Electrical Power and Energy Systems 144 (2023) 108477

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

Post-storm repair crew dispatch for distribution grid restoration using stochastic Monte Carlo tree search and deep neural networks[☆]

Hang Shuai, Fangxing Li [*], Buxin She, Xiaofei Wang, Jin Zhao

Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, 37996, TN, USA

ARTICLE INFO

MSC:
1111

Keywords:
Deep neural network (DNN)
AlphaZero
Distribution grid restoration
Crew dispatch
Power system resilience

ABSTRACT

Natural disasters such as storms usually bring significant damages to distribution grids. This paper investigates the optimal routing of utility vehicles to restore outages in the distribution grid as fast as possible after a storm. First, the post-storm repair crew dispatch task with multiple utility vehicles is formulated as a sequential stochastic optimization problem. In the formulated optimization model, the belief state of the power grid is updated according to the phone calls from customers and the information collected by utility vehicles. Second, an AlphaZero based utility vehicle routing (AlphaZero-UVR) approach is developed to achieve the real-time dispatching of the repair crews. The proposed AlphaZero-UVR approach combines stochastic Monte-Carlo tree search (MCTS) with deep neural networks to give a lookahead search decisions, which can learn to navigate repair crews without human guidance. Simulation results show that the proposed approach can efficiently navigate crews to repair all outages.

1. Introduction

Climate change and global warming have caused an increase in both the frequency and intensity of extreme weather events, such as storms, floods, etc. As distribution grids expose to natural environment and are quite vulnerable, natural disasters can significantly damage power distribution systems, which will lead to different levels of power outages. Unfortunately, according to the study in [1,2], extreme weather events (storms, droughts, and floods, etc.) have become the leading reasons for the U.S. electrical grid power outages. More importantly, power outages caused by extreme weather events severely affected the normal operation of society and has brought huge economic losses [3]. For example, the power outage costs of the February 2021 Texas winter storm, which left millions of people without power, are estimated to be \$80 billion–\$130 billion. Therefore, related studies have attracted much attention in recent years [4,5]. For instance, researchers proposed advanced methods to enhance resilience from power system expansion planning [6], blackstart [7], and after-event recovery [8] perspectives.

To decrease the impact of extreme weather events on distribution grids, utilities have identified that measures should be taken to enhance the resilience [9]. Regarding power system resiliency enhancement, researchers have conducted plenty of works [10–13] recently and proposed many measures which can be divided into prior-to-events measures, during-events measures, and after-events measures, according to events unfolding process. Prior to events, damage estimation modeling [14] and outage preventive planning strategies [15–17] (such as crews allocation) can largely help system operators to decrease the affection of bad weather. During events, topology switching strategy, generation re-dispatching and load shedding strategies [18,19] are proposed to increase the resiliency of distribution systems. In addition, after extreme weather events passed, utility will assess the actual damages, then recover power supply through system restoration and load restoration strategies [20–24] by coordinating all the generation resources and repair crews [25].

In this paper, the after-storm restoration strategy for distribution systems is investigated. More specifically, this work focuses on the utility vehicle routing (UVR) problem with the goal of dispatching repair crews to fix all outages in the system as fast as possible. After power outages caused by storms reported, utilities need to estimate possible locations of damages, then repair crews will be scheduled to fix faulted devices. Considering repair crews usually driving utility vehicles to fix outages, so repair crew dispatch problem is largely a UVR problem.

Regarding post events UVR, there exists some researches. For instance, a constraint injection based optimization algorithm was proposed in [26] to solve the UVR problem, with the assumption that system operators can get the precise locations of all faults according to the engineer's operational knowledge. Ref. [27] proposed a two-stage outage management method for distribution system repair and

[☆] Note: A copy of this manuscript has been posted to arXiv.
[*] Corresponding author.
E-mail address: fli6@utk.edu (F. Li).

https://doi.org/10.1016/j.ijepes.2022.108477
Received 5 February 2022; Received in revised form 2 July 2022; Accepted 10 July 2022
Available online 9 August 2022

Nomenclature

Sets

ℓ^e	Set of power line fault combinations on circuit e
Ξ	Circuits set
B	Node set of distribution grid
E	Node set of road network
K^s	Set of lines (if faulted) that will cause the power supply failure of segment s
S^e	Set of segments on circuit e

Variables

θ	Weights of neural network
A	Trajectory of the utility vehicle
H, G	Binary indicator of received phone calls and location of the vehicle
L_t^e	Possible realizations of faulted power lines
n_k^e	Number of customers connected to node k on circuit e
P, P^{post}	Prior and post probability of power line fault
Q	Mean action-value
r, a	Reward and decision (action) variable
S	State variable
$T_t^{travel}, T_t^{repair}$	Travel and repair time, respectively
W	Total action-value
z, v	Target value and computed value by neural network

Superscripts and subscripts

b, i	Index of node and power line, respectively
e	Circuit
l, t	Hypothetical time-step and time index, respectively
s	Index of segment (power lines that trigger the same protective device)
z	Index of vehicle

restoration problem (DSRRP), and the repair tasks are clustered to crews according to the damage location and damaged components information in the first stage. In [28], a multiperiod distribution system restoration model, which is in response to multiple outages caused by natural disasters, was proposed. A resilient after disaster recovery scheme was proposed in [29] to co-optimize distribution system restoration with the dispatch of repair crews and mobile power sources. Similar with [26], the works in [27–29] assume that system operators can obtain the precise locations of all faults in the grid. However, according to [30,31], this assumption is sometimes too idealistic.

In general, there are two types of methods to obtain the damage location information of distribution grids, one is to send out crews for damage assessment, and the other is the fault location technique based method. In practice, utilities will send out crews for damage assessment after storms. Once damage assessment is complete, line crews begin making repairs [32]. However, a comprehensive assessment of all damages could take time as damage to the distribution grid can be widespread after a storm. For instance, depending on the severity of the storm, Duke Energy's assessment process can take up to 24 h after the weather passes [32]. Thus, it will take much time before utilities can make an optimal crew dispatch scheduling to repair all damages. To improve restoration efficiency of distribution grids after a storm, this work attempts to propose a learning-based crew dispatch framework that combines damage assessment with optimal repair crew dispatch. In the proposed restoration strategy, fault location technique will be used.

Researchers have developed a variety of fault location methods by using different input data (e.g., non-electrical data, electrical data, network data, and measurements), but accurate detection of faults and locations in distribution systems is still a tough task (see [33–36]). For instance, traditional distribution grids identify power outages through trouble calls from customers due to the weak situation awareness ability of utilities [30]. Thus, some utilities may not aware of outages and possible locations of faults until receiving the outage-reporting calls from customers [30,31,35]. With the development of modern distribution networks, advanced metering infrastructure (AMI) enables the utilities to remotely read consumer consumption records, and provides a new source of information for outage location. However, similar to the trouble call-based methods mentioned above, AMI-based outage location method can only estimate the most likely area or location of faults [36]. The benefit of the AMI-based approaches is that the operators do not need to wait for a sufficient number of customer calls to locate the outage areas. In this way, developing a post storm restoration strategy which does not rely on the precise location information of faults is meaningful for utilities.

Different from the above research works, this paper focuses on the post-storm UVR problem without knowing the precise locations of damages. Considering the precise damage locations of distribution systems after a storm are usually unknown before damage assessment, Ref. [37] proposed an information collecting vehicle routing model, which uses trouble calls from customers and fault information dynamically collected by crews on a vehicle to create beliefs about outages. More importantly, in Ref. [37], the UVR problem was formulated as a sequential stochastic optimization model and a Monte Carlo tree search (MCTS) based vehicle navigation strategy was innovatively proposed. The work of [37] paves the way for the application of MCTS to distribution grid restoration. Based on the work of [37], the authors of this paper proposed an open loop upper confidence bound for trees (OLUCT) algorithm based utility vehicle routing strategy in [38]. However, the traditional tree search methods adopted in [37,38] need a huge number of iterations, which is relatively time consuming, to find the most efficient path. Besides, Ref. [37,38] simplified the post-storm UVR problem by assuming that there is only one utility vehicle in the system. Thus, the discussion in [37,38] are all focused on single UVR problem. Nevertheless, utilities usually have multiple repair vehicles which are standby after a storm. It is more realistic to develop a post-storm repair crew dispatch model with multiple repair vehicles and design a multi-vehicle routing algorithm. Solving the multi-vehicle routing problem without knowing the precise location of faults is a very challenging task. The advanced learning-based technique will be utilized to solve the optimization problem in this work.

Recently, with the development of deep reinforcement learning (DRL), plenty of machine learning algorithms have been proposed and obtained superhuman performance in a variety of sequential decision problems [39,40], including solving problems in power industry [41]. For instance, AlphaGo [39] and AlphaZero [40] algorithms achieved superhuman performance in board games including the game of Go. Specially, AlphaZero convincingly defeated world-champion players without human guidance and domain knowledge beyond game rules, by tabula rasa reinforcement learning from data generated using self-play mechanism. The core of AlphaZero algorithm [40] is the combination of MCTS with deep neural network (DNN) which consists of a policy network and a value network. Once well-trained off-line, the policy network and value network can effectively guide the tree search process to make optimal actions. The authors of this paper have discussed several potential AlphaGo-like application scenarios in power systems in [42], and also investigated the application of MuZero [43] in the microgrid optimal scheduling problem in [44]. The research works in [42,44] indicate that the AlphaGo/AlphaZero/MuZero based algorithms are promising for solving challenging problems in power systems. However, the AlphaGo/AlphaZero algorithm was originally designed to play game of Go. It is worth noting that there are many differences between the distribution grid restoration problem and playing Go. For example, the state transition function of the board game is deterministic. But the transition function of the UVR problem is stochastic as the partially observation characteristic of the post-storm distribution system.

