgrid modified based on the IEEE 34-bus distribution test network for post-fault network reconfiguration. The outage candidate lines are marked in red.

In the post-fault condition, the action space was not reduced by removing infeasible switching configuration. As a result, the action space gets much bigger since both the loading and

TABLE II
NORMAL OPERATION: STATUSES OF SWITCHES IN THE MICROGRID NETWORK UNDER DIFFERENT LOAD CONDITIONS

Method	Load condition 1 (base load)		Load condition 2 (50% loading)		Load condition 3 (150% loading)		Mean Time (s)
	Switch status	Loss (kW)	Switch status	Loss (kW)	Switch status	Loss (kW)	
PPO-RL	[1, 1, 1, 0, 0 1, 0, 0, 1]	57.63	[1, 0, 1, 0, 1 1, 0, 1, 0]	22.38	[1, 1, 1, 0, 1 1, 0, 0, 0]	119.65	0.08
BPSO	[1, 1, 1, 0, 1, 1, 0, 0, 0]	56.66	[1, 0, 1, 0, 1, 1, 0, 1, 0]	22.38	[0, 1, 1, 1, 0, 1, 0, 0, 1]	110.11	245.398

line outages are randomly fed to the model. Unlike the normal operation scheme, the transition probability of individual switches is calculated and a joint probability for a configuration is approximated by the learning process. Fig. 4 illustrates the training performance of the learning algorithm, showing the tendency to reach to the steady state after a higher number of iterations due to the large action space. The learning process is enforced with the changing network configuration through the inclusion of network adjacency matrix into the state vector. It has been observed that the learning performance is largely dependent on the hyperparameter selection, especially on the learning rate and entropy coefficient. Table III shows the PPO-RL's reconfiguration results for different loading condition and line outages. Since there are 10 lines in the candidate pool and 3 different loading condition under consideration, there are 30 possible cases for this study. For brevity, only the results for all three loading conditions with a few line outages are presented here. It is observed that the 'unserved load' is slightly higher than the baseline BPSO approach but still within a reasonable limit in most of the cases, and the optimal configuration is obtained at a much faster speed compared with the baseline BPSO method, which is highly critical in post-fault operations.

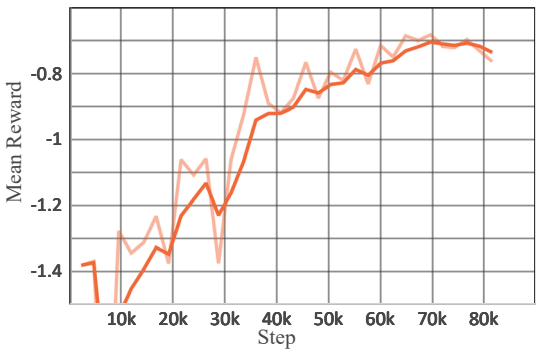

Fig. 4: Mean reward per episode during training in the post-fault operation (the brighter line represents the smoothed average value).

V. Conclusion

In this work, a reinforcement learning based algorithm was developed to address the inherent computational challenges of the network reconfiguration problem. The proposed proximal policy optimization is an advanced actor-critic type learning framework which leverages the benefit of both the value- and policy-based algorithms. The hybrid combination of these two provides advantages compared to the widely used Q-learning algorithm. Both the normal and post-fault operating conditions were taken into account to investigate the use of the proposed RL framework. Experimental results on a microgrid test system have shown the effectiveness of the PPO algorithm to produce reasonable solutions at a much faster speed. At the post-fault situation, the PPO-RL has been able to produce reasonable reconfiguration decisions with a reasonable loss in load in a much shorter time, showing the potential to be deployed as a decision assistance tool for the system operator during both normal and contingency operations.

Potential future work will refine the reward functions and fine-tune the hyperparameters to further improve the performance. The proposed algorithm will also be tested with larger networks to validate its fidelity for large-scale deployment.

Acknowledgment

This material is based upon work partially supported by the Engineering Research and Development Center – Construction Engineering Research Laboratory (ERDC-CERL) under Contract No. W9132T21C0007 and the Department of the Navy, Office of Naval Research under ONR award number N00014-21-1-2530. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of ERDC-CERL and the Office of Naval Research.

References

[1] P. Ramsami and R. T. A. King, "Dynamic distribution network reconfiguration for distributed generation integration: A systematic review," in *2021 IEEE 2nd China International Youth Conference on Electrical Engineering (CIYCEE)*. IEEE, 2021, pp. 1–8.

[2] S. Mishra, D. Das, and S. Paul, "A comprehensive review on power distribution network reconfiguration," *Energy Systems*, vol. 8, no. 2, pp. 227–284, 2017.

[3] Y. Ma, X. Tong, X. Zhou, and Z. Gao, "The review on distribution network reconfiguration," in *2017 29th Chinese Control And Decision Conference (CCDC)*. IEEE, 2017, pp. 2292–2297.

[4] O. Badran, S. Mekhilef, H. Mokhlis, and W. Dahalan, "Optimal reconfiguration of distribution system connected with distributed generations: A review of different methodologies," *Renewable and Sustainable Energy Reviews*, vol. 73, pp. 854–867, 2017.

[5] B. Sultana, M. Mustafa, U. Sultana, and A. R. Bhatti, "Review on reliability improvement and power loss reduction in distribution system via network reconfiguration," *Renewable and sustainable energy reviews*, vol. 66, pp. 297–310, 2016.