To this end, this paper focuses on post-storm repair crew dispatch problem with multiple utility vehicles and investigates the application of AlphaZero in post-storm UVR to restore distribution grids as fast as possible. With the advantages of the AlphaZero approach, the results of this work demonstrated that the proposed navigation strategy can make real-time crew dispatch decisions according to the current system state, which is critical for this specific application problem. The main contributions of this work are summarized as follows:

(1) A post-storm repair crew dispatch model with multiple utility vehicles is formulated.
(2) An AlphaZero [40] based post-storm utility vehicle routing (AlphaZero-UVR) algorithm is proposed to navigate multiple utility vehicles to restore the distribution grid as fast as possible.
(3) AlphaZero algorithm has been applied to playing board games which are with deterministic state transition functions. However, the state transition of the multiple time-step optimization problem in this work is stochastic as the actual damages of the unvisited power lines are unknown. To apply AlphaZero to this problem, the authors modified the original AlphaZero algorithm by combining stochastic MCTS method with DNN.
(4) The simulation results on a 8-node test system and a modified IEEE 123-node distribution system demonstrate the effectiveness of the proposed crew dispatch strategy.

This paper is organized as follows. The stochastic UVR problem is formulated as a Markov decision process (MDP) problem in Section 2. Section 3 presents the developed AlphaZero based UVR algorithm. The simulation results on two distribution systems are given in Section 4. Section 5 concludes the paper.

2. Post-storm repair crew dispatch problem

The repair crew dispatch problem is modeled as the UVR problem. In this section, the distribution network fault location method used in this work is presented firstly. Then, the trouble call-based UVR problem is introduced. Finally, the post-storm UVR problem with multiple utility vehicles is formulated as an MDP problem. Note that, in [38,45], a trouble call-based UVR model with single utility vehicle was formulated. In this work, the UVR model is extended for multiple utility vehicles, and a new AlphaZero based optimization method is designed to solve the problem in this paper.

2.1. Post-storm fault location method

The distribution system includes substations, overhead power lines, poles, transformers, protective devices, and customers, as shown in Fig. 1. The electricity is distributed through power lines and delivered to customers. Customers are connected to transformers which are fixed on the poles. Besides, to isolate faults, a number of protective devices, such as protective relays, disconnect switches, etc., are also installed on the poles. Once a protective device is triggered, all the downstream customers will suffer from power failure. For instance, when the protective device on pole 2 in Fig. 1 is triggered, then, the downstream customers connected to node 4 and 5 will lost power supply.

The input data used by fault location algorithms can be classified into four groups [36], namely non-electrical data (e.g., customer calls, weather data), electrical data (e.g., SCADA data, smart meter data), network data (e.g., line, network topology), and measurements (e.g., substation voltage and current). Based on the required inputs, the outage location methods can be classified to trouble call-based methods, historical data-based methods, fault indicators based algorithms, AMI-based methods, and algorithms using a combination of different sources of data. In this work, the trouble call-based method is adopted to estimate the possible locations of faults after storms. So, the utility identifies the damages in the system according to the phone-call reports received by electric utility center (EUC), weather data, and other network data.

After a storm passed, each equipment in the system has a fault possibility. This possibility is determined by many factors, such as the trajectory, strength, and duration time of a storm weather, and the anti-storm capacity of a device, etc. So, after storms passed, EUC will evaluate the damage probabilities of all the devices, and dispatch crews to repair all outages. To evaluate the damage status, the information that the EUC can take use of including the trouble calls from customers, the actual damage status information collected by the crews on their passed paths, and the prior fault probabilities of devices which can be evaluated according to the storm forecast information and the operation experiences from the system operators.

Based on Refs. [37,38,45], the posterior probability of power line i on circuit e being in fault at time t given the phone calls H_t and the trajectory of the vehicles A_{t-1}, can be calculated using Bayes' theorem as follows:

$$p(L^e_{t,i} = 1|H_t, A_{t-1}) = \frac{\Sigma_{L^e_t \in \{\ell^e\}_{L^e_{t,i}=1}} p(H_t|L^e_t)p(L^e_t|A_{t-1}, O_{t-1})}{\Sigma_{L^e_t \in \{\ell^e\}} p(H_t|L^e_t)p(L^e_t|A_{t-1}, O_{t-1})} \tag{1}$$

where A_t represents the vehicle's trajectory up to time t. In other words, A_t consists of all the past routing decisions. For instance, when the vehicle located at node 2 of the distribution grid in Fig. 1, one possible trajectory is $[0 \longrightarrow 1 \longrightarrow 6 \longrightarrow 2]$. $H_t = \{H^e_{t,b} : b \in B\}$ is a possible realization of the received trouble calls. $H^e_{t,b}$ represents whether the EUC received reporting calls from node b on circuit e by time t. For a circuit e, the possible realizations of the faulted power lines is represented by the vector L^e_t, and the ith element of the vector is $L^e_{t,i}$. When power line i is faulted at time t, $L^e_{t,i}$ equals to 1. So, $\{\ell^e\}_{L^e_{t,i}=1}$ is a subset of vectors of L^e where power line i is faulted. Note that line faults may be caused by damage to the line itself or to other equipment (such as transformers) connected to the line. The likelihood $p(L^e_t|A_{t-1}, O_{t-1})$ is the prior fault probability of power lines. O_t is a vector which contains the information collected (observed) by the crews by time-step t. For example, if the crew visited a power line i at time-step t, the collected outage information of the power line i will be added to O_t. The prior fault probability can be obtained according to the operational experience and the storm information. Besides, prior fault probability keeps being updated in the following time periods using the information collected by the vehicles. $p(H_t|L^e_t)$ is the likelihood of the calls given the power line faults on circuit e, which can be calculated by Eq. (2).

$$p(H_t|L^e_t) = \begin{cases} \prod\limits_{i \in \Psi_{J(L^e_t)}} 1 - (1 - \rho_i)^{n_i}, \, if \;\; L^e_t \in Z(H_t) \\ 0, otherwise \end{cases} \tag{2}$$

where $Z(H_t)$ represents all the combinations of power lines that will cause all customers in H_t to call if faulted. $\Psi_J(L^e_t)$ denotes the set of affected nodes (lose power) if the protective devices of the faulted power lines L^e_t are triggered. ρ_i is the customer calling probability when they suffer from power outage. n_i is the total number of customers connected to node i.

As indicated by the research in [45], the computational complexity of the fault probability model (1) is mainly affected by the number of power lines in a power grid. Fortunately, there are several approaches that can largely decrease the computational complexity such as power lines aggregation and Monte Carlo simulation. Readers are referred to Ref. [45] for more details.

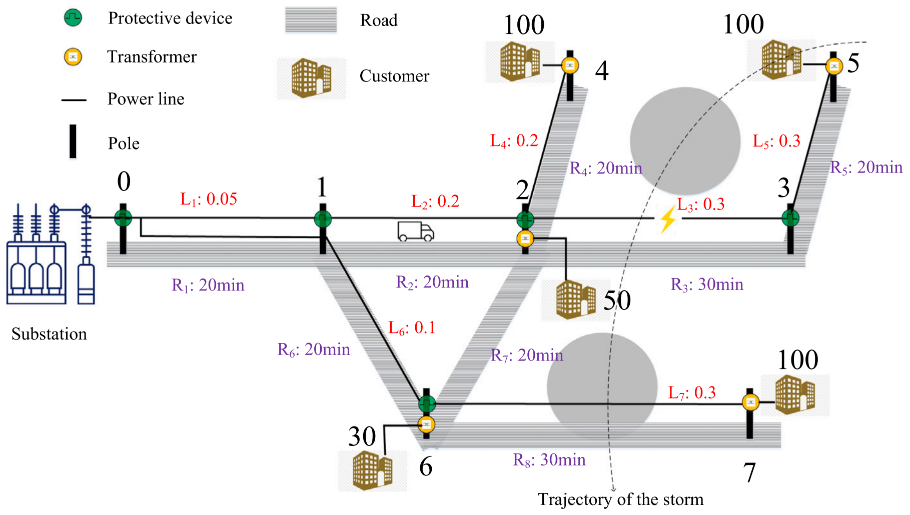

Fig. 1. Utility vehicle routing in a distribution system [38].

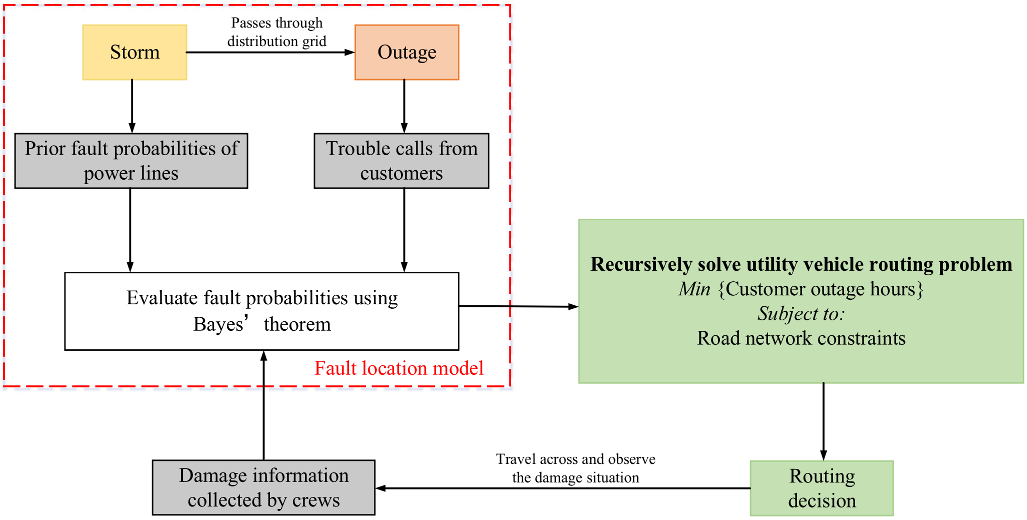
Storm
Passes through distribution grid
Outage
Prior fault probabilities of power lines
Trouble calls from customers
Evaluate fault probabilities using Bayes' theorem
Fault location model
Recursively solve utility vehicle routing problem
Min {Customer outage hours}
Subject to:
Road network constraints
Routing decision
Travel across and observe the damage situation
Damage information collected by crews

Fig. 2. Illustration of the post-storm UVR problem.

2.2. Post-storm UVR problem

After evaluated the fault probabilities of all devices according to the reporting calls from customers and the latest damage information collected by utility vehicles, EUC needs to make real-time routing decisions recursively. The real-time decision is obtained by solving the UVR problem that minimizes the outage-hours of the system. In addition, the optimization problem must subjects to road network constraints. The road network constraints mainly determine the feasible action space of the vehicle. For instance, if the vehicle located in node 2 of the system shown in Fig. 1, the feasible destination nodes of the vehicle at the next time-step will be 1, 3, 4, and 6. The proposed UVR scheme can be summarized as Fig. 2. Although the UVR optimization scheme shown in Fig. 2 adopts the trouble call-based outage location method, other outage/fault location methods (e.g., AMI based methods) can be easily applied in the architecture, as the UVR problem only needs to know the fault probabilities of devices provided by the fault location model. The power flow constraints are not included in the UVR optimization problem, as the damage status of a power line that unvisited by the vehicles is unknown.

2.3. Optimization model of the post-storm UVR problem

After the storm passed, EUC will dispatch multiple utility vehicles (crews) travel across the system to fix all possible damages, with the objective of scheduling the crews to restore the grid as quickly as possible. Based on the work of [37,38], the sequential stochastic optimization problem is formulated as an MDP. However, different from Refs. [37,38] which focus on single vehicle routing, this work investigates the multiple utility vehicles routing problem.

The designed post-storm multi-vehicle scheduling architecture is shown in Fig. 3. The distribution grid is divided into Z zones, and each repair vehicle travels across a specific zone to fix all the damages found in that area. When the vehicle travels from the starting node to the end node of a power line, it will check the damage status of the devices (such as the passed lines, transformers), and the crews on the vehicle will report the observation to the EUC. Then, the EUC will update the fault probabilities of all power lines in the whole distribution system using Eq. (1). Once a vehicle reached the end node of a line and did not find any fault or it found faults and repaired the damages, the vehicle will send a scheduling request to the EUC in order to get the optimal action of the next time-step. After the EUC received the request, it will calculate the optimal travel action of this vehicle according to the updated fault probabilities of all power lines and the feasible action

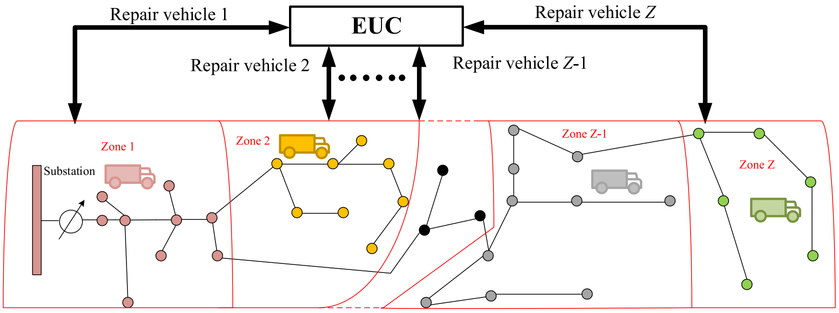

Fig. 3. Illustration of the multi-vehicle routing in a distribution system. The red line represents boundaries of each zone.

space of the vehicle. The feasible action space is related to the current position of the vehicle and the topology of the road system. In this work, the EUC will use the well-trained AlphaZero agent to get the optimal action of the vehicle. After received the action signal from the EUC, the vehicle will go to the destination node. The above process is repeated until all the possible damages have been repaired. If the EUC received scheduling requests from multiple vehicles at the same time, each time the EUC will select one repair vehicle from the request queue according to the priority to give scheduling decision until the queue becomes empty. Although the unselected vehicle should wait for EUC to dispatch all previous vehicles, this will not affect the optimality of the dispatch of the vehicles as the time consumption of the decision-making process is negligible.

In the following section, the basic elements of the MDP model will be defined.

2.3.1. State variable

$$S_t = \left\{ G_t, P_t, H_t \right\} = \left\{ G_t, P_t^{post} \right\} \tag{3}$$

The state variables consist of three parts, namely, the physical state G_t, the belief state P_t, and the informational states H_t. G_t is location of the vehicle under scheduling at current time-step, and P_t is the prior fault probabilities of power lines. The elements of the vector P_t are $p(L^e_{t,i} = 1|A_{t-1})$. Using P_t and H_t, the posterior fault probability P_t^{post} can be obtained according to Eq. (1).

2.3.2. Decision variable

$$a^z_t = (a^z_{t,ij})_{i,j\in E} \tag{4}$$

where a^z_t is the routing decision of the zth vehicle at time-step t. If the zth vehicle goes from node i to j at time-step t, then $a^z_{t,ij} = 1$.

2.3.3. Transition function

After taking action given in Eq. (4), the physical location of the zth vehicle, fault probability of the line visited by the vehicle, and informational states H_t will be transited to a new state, as shown in the following equations:

$$G_{t+1} = j,\ if\ a^z_{t,ij} = 1\ and\ I^z_{t+1} = 1 \tag{5}$$

$$p(L^e_{t+1,j}|\sum_i \sum_v a^v_{t,ij} = 1) = 0 \tag{6}$$

$$\begin{cases} H_{t+1,i} = H_{t,i}, if\ \hat{H}_{t+1,i} = 0 \\ H_{t+1,i} = 1, if\ \hat{H}_{t+1,i} = 1 \end{cases} \tag{7}$$

where $I^z_{t+1} = 1$ represents the zth vehicle sends the scheduling request at time-step $t+1$. After taking action $a^z_{t,ij}$, the zth vehicle will locate at node j at time-step $t+1$, as shown in Eq. (5). In this work, once the faulted power lines were fixed by crews, the fixed lines will not be in fault again in the following times, as shown in Eq. (6). It is worth noting that if the fault probability of the jth power line is updated, the fault probabilities of other power lines also need to be updated using Eq. (1). $\hat{H}_{t+1,i}$ is the indicator of newly arrived trouble calls from node i at time-step $t+1$, and '1' represents received new calls. In Eq. (7), H_{t+1} is a binary vector variable and $H_{t+1,i}$ represents whether the EUC received reporting calls from node i by time $t+1$. From Eq. (7), if the EUC received newly arrived reporting calls from node i, the ith element of H_{t+1} will be '1'. Otherwise, the value of the ith element will be unchanged.

2.3.4. Objective function

The objective function of the UVR problem is given by:

$$\begin{aligned} F &= \min_{\pi} \mathrm{E}\left\{ \sum_{t=0}^{T} C_t(S_t, A^\pi(S_t))|S_0 \right\} \\ &= \max_{\pi} \mathrm{E}\left\{ \sum_{t=0}^{T} r_t(S_t, A^\pi(S_t))|S_0 \right\} \end{aligned} \tag{8}$$

The optimization objective is to minimize the cumulative custom outage hours. π represents the policy adopted in the optimization. $C_t(S_t, a_t)$ is the custom outage hour when the system at state S_t and takes decision a_t. $r_t(S_t, a_t)$ is the reward function and $r_t(S_t, a_t) = -C_t(S_t, a_t)$. In Eq. (8) and the following equations, a^z_t is abbreviated as a_t.

According to Bellman's optimality, the optimal policy can be solved by:

$$A^*_t(S_t) = \arg\max_{a_t\in\chi_t(S_t)}\left(r_t(S_t, a_t) + \max_{\pi} \mathrm{E}\left\{ \sum_{\tau=t+1}^{T} r_\tau(S_\tau, A^\pi_\tau(S_\tau))|S_0 \right\}\right) \tag{9}$$

Note that $r_t(S_t, a_t)$ cannot be exactly calculated during the repairing process even though S_t and a_t are known. Because the number of restored customers after the utility vehicle visited a power line depends on upstream and downstream outages of the system. Unfortunately, these outages are uncertain. However, the expected value of the reward can be evaluated as follows:

$$\begin{cases} r_t(S_t, a_t) = -\left(\sum_{e\in\Xi} \sum_{s\in S^e} \left(1 - \Pi_{k\in K^s} p(L^e_{t,k} = 0)\right) \sum_{i\in s} n^e_i \right) \cdot T_t \\ T_t = T^{travel}_t + T^{repair}_t \end{cases} \tag{10}$$

where K^s is the set of lines (if faulted) that will cause the power supply failure of segment s. Ξ is the set of circuits in the system. S^e is the set of segments on circuit e. n^e_i is the number of customers connected to node i on circuit e. $\sum_{i\in s} n^e_i$ is the number of customers across segment s. T_t is the time needed to go from the current node to the destination node, which includes the travel time and the repair time. The travel time is determined by the length of the road and the driving speed of the vehicle. The repair time is related to the fault locations. Using Eq. (10), the expectation of unserved customers at each time step can be evaluated.