[6] R. Saxena, M. Jain, D. Sharma, and A. Mundra, "A review of load flow and network reconfiguration techniques with their enhancement for radial distribution network," in *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*. IEEE, 2016, pp. 569–574.

[7] M. J. Quintero Duran, J. E. Candelo Becerra, and V. Sousa Santos, "Recent trends of the most used metaheuristic techniques for distribution network reconfiguration," *Journal of Engineering Science and Technology Review*, vol. 10, pp. 159–173, 2017.

TABLE III
POST-FAULT OPERATION: STATUSES OF SWITCHES IN THE MICROGRID NETWORK UNDER DIFFERENT LOAD CONDITIONS & LINE OUTAGES

Line outage	**Method**	**Load condition 1 (base load)**		**Load condition 2 (50% loading)**		**Load condition 3 (150% loading)**		**Mean Time (s)**
		Switch status	Unserved load (kW)	**Switch status**	Unserved load (kW)	**Switch status**	Unserved load (kW)	
L15	PPO-RL	[1, 1, 0, 1, 1 1, 1, 1, 1]	1071.54	[1, 1, 1, 1, 1 0, 1, 1, 1]	477.68	[1, 1, 1, 1, 1 0, 1, 1, 1]	1718.86	0.06
	BPSO	[1, 1, 0, 1, 1 1, 1, 1, 0]	1053.23	[1, 1, 1, 1, 1 1, 1, 1, 0]	458.34	[1, 1, 0, 1, 1 1, 1, 1, 0]	1712.32	1624.22
L16	PPO-RL	[1, 1, 0, 1, 1 1, 1, 1, 1]	715.30	[1, 1, 1, 1, 1 0, 1, 0, 0]	261.89	[1, 1, 1, 1, 1 0, 1, 1, 1]	1202.50	0.07
	BPSO	[1, 1, 1, 1, 1 0, 1, 1, 0]	687.77	[1, 1, 1, 1, 1 0, 1, 1, 0]	261.85	[1, 1, 1, 1, 1 0, 1, 1, 0]	1183.86	1650.15
L18	PPO-RL	[1, 1, 1, 1, 1 0, 1, 1, 1]	381.13	[1, 1, 1, 1, 1 0, 1, 0, 1]	101.34	[1, 1, 1, 1, 1 1, 1, 1, 1]	770.46	0.08
	BPSO	[1, 1, 1, 1, 1 0, 0, 1, 0]	356.85	[1, 1, 1, 1, 1 0, 0, 1, 0]	70.01	[1, 1, 1, 1, 1 0, 1, 0, 0]	746.54	1785.36

[8] X. Chen, G. Qu, Y. Tang, S. Low, and N. Li, "Reinforcement learning for selective key applications in power systems: Recent advances and future challenges," *IEEE Transactions on Smart Grid*, 2022.

[9] J. G. Vlachogiannis and N. Hatziargyriou, "Reinforcement learning (rl) to optimal reconfiguration of radial distribution system (rds)," in *Hellenic Conference on Artificial Intelligence*. Springer, 2004, pp. 439–446.

[10] D. Ye, M. Zhang, and D. Sutanto, "A hybrid multiagent framework with q-learning for power grid systems restoration," *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 2434–2441, 2011.

[11] R. Ribeiro, F. Enembreck, D. M. Guisi, D. Casanova, M. Teixeira, F. A. de Souza, and A. P. Borges, "An advanced software tool to simulate service restoration problems: a case study on power distribution systems," *Procedia Computer Science*, vol. 108, pp. 675–684, 2017.

[12] L. R. Ferreira, A. R. Aoki, and G. Lambert-Torres, "A reinforcement learning approach to solve service restoration and load management simultaneously for distribution networks," *IEEE Access*, vol. 7, pp. 145 978–145 987, 2019.

[13] S. Das, S. Bose, S. Pal, N. N. Schulz, C. M. Scoglio, and B. Natarajan, "Dynamic reconfiguration of shipboard power systems using reinforcement learning," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 669–676, 2012.

[14] Y. Gao, J. Shi, W. Wang, and N. Yu, "Dynamic distribution network reconfiguration using reinforcement learning," in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2019, pp. 1–7.

[15] Y. Gao, W. Wang, J. Shi, and N. Yu, "Batch-constrained reinforcement learning for dynamic distribution network reconfiguration," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5357–5369, 2020.

[16] T. Zhao and J. Wang, "Learning sequential distribution system restoration via graph-reinforcement learning," *IEEE Transactions on Power Systems*, 2021.

[17] B. Wang, H. Zhu, H. Xu, Y. Bao, and H. Di, "Distribution network reconfiguration based on noisynet deep q-learning network," *IEEE Access*, vol. 9, pp. 90 358–90 365, 2021.

[18] R. A. Jacob, S. Paul, W. Li, S. Chowdhury, Y. R. Gel, and J. Zhang, "Reconfiguring unbalanced distribution networks using reinforcement learning over graphs," in *IEEE Texas Power and Energy Conference (TPEC)*. IEEE, 2022, pp. 1–6.

[19] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 1889–1897. [Online]. Available: https://proceedings.mlr.press/v37/schulman15.html

[20] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.