From the above equations, it can be found that the UVR problem is formulated as a sequential stochastic optimization problem. Solving the

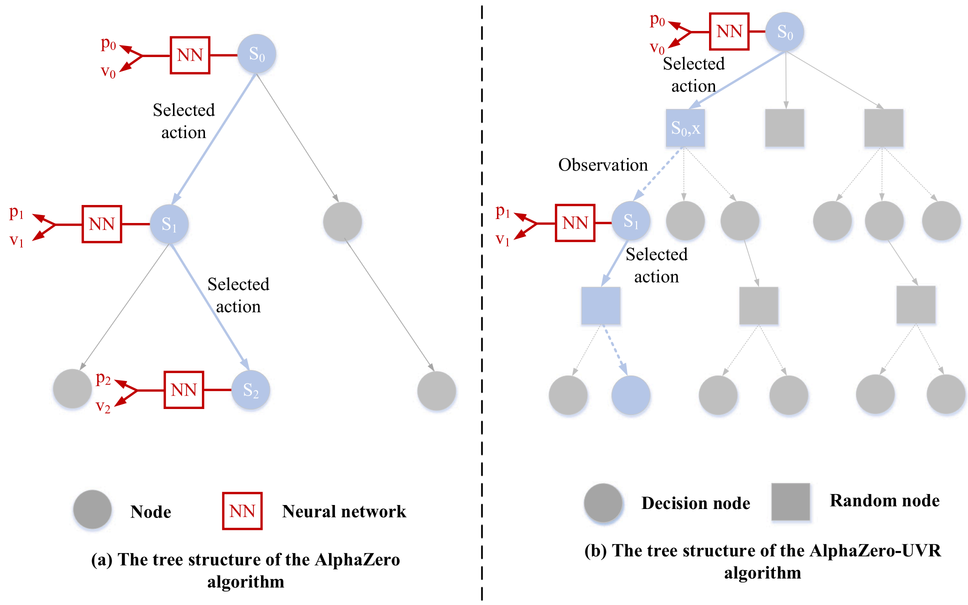

Fig. 4. Tree structure of the original AlphaZero algorithm [40] and the developed AlphaZero-UVR algorithm. The neural network takes system state as input and outputs action probabilities **p** with components $p_a = Pr(a|S)$ for each action a, and a scalar value v which represents the estimation of the expected outcome from the current state.

above problem faces several challenges. First, how many customers are restored after a line fixed by the crews is uncertain since the incomplete information of other faults. Second, the changing belief states (keep updating during the repairing process) of the grid further increases the difficulty of the problem solving. Traditional mathematical optimization approaches (i.e., integer programming) are difficult to solve this optimization problem. Ref. [37] proposed a MCTS based single utility vehicle routing approach which is a machine learning based optimization algorithm. Nevertheless, the performance of the MCTS based approach still has much room for improvement. Reinforcement learning (RL) methods have been widely adopted to solve MDP problems and plenty of research works have demonstrated that DRL methods can effectively solve many challenging tasks, like game of Go [40]. So, this work investigates using the state-of-the-art DRL technique developed by [40], to solve the multi-vehicle routing problem in this work.

3. AlphaZero based post-storm utility vehicle routing algorithm

In this section, an AlphaZero [40] based post-storm UVR algorithm is proposed to guide the crews to repair grid outages as fast as possible. The challenges of directly using AlphaZero algorithm to solve the UVR problem are presented. In this way, a modified AlphaZero architecture is developed for the problem. Then the AlphaZero based utility vehicle routing strategy is designed.

3.1. AlphaZero algorithm

AlphaZero [40] is a model-based deep reinforcement learning algorithm developed by DeepMind in 2018 to play the games of chess, shogi, and Go, which has achieved superhuman performance. Similar with AlphaGo, the key issue of AlphaZero is to integrate deep learning technique into Monte-Carlo tree search (MCTS). The advantage of AlphaZero is that it does not need human guidance and domain knowledge of the game except the game rules, and learns to play the game entirely by self-play. Besides, a single DNN architecture that contains policy outputs and value output is proposed in AlphaZero.

The general principle of AlphaZero is that it plays against itself to generate training data, with each side of the game choosing actions by MCTS strategy, and the generated self-play game data are sampled to continually *train the deep neural network*. Then, using the latest DNN, the new game data are generated by *self-play*. These procedures repeat until the algorithm converged. The *self-play* and *neural network training* are conducted in parallel, each improving the other. During the self-play procedure, MCTS is used to get the optimal action of each time-step, while the MCTS uses the neural network to guide its simulation as shown in Fig. 4(a). In the figure, every node is associated with a system state S, and the edge (S,a) of the search tree contains the prior probability of selecting the edge $P(S,a)$, the visit count $N(S,a)$, the total action-value $W(S,a)$, and the mean action-value $Q(S,a)$. The search tree shown in the figure is constructed by conducting a predefined number of simulations, and each simulation consists of the *selection*, *expansion*, and *backpropagation* procedures. Details about the MCTS simulation are elaborated in the following section.

3.2. AlphaZero based post-storm utility vehicle routing approach

Plenty of machine learning algorithms need massive training data. However, extreme weather events are usually low probability, which means EUC only has limited event data and the corresponding action (crew dispatch) data. Moreover, it is not sure whether the historical action data is optimal. So, to solve the post-storm UVR problem by using machine learning approaches, one challenge is that researchers do not have enough (labeled) data to train the designed algorithm. Luckily, the self-play mechanism proposed in AlphaZero provides a good solution to this problem. However, playing game of Go is very different from the UVR problem in this work. Actually, to apply AlphaZero to the UVR problem, it faces challenges brought by the differences between board games and the problem in this work. More specifically, the original AlphaZero algorithm is designed for the two-player games, while the UVR problem can only be viewed as a single-player game. Fortunately, this difference will not hinder the application of the algorithm in the UVR problem and it even simplifies the application of the algorithm as the agent does not need to change the players during the training and online application processes.

Besides, the original AlphaZero algorithm is used to solve the MDP problems with a deterministic transition function. For instance, in the game of Go, the next board state is deterministic after taking a specific action. However, the transition function of the UVR problem in this work is stochastic. The uncertainty comes from the partially observation characteristic of the post-storm distribution system. For

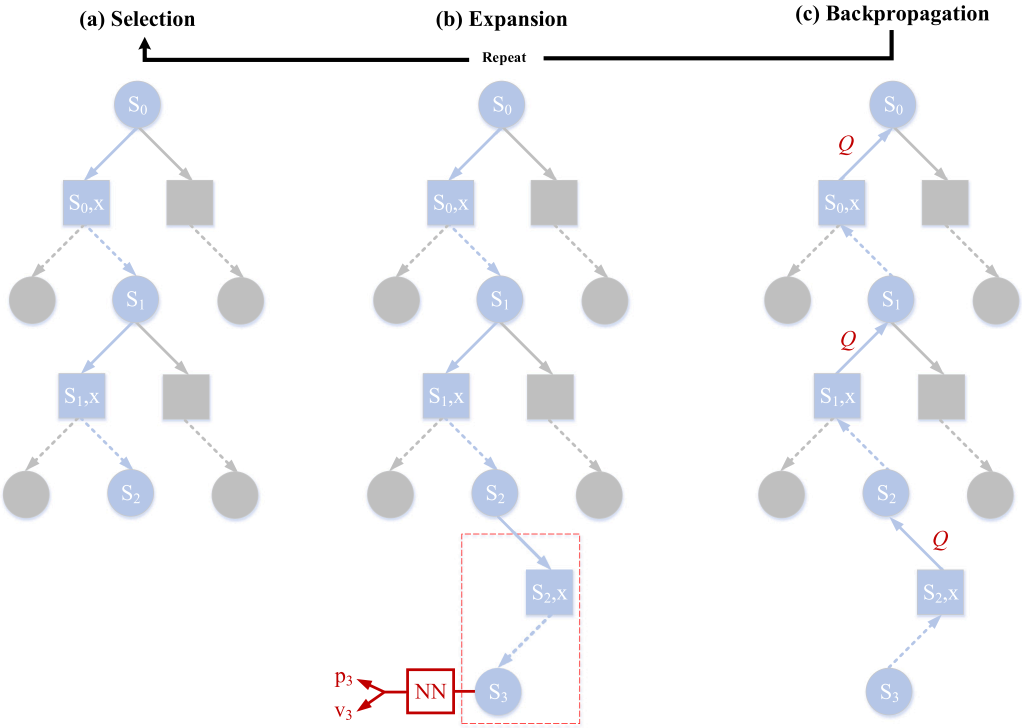

Fig. 5. MCTS in AlphaZero-UVR.

example, at time t, it is assumed that the vehicle is located in node 1 of Fig. 1, and makes the decision (action) to head to node 2, which means it will pass through line L_2. After the vehicle passed the power line, the actual damage status of L_2 will be observed (or detected), either damaged or not. From Eq. (1), different observed damage status will lead to different evaluation value of the other lines' fault probabilities. So, even if the vehicle takes the same navigation decision, it has the probability of transferring to different states.

*(1) **Combine stochastic MCTS with DNN***: To deal with the stochastic transitions in the UVR problem, this work combines the stochastic MCTS method with DNN, and extend the original AlphaZero algorithm to deal with the stochastic dynamics in the environment (distribution grid). The tree structure difference of the developed AlphaZero-UVR approach and the original AlphaZero algorithm is shown in Fig. 4. In the AlphaZero-UVR algorithm, there includes random nodes (contain state–action pair) and decision nodes (contain state information). The state information of decision nodes is defined as Eq. (3). The action space is defined in Eq. (4). Starting from the decision node, different random nodes are created by taking different actions (crew routing decisions). The action a is selected from the feasible action space which is determined by the current location of the vehicle and the road network. The uncertainties in the distribution grid will lead to various observations o (the power line passed through is damaged or not), then it results in different decision nodes. For each action a from decision node there is an edge that stores the statistics $\{N(S,a), W(S,a), Q(S,a), P(S,a), r(S,a)\}$, and for each observation o from random node there is an edge that stores the statistics $\{N(S,a,o)\}$. Note that $r(S,a)$ can be calculated by Eq. (10).

*(2) **MCTS simulation in AlphaZero-UVR***: Since the tree structure differences, the MCTS simulation, which includes *selection*, *extension*, and *backpropagation*, of the AlphaZero-UVR algorithm is different from the original AlphaZero algorithm. The authors designed the MCTS simulation procedures for the UVR problem, as shown in Fig. 5 and below:

Selection: Each simulation always starts from the root node S_0, and finishes when it reaches a leaf node S_L. For each hypothetical time-step $l = 1, 2, \dots, L$ of the selection stage, action a_l is selected according to the stored statistics for node S_{l-1} using the PUCT strategy:

$$a_l = \arg\max_a \left\{ Q(S,a) + c^{puct} P(S,a) \frac{\sqrt{\sum_b N(S,b)}}{1 + N(S,a)} \right\} \tag{11}$$

where, c^{puct} is a constant value. After taking the selected action, it reaches a random node (S_{l-1}, a_l). Then it generates a damage observation o of the passed line according to the current state information of the system, and it reaches a decision node. This process repeats until the final time-step L.

Expansion: At the final time-step L of the simulation, the state S_L and reward $R(S_{L-1}, a_L) = r_L(S_{L-1}, a_L)$ are respectively computed by the transition function and the reward function (10) according to S_{L-1}, a_L, and o_L. A new random node, corresponding to state–action (S_{L-1}, a_L) is added to the tree, and a new decision node, corresponding to state S_L is also added to the tree as shown in Fig. 5. Each edge (S_{L-1}, a, o) is initialized to:

$$\left\{ N(S_{L-1}, a_L, o) = 0 \right\} \tag{12}$$

In the same time, the statistics (p_a, v_L) of the new node S_L is computed by the neural network:

$$(p_a, v_L) = f_\theta(S_L) \tag{13}$$

where, θ represents the parameters of the neural network. This neural network takes system state as input and outputs action probabilities $\mathbf{p}$ with components $p_a = Pr(a|S)$ for each action a, and a scalar value v which represents the estimation of the expected outcome z from current state, as shown in Fig. 6. More specifically, using the DNN, features of the system state (including the physical state and belief state in Eq. (3)) are extracted by the multiple hidden layers to better evaluate the action probabilities and value estimated by the output layer. Each edge (S_L, a) is initialized to:

$$\left\{ N(S_L, a) = 0, W(S_L, a) = 0, Q(S_L, a) = 0, P(S_L, a) = p_a \right\} \tag{14}$$

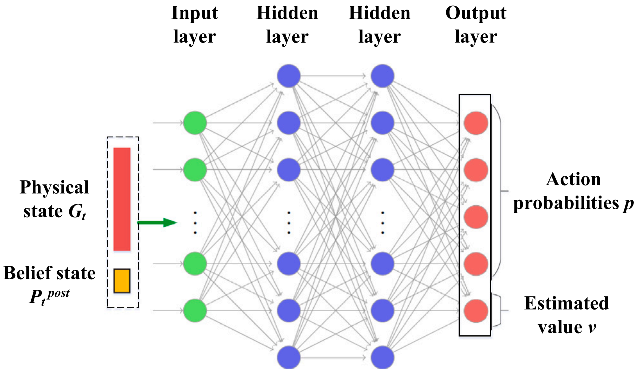

Fig. 6. The architecture of the deep neural network adopted by the AlphaZero-UVR algorithm.

Note that the reward information of each edge (S_L, a) is also initialized to zero and will be calculated using Eq. (10) when the agent visits the corresponding edge.

Backpropagation: At the end of the simulation, the statistics of the passed edges are updated back through the path. For $l = L, \dots, 1$, the statistics of the edges in the simulation paths are updated by:

$$\begin{aligned} &N(S_{l-1}, a_l) = N(S_{l-1}, a_l) + 1 \\ &N(S_{l-1}, a_l, o_l) = N(S_{l-1}, a_l, o_l) + 1 \\ &W(S_{l-1}, a_l) = W(S_{l-1}, a_l) + G_l \\ &Q(S_{l-1}, a_l) = \frac{W(S_{l-1}, a_l)}{N(S_{l-1}, a_l)} \end{aligned} \tag{15}$$

where G_l is the $(L - l)$-step estimate of the cumulative discounted reward, bootstrapping from v_L:

$$G_l = \sum_{\varsigma=0}^{L-1-l} \gamma^{\varsigma} r_{l+1+\varsigma} + \gamma^{L-l} v_L \tag{16}$$

where γ is the discount factor. $r_{l+1+\varsigma}$ represents the reward function at the hypothetical time-step $l+1+\varsigma$. v_L is the value computed in Eq. (13).

At each time-step t, starting from the root position S_t (used to generate the root node S_0), an MCTS search $\pi_t = \alpha_{\theta_{i-1}}(S_t)$ which consists of hundreds of simulations shown in (11)–(16) is executed using the previous iteration of neural network $f_{\theta_{i-1}}$. After finished the MCTS search, an action is sampled according to the obtained search probabilities π_t. The search probabilities π_t is actually a vector that represents the selection probability of each feasible action a_t when system in current state S_t, and the probability is proportional to the exponentiated visit count for each action:

$$\pi_{a_t} \propto N(S_t, a_t)^{\frac{1}{\tau}} \tag{17}$$

where τ is a temperature parameter.

(3) ***Immediate reward signal***: For board games, the agent does not receive the immediate reward signal at the intermediate time-steps. When game terminates at step T, then the game is scored to a final reward $r_T \in \{-1, 0, 1\}$ according to the rules of win/loss/draw. And the data of each time-step t, (S_t, π_t, z_t), is stored in the replay buffer, where $z_t = \pm r_T$. To this end, a game is played and the training data are generated by self-play.

However, for the UVR problem, there exists an immediate reward signal after the agent takes a decision, as shown in (10). Thus, it needs to design the value target z_t which corresponds to the cumulative reward function of the UVR problem. In this paper, the authors use the result of the MCTS search as a target value estimator [46], leveraging the action value estimates $Q(S_t, a)$ at the root state S_t. The designed target value estimation method is given in (18).

$$z_t = \max_a Q(S_t, a) \tag{18}$$

For each actual time-step t, the generated vehicle routing data (S_t, π_t, z_t) is stored in the replay buffer and used to train the neural network.

(4) ***Value function normalization***: The range of the value function V of the game of Go, shogi, and chess is $[-1, 1]$, which is very helpful for the algorithm convergence. However, the reward/value range of the UVR problem is $(-\inf, 0)$, as shown in Eq. (10). The unbounded value range will bring difficulties to the training of the AlphaZero-UVR algorithm and even causes the algorithm to diverge [47]. To deal with this problem, this work proposes to normalize Q value within $[0, 1]$ interval by using the minimum–maximum values observed in the search tree up to that point [43]. When a node is reached during the selection procedure, the normalized $\bar{Q}$ value is computed by:

$$\bar{Q}(S_t, a) = \frac{Q(S_t, a) - \min_{S,a\in Tree} Q(S, a)}{\max_{S,a\in Tree} Q(S, a) - \min_{S,a\in Tree} Q(S, a)} \tag{19}$$

Thus, in the above equation, the normalized value $\bar{Q}(S_t, a)$ is utilized to replace the original value $Q(S_t, a)$ in Eq. (11).

Finally, during the self-play process, new network parameters θ_i are trained in parallel by uniformly sampling data from the replay buffer. The new network $f_{\theta_i}(\cdot)$ is adjusted to minimize the following losses:

$$loss = (z - v)^2 - \pi^{\intercal} log\mathbf{p} + c\|\theta\|^2 \tag{20}$$

where, $\mathbf{p}$ and v are the policy and the value output of the new neural network $f_{\theta_i}(\cdot)$. θ represents the weights of the neural network. π and z are the sampled policy and value data from the reply buffer. c is a parameter to prevent overfitting.

3.3. Implementation details of the AlphaZero-UVR algorithm

The AlphaZero-UVR algorithm is trained off-line first to get a well-trained neural network model. Then, utilities can apply the well-trained agent to navigate the post-storm restoration of the same distribution system sequentially according to the actual state of the system. Using the well-trained model, the application process of the AlphaZero-UVR algorithm is shown in **Algorithm 1**. At each time-step, the EUC checks if there are scheduling requests from vehicles. If it receives any request, the agent will get the current state information of the system as shown in Eq. (3), and construct the root node using current state information. Then, starting from the root node, the agent conducts a predefined number of MCTS simulations to set up a search tree. And each MCTS simulation consists of selection, expansion, and backpropagation. Note that the well-trained neural network model will be used in expansion stage to calculate the action policy and value of the new node as shown in Eq. (13). After getting the search tree, the visiting times of each feasible routing actions of the root node can be easily obtained. The action with the highest visiting times is selected as the optimal routing action. Next, the vehicle travels to the destination node and gets the damage information of the power line it passed through. If the visited power line is damaged, the vehicle will repair it. The vehicle also needs to report the actual damage status of the visited line to the EUC. Finally, if current scheduling request queue is empty, the agent calculates the updated belief state of the distribution system using Eq. (1). The repairing process stops when the damage probability of each line is below a threshold ϵ^{thr}. It can be found that the optimal repair trajectory of the vehicles are calculated online according to the real-time status of the distribution system.

Algorithm 1 AlphaZero based Utility Vehicle Routing Approach.

```
Initialize the state $S_0$. Set time-step $t = 0$.
while $p(L^{e}_{t,i} = 1|H_t, A_{t-1}) \geq \epsilon^{thr}$ do:    ▷ Stop repairing the grid until the damage probability of each line is below a threshold.
    1. At time-step $t$, get the physical state (according to the scheduling requests from vehicles), belief state, and informational state of the distribution system defined in Eq. (3).
    2. Use MCTS with the learned neural network to solve Eq. (9) that determines the optimal action.    ▷ Perform a number of MCTS simulations to construct the search tree and the action with the highest visiting times is the optimal action.
    3. Move the utility vehicle, which sends the scheduling request to the EUC, according to the obtained optimal action, then get the observation of the damage status of the passed line.
    4. Delete the dispatched vehicle from current scheduling request queue. If the scheduling request queue is not empty, select the next vehicle from the queue and go back to step 1).
    5. Update the belief state of the distribution system using Eq. (1).
    6. $t = t + 1$.
end while
```

The neural network model used in the MCTS simulations needs to be trained off-line before the on-line application. **Algorithm 2** shows the training method of the AlphaZero-UVR approach. The training procedure consists of two parts, namely self-play and neural network updating. The neural network parameters are randomly initialized and will be updated by N loops. In each loop, the algorithm generates the training data (S_t, π_t, z_t) by self-play using the MCTS simulations with the latest neural network, then randomly sample data from the replay buffer, and update the neural network to minimize the losses shown in Eq. (20). Each self-play epoch starts from the first time-step, and moves forward until the end of the repairing process. At each time-step t, the optimal action a_t is sampled from the search tree constructed by M MCTS simulations, which repeats the selection (Eqs. (11) and (19)), expansion (Eq. (12)–Eq. (14)), and backpropagation (Eq. (15)–Eq. (16)) stages M times. In parallel, the neural network training procedure randomly samples data from the replay buffer, and trains the neural network using Eq. (20). The training process will be terminated when the algorithm converged.

4. Simulation results

In this section, the effectiveness of the AlphaZero-UVR algorithm is demonstrated by the numerical simulations on two distribution systems. All the simulations were conducted on an Intel Core i7 @1.90 GHz Windows-based PC with 16 GB RAM. This paper implemented the proposed AlphaZero-UVR algorithm using Tensorflow library in Python.

4.1. Case study I: 8-node test system

The first test distribution system used in the simulation is shown in Fig. 1. In this case study, the authors focus on the single utility vehicle routing problem which is a special case of multi-vehicle routing. It is assumed that the vehicle always depots from the substation. The prior damage probabilities of the power lines after a storm are given in Fig. 1, which can be estimated by storm weather information and operator's experience. In addition, the number of customers and the time for the vehicle passing through each road are also shown in the figure. The calling probability of customers when suffering from outages is set to $\rho = 5\%$. The stopping threshold ϵ^{thr} is set to 2%. To simplify the problem, the required repair time for each line is assumed to be 1 h. A four-layer neural network is adopted for this case study. The architecture of the neural network is given in Fig. 6, and the parameters of the network is provided in Table 1. Note that the maximum feasible actions of the vehicle in Fig. 1 is 4, so the action probabilities output by the neural network is a 4 dimensional vector.

Algorithm 2 Off-line Training of the AlphaZero-UVR Approach.

```
Randomly initialize the neural network. Set training index $n = 0$.
for $n \in 1, 2, \cdots, N$ do
    Reset the distribution system environment. Set $t = 1$ and $Terminal = False$.    ▷ Set the damaged line set according to the prior damage probabilities of the lines, and set the phone calls from customers according to topology of the system.
    Get the latest neural network parameters.
    while $(t \leq T)$ & $(Terminal = False)$ do:
        1) Get the vehicle index from the scheduling request queue, and generate the root node according to the current state of the distribution system $S_t$.
        2) Starting from the root node, perform $M$ MCTS simulations with the guidance of the latest neural network to construct a search tree.
        3) Get action probabilities $\pi_t$ and the target value $z_t$.    ▷ $\pi_t$ is computed according to the visiting counts of each action of the rode node using Eq. (17), $z_t$ is calculated using Eq. (18).
        4) Store the data $(S_t, \pi_t, z_t)$ to the replay buffer.
        5) Sample an action $a_t$ according to $\pi_t$.    ▷ (17).
        6) Execute the selected action, then calculate the reward $r_t$ using Eq. (10).
        7) If the scheduling request queue is not empty, randomly select a vehicle from the queue and go back to step 1).
        8) Get the observation of the passed lines and update the state of the system using Eq. (1) and Eq. (5) - Eq. (7).
        9) If the fault probabilities of all lines are below a threshold, set $Terminal = True$, else $Terminal = False$.
        10) If the repairing process terminated, calculate the actual customer-outage hours.    ▷ This is just used to plot the training convergence curve.
    end while
    Sample a mini-batch of data $B$ from replay buffer, and update neural network to minimize losses shown in Eq. (20).
end for
Output the well-trained neural network model.
```

Table 1
Neural network parameters/hyperparameters of case study I.

Parameters/Hyperparameters	Value
Number of layers	4
Neurons of input layer	8
Neurons of the 1st hidden layer	120
Neurons of the 2st hidden layer	120
Neurons of output layer	5
Learning rate	0.0001
Batch size (B)	32
Optimizer	RMSprop
Training steps	10,000

The number of simulations (M) of the MCTS search procedure in **Algorithm 2** affects the performance of the proposed algorithm. To analyze the sensitivity of the performance of the algorithm with respect to the number of simulations per move, the convergence performance of the proposed AlphaZero-UVR algorithm under different M value was tested, as shown in Fig. 7. It can be found that the cumulative outage hours of the customers decreases rapidly in the first 500 training steps, and then it slowly approaches to the optimal value. Besides, with more

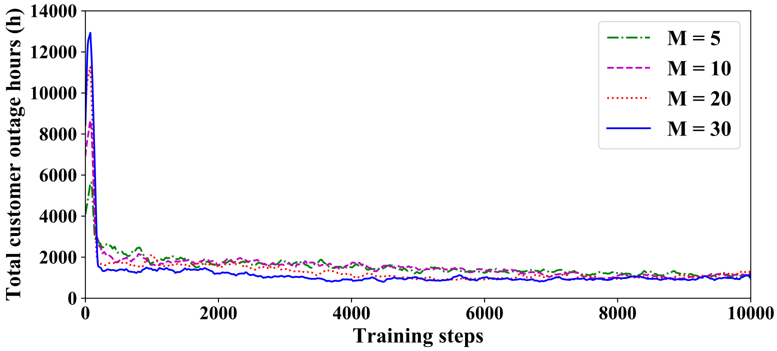

Fig. 7. The convergence process of the AlphaZero-UVR algorithm under different number of simulations per move.

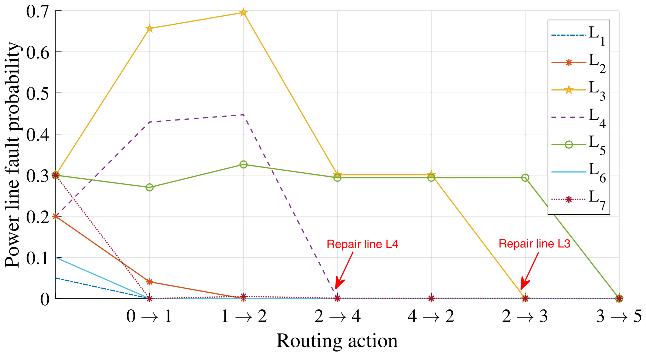

Fig. 8. Power line fault probabilities during the repairing procedure of case 6 in Table 2. '0 → 1' means the vehicle travels from node 0 to node 1.

simulations per move, the algorithm can achieve better optimization performance.

To validate the effectiveness of the proposed algorithm, the performance of the algorithm was compared with the fixed repair strategy, the traditional MCTS [37], and the OLUCT based method [38]. In this paper, the fixed repair strategy dispatches crews along the feeder and determines the repair trajectory based on the topology and customer distribution of the grid. For the distribution grid shown in Fig. 1, the designed fixed repair strategy is [0 ⟶ 1 ⟶ 6 ⟶ 2 ⟶ 1 ⟶ 2 ⟶ 4 ⟶ 2 ⟶ 3 ⟶ 5 ⟶ 3 ⟶ 2 ⟶ 6 ⟶ 7]. The optimization performance of the traditional tree search methods adopted in [37,38] are also influenced by the number of simulations per move. For the MCTS based comparing methods, the authors set M to 200 using which both methods perform very well, while the M value of the proposed algorithm is set to 30. The simulation results are shown in Table 2. In the table, the performance of the algorithm was tested using 10 different fault settings. For each case setting, the actual damaged lines are preset, which is unknown to the agent, and the phone calls (received or not) from customer node $[2,4,5,6,7]$ are also given. Fig. 8 shows the change process of power line fault probabilities of case 6 in Table 2. From the figure, the fault probabilities of power lines decrease stablely with the routing process until all failures are repaired. It is also noticed that the vehicle keeps searching faults after the faults at line L_4 and L_3 have been repaired (as marked in Fig. 8). This is because the fault probability of line L_5 is still greater than the stopping threshold (2%). Thus, after repaired L_3 and L_4, the vehicle started form node 3 and go to node 5 to check the fault status of line L_5. After confirming the faults probabilities of all lines below the threshold, the repairing process was ended.

The fixed repair strategy performs worst among all methods. This is because the fixed dispatch strategy cannot adapt its routing decisions according to the real-time updated state information of the grid. From the above results, it can be found that the proposed algorithm outperforms the other two tree search based methods even if the number of simulations per move of the proposed algorithm is much less than the other methods. The good performance of the proposed algorithm can be attributed to the guidance of the well-trained neural network model during the tree search process. Besides, compared with the other two tree search algorithms, the proposed algorithm has higher computational efficiency. The computational time required for a single time step scheduling of the proposed algorithm is 0.76 s, and the corresponding computational time of the traditional MCTS and the OLUCT methods are 6.7 s and 19 s, respectively.

4.2. Case study II: modified IEEE 123-node test system

To further validate the effectiveness of the proposed algorithm, the performance of the algorithm is tested on a modified IEEE 123-node distribution system, as shown in Fig. 9. It is assumed that the road paths are along each lines in the system. The locations of protective devices are placed according to Ref. [48], as shown in Fig. 9. The distribution system contains 123 nodes and 197 lines. The parameters of the modified IEEE 123-node system can be found in [49]. Ref. [49] also provided the length of each power line. In this work, the authors enlarged the length of each line to expand the coverage area of the distribution grid, and the average travel velocity of each vehicle is assumed to be 20 miles per hour. According to the topology of the test system, the system is divided into four zones, and each vehicle is responsible for one specific zone. The repair time for a damaged line in zone 1 and zone 3 is assumed to be 60 min, and the repair time for a damaged line in other zones is 120 min [28]. In addition, the vehicle 1 has the highest priority, and the vehicle 4 has the lowest priority. To decrease the computational complexity of the fault probability model shown in Eq. (1), the power lines of the same segment was aggregated and Monte Carlo simulation was adopted to obtain the approximated fault probabilities of the segments. After aggregation, the system contains 62 segments and 42 customer nodes.

The designed neural network model is a four-layer fully connected network. As the state information consists of the position information of the vehicle that should be scheduled immediately and the fault

Table 2
The optimized customer outage hour (h) using different algorithms for the 8-node test system.

Case	Case setting		AlphaZero-UVR		F_{MCTS} (hours)	F_{OLUCT} (hours)	F_{fixed} (hours)
	Phone calls	Damaged lines	$F_{AlphaZero-UVR}$(hours)	Repairing trajectory			
1	[0 0 1 0 0]	$[L_5]$	250.0	$0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 3 \longrightarrow 5$	250.0	250.0	416.7
2	[0 0 0 0 1]	$[L_7]$	383.3	$0 \longrightarrow 1 \longrightarrow 6 \longrightarrow 1 \longrightarrow 2 \longrightarrow 6 \longrightarrow 7$	516.67	516.67	583.3
3	[0 1 1 0 1]	$[L_3\ L_7]$	1016.7	$0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 3 \longrightarrow 5 \longrightarrow 3 \longrightarrow 2 \longrightarrow 4 \longrightarrow 2 \longrightarrow 6 \longrightarrow 7$	1516.7	1216.7	1450.0
4	[1 1 1 0 0]	$[L_2\ L_3]$	850.0	$0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 4 \longrightarrow 2 \longrightarrow 3 \longrightarrow 5$	716.66	716.66	1083.3
5	[1 1 1 0 0]	$[L_2\ L_4]$	683.3	$0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 4 \longrightarrow 2 \longrightarrow 3 \longrightarrow 5$	1016.7	850.0	916.7
6	[0 1 1 0 0]	$[L_3\ L_4]$	766.7	$0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 4 \longrightarrow 2 \longrightarrow 3 \longrightarrow 5$	766.7	766.7	966.7
7	[0 1 1 0 0]	$[L_4\ L_5]$	616.7	$0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 4 \longrightarrow 2 \longrightarrow 3 \longrightarrow 5$	1066.67	1066.67	816.7
8	[0 0 1 0 1]	$[L_5\ L_7]$	733.3	$0 \longrightarrow 1 \longrightarrow 6 \longrightarrow 7 \longrightarrow 6 \longrightarrow 1$	1266.7	1200.0	1100.0
9	[0 1 1 0 0]	$[L_3\ L_4\ L_5]$	900.0	$0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 4 \longrightarrow 2 \longrightarrow 6 \longrightarrow 7$	1133.0	1133.0	1283.3
10	[0 1 1 0 1]	$[L_3\ L_4\ L_7]$	1516.7	$0 \longrightarrow 1 \longrightarrow 2 \longrightarrow 4 \longrightarrow 2 \longrightarrow 3 \longrightarrow 5 \longrightarrow 3 \longrightarrow 5 \longrightarrow 3 \longrightarrow 2 \longrightarrow 6 \longrightarrow 7$	1616.7	1616.7	1750.0

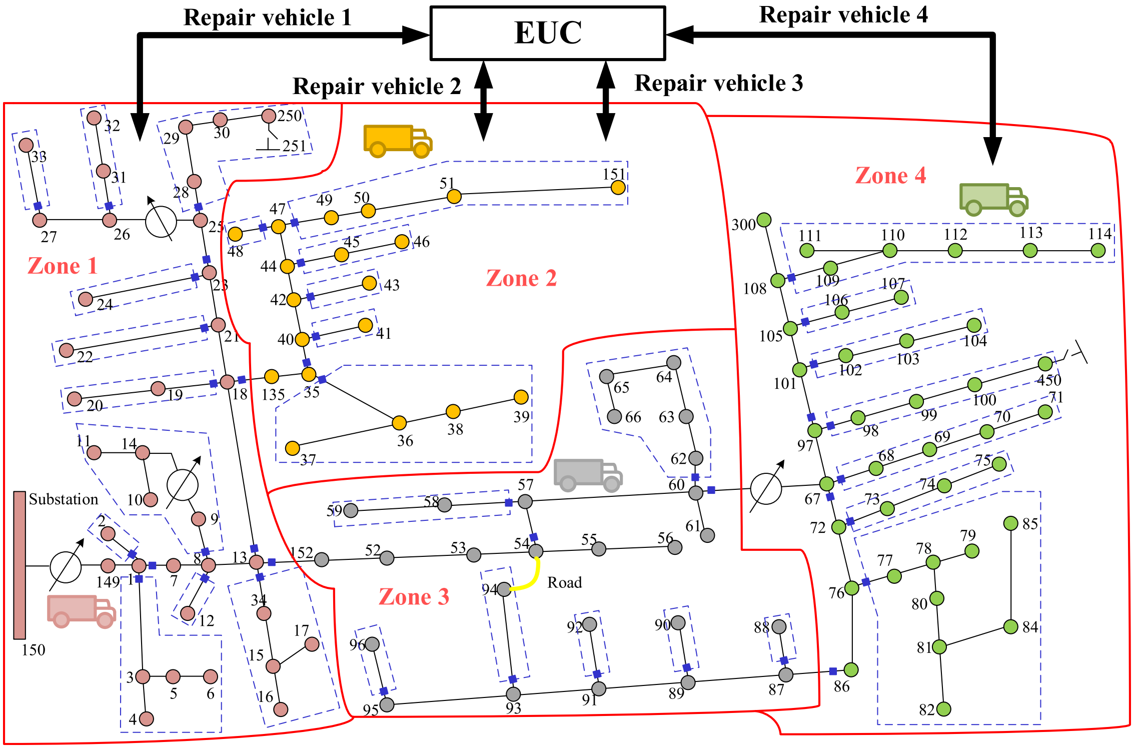

Fig. 9. The modified IEEE 123-node distribution system. The blue rectangles represent the locations of protective devices, and the dashed frame indicates the protection scope of the protective devices. The red line represents boundaries of each zone.

probability of each segment, the number of neurons of the input layer is 63. There are two hidden layers and each layer contains 150 neurons. Note that the maximum feasible actions of the vehicles in this case study is 3, so the action policy output by the neural network is a three-dimensional vector. Estimated value is also outputted by the neural network, so the output layer contains 4 neurons. The learning rate of the algorithm is set to 0.001, and all the other hyperparameters are the same with case study I.

In Fig. 10, the convergence process of the AlphaZero-UVR algorithm tested on the modified IEEE 123-node distribution system is given. It can be found that the algorithm converged after 7000 of training steps. It is also noticed that the total customer hours optimized by the algorithm was still oscillating slightly after trained by 5000 steps. This can be attributed to the fact that the agent was trained under a different scenario at each training step. Considering the actual damaged power lines and received customer calls are different in each training scenario, so the optimal total customer hours of each scenario is different. The authors also compared the performance of the proposed algorithm with the fixed repair strategy, the traditional MCTS approach, and the OLUCT method. The methods were compared under ten different cases, and each case was with different damage sets. The results are shown in Fig. 11. It can be found that the proposed method performs better than the comparing methods. The computational time required for a single time step scheduling of the proposed algorithm is 0.83 s. The results demonstrated the effectiveness of the proposed multi-vehicle routing algorithm.

From the results of the above two case studies, it can be found that the proposed post-storm UVR strategy can learn to find the repair routing for multiple utility vehicles and restore the damaged distribution grid efficiently based on the updated state information. Compared with fixed repair strategy and the traditional MCTS based methods, the total customer outage hours optimized by the proposed method is much lower. In addition, the computational time of the AlphaZero-UVR algorithm is acceptable for the real-time repair crew scheduling application.

5. Conclusion

In this work, an AlphaZero based post storm utility vehicle routing algorithm was proposed to guide repair crews in multiple vehicles to fix the damages in the distribution grid as fast as possible. The utility vehicle routing optimization problem was modeled as an MDP problem.

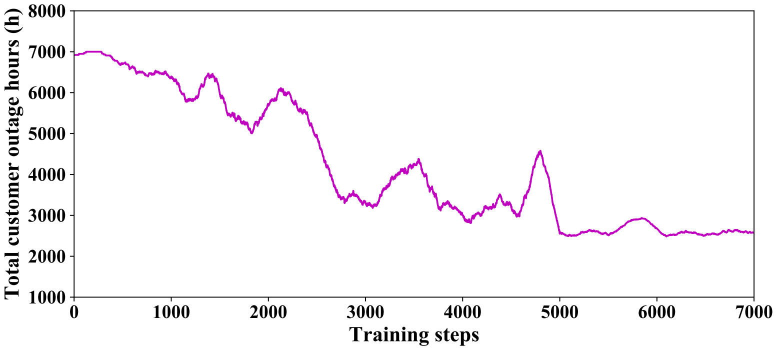

Fig. 10. The convergence process of the AlphaZero-UVR algorithm on the modified IEEE 123-node test system.

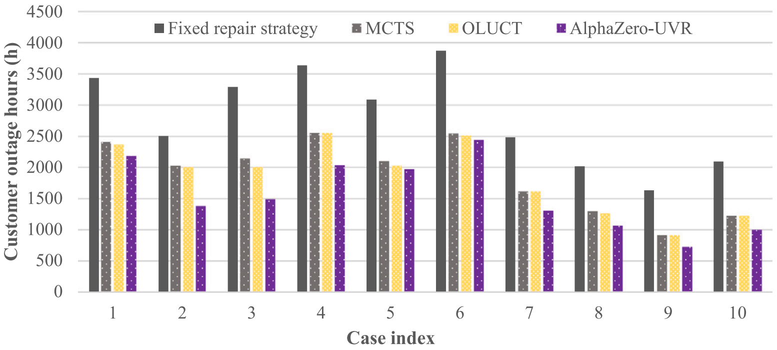

Fig. 11. The performance of the proposed algorithm and the comparing methods on the modified IEEE 123-node test system.

Then, the challenges of using the AlphaZero algorithm to solve the optimization problem in this work were presented and the corresponding solutions were proposed. The developed AlphaZero-UVR algorithm combined stochastic Monte-Carlo tree search (MCTS) with deep neural network, and can teach itself to navigate the crews to restore the distribution grid through self-play. To validate the effectiveness of the proposed algorithm, the performance of the algorithm was tested by numerical simulations on a 8-node distribution grid and a modified IEEE 123-node distribution grid. Simulation results demonstrated that the proposed algorithm outperforms traditional MCTS based methods.

Many issues need to be addressed for a more efficient restoration of distribution grids after extreme events. For instance, cooperation between different vehicles can significantly improve the damage repair efficiency. Combining muti-agent deep reinforcement learning with MCTS will be investigated in future work. Besides, traditional fully connected neural networks were adopted in the designed AlphaZero based UVR approach. Considering features of a damaged distribution grid can be well represented and learned by graph neural networks (GNNs), GNNs based AlphaZero-UVR algorithm will be studied in future work.

CRediT authorship contribution statement

Hang Shuai: Concept development, Algorithm development, Algorithm implementation, Writing – original draft. **Fangxing Li:** Concept development, Review & editing, Technical supervision, Funding acquisition. **Buxin She:** Algorithm development, Review & editing. **Xiaofei Wang:** Algorithm development, Review & editing. **Jin Zhao:** Algorithm development, Review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

The research was supported in part by U.S. National Science Foundation (NSF) Award ECCS-1809458 and in part by NSF Award ECCS-2033910.

References

[1] Perera A, Nik VM, Chen D, Scartezzini J-L, Hong T. Quantifying the impacts of climate change and extreme climate events on energy systems. Nat Energy 2020;5(2):150–9.
[2] Kenward A, Raja U. Blackout: Extreme weather climate change and power outages. Climate Central 2014;10:1–23.
[3] Tari AN, Sepasian MS, Kenari MT. Resilience assessment and improvement of distribution networks against extreme weather events. Int J Electr Power Energy Syst 2021;125:106414.
[4] Liu Y, Fan R, Terzija V. Power system restoration: a literature review from 2006 to 2016. J Mod Power Syst Clean Energy 2016;4(3):332–41.
[5] Shi Q, Liu W, Zeng B, Hui H, Li F. Enhancing distribution system resilience against extreme weather events: Concept review, algorithm summary, and future vision. Int J Electr Power Energy Syst 2022;138:107860.
[6] Firoozjaee MG, Sheikh-El-Eslami MK. A hybrid resilient static power system expansion planning framework. Int J Electr Power Energy Syst 2021;133:107234.
[7] Jiang Y, Chen S, Liu C-C, Sun W, Luo X, Liu S, Bhatt N, Uppalapati S, Forcum D. Blackstart capability planning for power system restoration. Int J Electr Power Energy Syst 2017;86:127–37.
[8] Smith AM, Pósfai M, Rohden M, González AD, Dueñas Osorio L, D'Souza RM. Competitive percolation strategies for network recovery. Sci Rep-UK 2019;9(1):1–12.
[9] Directive PP. Presidential policy directive—Critical infrastructure security and resilience. National Arch Rec Adm Retriev Dec 2013;24:2021.
[10] Zhang B, Wang M, Su W. Reliability assessment of converter-dominated power systems using variance-based global sensitivity analysis. IEEE Open Access J Power Energy 2021;8:248–57.
[11] Peyghami S, Palensky P, Fotuhi-Firuzabad M, Blaabjerg F. System-level design for reliability and maintenance scheduling in modern power electronic-based power systems. IEEE Open Access J Power Energy 2020;7:414–29.

[12] Wu H, Xie Y, Xu Y, Wu Q, Yu C, Sun J. Resilient scheduling of MESSs and RCs for distribution system restoration considering the forced cut-off of wind power. Energy 2022;123081.

[13] Yan J, Hu B, Xie K, Tai H-M, Li W. Post-disaster power system restoration planning considering sequence dependent repairing period. Int J Electr Power Energy Syst 2020;117:105612.

[14] Guikema SD, Nateghi R, Quiring SM, Staid A, Reilly AC, Gao M. Predicting hurricane power outages to support storm response planning. IEEE Access 2014;2:1364–73.

[15] Yuan W, Wang J, Qiu F, Chen C, Kang C, Zeng B. Robust optimization-based resilient distribution network planning against natural disasters. IEEE Trans Smart Grid 2016;7(6):2817–26.

[16] Amirioun M, Aminifar F, Lesani H. Resilience-oriented proactive management of microgrids against windstorms. IEEE Trans Power Syst 2017;33(4):4275–84.

[17] Qiu F, Wang J, Chen C, Tong J. Optimal black start resource allocation. IEEE Trans Power Syst 2015;31(3):2493–4.

[18] Yan M, He Y, Shahidehpour M, Ai X, Li Z, Wen J. Coordinated regional-district operation of integrated energy systems for resilience enhancement in natural disasters. IEEE Trans Smart Grid 2018;10(5):4881–92.

[19] Trakas DN, Hatziargyriou ND. Optimal distribution system operation for enhancing resilience against wildfires. IEEE Trans Power Syst 2017;33(2):2260–71.

[20] Hou Y, Liu C-C, Sun K, Zhang P, Liu S, Mizumura D. Computation of milestones for decision support during system restoration. In: 2011 IEEE power and energy society general meeting. IEEE; 2011, p. 1–10.

[21] Sun W, Liu C-C, Zhang L. Optimal generator start-up strategy for bulk power system restoration. IEEE Trans Power Syst 2010;26(3):1357–66.

[22] Arif A, Ma S, Wang Z, Wang J, Ryan SM, Chen C. Optimizing service restoration in distribution systems with uncertain repair time and demand. IEEE Trans Power Syst 2018;33(6):6828–38.

[23] Shi Q, Li F, Olama M, Dong J, Xue Y, Starke M, Feng W, Winstead C, Kuruganti T. Post-extreme-event restoration using linear topological constraints and DER scheduling to enhance distribution system resilience. Int J Electr Power Energy Syst 2021;131:107029.

[24] Xie Y, Chen X, Wu Q, Zhou Q. Second-order conic programming model for load restoration considering uncertainty of load increment based on information gap decision theory. Int J Electr Power Energy Syst 2019;105:151–8.

[25] Coffrin C, Van Hentenryck P. Transmission system restoration with co-optimization of repairs, load pickups, and generation dispatch. Int J Electr Power Energy Syst 2015;72:144–54.

[26] Van Hentenryck P, Coffrin C, Bent R, et al. Vehicle routing for the last mile of power system restoration. In: Proceedings of the 17th power systems computation conference, stockholm, sweden. Citeseer; 2011.

[27] Arif A, Wang Z, Wang J, Chen C. Power distribution system outage management with co-optimization of repairs, reconfiguration, and DG dispatch. IEEE Trans Smart Grid 2018;9(5):4109–18. http://dx.doi.org/10.1109/TSG.2017.2650917.

[28] Ding T, Wang Z, Jia W, Chen B, Chen C, Shahidehpour M. Multiperiod distribution system restoration with routing repair crews, mobile electric vehicles, and soft-open-point networked microgrids. IEEE Trans Smart Grid 2020;11(6):4795–808. http://dx.doi.org/10.1109/TSG.2020.3001952.

[29] Lei S, Chen C, Li Y, Hou Y. Resilient disaster recovery logistics of distribution systems: Co-optimize service restoration with repair crew and mobile power source dispatch. IEEE Trans Smart Grid 2019;10(6):6187–202. http://dx.doi.org/10.1109/TSG.2019.2899353.

[30] Chen C, Wang J, Ton D. Modernizing distribution system restoration to achieve grid resiliency against extreme weather events: an integrated solution. Proc IEEE 2017;105(7):1267–88.

[31] Sun H, Wang Z, Wang J, Huang Z, Carrington N, Liao J. Data-driven power outage detection by social sensors. IEEE Trans Smart Grid 2016;7(5):2516–24. http://dx.doi.org/10.1109/TSG.2016.2546181.

[32] Understanding what “Damage Assessment" means. 2022.

[33] Liu Y, Schulz NN. Knowledge-based system for distribution system outage locating using comprehensive information. IEEE Trans Power Syst 2002;17(2):451–6. http://dx.doi.org/10.1109/TPWRS.2002.1007917.

[34] Sridharan K, Schulz NN. Outage management through AMR systems using an intelligent data filter. IEEE Trans Power Deliver 2001;16(4):669–75. http://dx.doi.org/10.1109/61.956755.

[35] USDepartment of Energy. Economic benefits of increasing electric grid resilience to weather outages. Exec Off Pres 2013.

[36] Bahmanyar A, Jamali S, Estebsari A, Bompard E. A comparison framework for distribution system outage and fault location methods. Electr Pow Syst Res 2017;145:19–34.

[37] Al-Kanj L, Powell WB, Bouzaiene-Ayari B. The information-collecting vehicle routing problem: Stochastic optimization for emergency storm response. 2016, arXiv preprint arXiv:1605.05711.

[38] Shuai H, He H, Wen J. Post-storm vehicle routing for distribution grid restoration: An OLUCT based learning approach. In: 2020 IEEE power and energy society general meeting. IEEE; 2020, p. 1–5.

[39] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, et al. Mastering the game of go with deep neural networks and tree search. Nature 2016;529(7587):484.

[40] Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A, et al. Mastering the game of go without human knowledge. Nature 2017;550(7676):354–9.

[41] Kou X, Du Y, Li F, Pulgar-Painemal H, Zandi H, Dong J, Olama MM. Model-based and data-driven HVAC control strategies for residential demand response. IEEE Open Access J Power Energy 2021;8:186–97.

[42] Li F, Du Y. From AlphaGo to power system AI: What engineers can learn from solving the most complex board game. IEEE Power Energy Mag 2018;16(2):76–84. http://dx.doi.org/10.1109/MPE.2017.2779554.

[43] Schrittwieser J, Antonoglou I, Hubert T, Simonyan K, Sifre L, Schmitt S, Guez A, Lockhart E, Hassabis D, Graepel T, et al. Mastering atari, go, chess and shogi by planning with a learned model. Nature 2020;588(7839):604–9.

[44] Shuai H, He H. Online scheduling of a residential microgrid via Monte-Carlo tree search and a learned model. IEEE Trans Smart Grid 2021;12(2):1073–87. http://dx.doi.org/10.1109/TSG.2020.3035127.

[45] Al-Kanj L, Bouzaiene-Ayari B, Powell WB. A probability model for grid faults using incomplete information. IEEE Trans Smart Grid 2015;8(2):956–68.

[46] Moerland TM, Broekens J, Plaat A, Jonker CM. A0C: Alpha zero in continuous action space. 2018, arXiv preprint arXiv:1805.09613.

[47] Schadd MP, Winands MH, Van Den Herik HJ, Chaslot GM-B, Uiterwijk JW. Single-player Monte-Carlo tree search. In: International conference on computers and games. Springer; 2008, p. 1–12.

[48] Butler-Purry KL, Funmilayo HB. Overcurrent protection issues for radial distribution systems with distributed generators. In: 2009 IEEE power & energy society general meeting. IEEE; 2009, p. 1–5.

[49] IEEE 123 node test feeder. 2021, URL https://site.ieee.org/pes-testfeeders/resources/ [Accessed 5 Febrary 2021].