

Qiang FENG[†], Qilong WU[†], Xingshuo HAI, Yi REN, Changyun WEN, Zili WANG

Deep reinforcement learning-based resilience optimization for infrastructure networks restoration with multiple crews

© Higher Education Press 2025

Abstract Restoration of infrastructure networks (INs) following large disruptions has received much attention lately due to examples of massive localized attacks. Within this challenge are two complex but critical problems: repair route identification and optimizing the sequence of the repair actions for resilience improvement. Existing approaches have not, however, given due consideration to globally optimal enhancement in resilience, especially with multiple repair crews that have uneven capacities. To address this gap, this paper focuses on a resilience optimization (RO) strategy for coordinating multiple crews. The objective is to determine the optimal routes for each crew and the best sequence of repairs for damaged nodes and links. Given the two-layered decision-making required—coordinating between multiple crews and optimizing each crew’s actions—this study develops a deep reinforcement learning (DRL) framework. The framework leverages an actor-critic neural network that processes IN damage data and guides Monte Carlo tree search (MCTS) to identify optimal repair routes and actions for each crew. A case study based on the 228-node power grid, simulated using Python, demonstrates that the proposed DRL approach effectively supports restoration decision-making.

Keywords Infrastructure network restoration, Deep reinforcement learning, Resilience optimization, Repair routing, Multiple crews

1 Introduction

The functionality of modern cities relies heavily on infrastructure networks (INs), such as those for water, power, and transportation. These networks play crucial roles in improving economic prosperity and enabling the movement of resources (Carnevali et al., 2020; Xu and Chopra, 2022; Xie et al., 2023; Dui et al., 2024a). However, disasters such as earthquakes and typhoons can damage the components (nodes and links) of INs (Lu et al., 2018; Liu and Wang, 2021). With increasing concerns related to attacks on the cyber and physical components associated with networks, as well as the need to mitigate the effects of natural disasters on networks, a variety of postdisaster restoration countermeasures (e.g., monitoring, inspection, and repair) have been proposed in the literature (Fu et al., 2022; Ganganath et al., 2018; Wang et al., 2021; Yu et al., 2017; Dui et al., 2024b).

Resilience has emerged as a critical and desirable characteristic, and there is an abundance of detailed reviews and comparative analyses of qualitative frameworks, as well as quantitative metrics for studying resilience (Bai et al., 2021; Zuo, 2021; Si et al., 2020; Das et al., 2020; Ahmadian et al., 2020; Dui et al., 2024d). Moreover, two fundamental yet challenging issues in postdisaster restoration are repair sequence (Xu et al., 2022; Zhang et al., 2022; Wu et al., 2022; Du and Wu, 2022) and routing (Ruan et al., 2021; Allal et al., 2021; Díaz-Ramírez et al., 2014) decisions. Network resilience can be considered an optimization objective to recover network functionality within a finite repair time.

To increase network resilience, various models for emergency repair sequence decision-making have been proposed in the field of postdisaster restoration (Xu et al., 2022; Zhang et al., 2022; Wu et al., 2022; Du and Wu,

Received May 31, 2024; revised Aug. 6, 2024; accepted Aug. 11, 2024

Qiang FENG, Yi REN (✉), Zili WANG
School of Reliability and Systems Engineering, Beihang University,
Beijing 100191, China
E-mail: renyi@buaa.edu.cn

Qilong WU
Hangzhou International Innovation Institute, Beihang University,
Hangzhou 311115, China

Xingshuo HAI
School of Reliability and Systems Engineering, Beihang University,
Beijing 100191, China; School of Electrical and Electronic Engineer-
ing, Nanyang Technological University, Singapore 639798, Singapore

Changyun WEN
School of Electrical and Electronic Engineering, Nanyang Technological
University, Singapore 639798, Singapore

[†]These authors are co-first authors with equal contribution to this work.

2022). For example, Xu et al. (2022) developed a deterministic formulation of the repair sequencing decision problem for postdisaster INs under uncertain conditions. They proposed a two-stage stochastic model to solve the problem effectively, utilizing a scenario generation and reduction method to generate a limited number of repair time scenarios. Moreover, they proposed a heuristic algorithm for managing large-scale disruptions and incorporated an efficient enumeration algorithm as a module for addressing small-scale disruptions. Zhang et al. (2022) studied the optimization of restoration schedules for damaged highway-bridge networks, focusing on maximizing a resilience measure associated with network travel time. They formulated restoration-scheduling problems as integer programs to determine optimal schedules. Wu et al. (2022) proposed a novel method, which is based on a heuristic search algorithm and graph theory, for devising generator start-up sequences to select a feasible scheme with maximal power generation for system restoration strategies. Finally, in the context of utility grid outages, Du and Wu (2022) presented a two-stage learning framework to identify optimal restoration strategies that improve system resilience.

In contrast to resilience-oriented repair sequence decision-making strategies, repair routing strategies focus on optimizing routes for repairing damage. Several repair routing strategies have been proposed for various applications (Ruan et al., 2021; Allal et al., 2021; Díaz-Ramírez et al., 2014; Dui et al., 2024c). For example, Ruan et al. (2021) studied an operational aircraft repair routing problem and developed an integer linear programming framework based on network flow. They also developed a reinforcement learning-based algorithm that considers factors such as flying hours, workforce capacity, and the number of take-offs between repair checks. Allal et al. (2021) developed a simulation-optimization approach for routing and scheduling wind turbine repairs in offshore wind farms, utilizing the ant colony system (ACS) algorithm. Díaz-Ramírez et al. (2014) proposed scheduling planning methods for airlines with a single fleet and single repair and crew base, addressing aircraft repair routing and crew scheduling problems. Dui et al. (2024c) integrated repair strategies into unmanned vehicle distribution networks while considering cascading failures and proposed a resilience optimization (RO) algorithm.

However, limited attention has been given to RO strategies for IN restoration with multiple crews (Dui et al., 2023). For a single crew, the RO strategy takes into account both repair sequence and routing decisions. However, for multiple crews, the focus is on developing a global optimal strategy that considers the coupling characteristics between each crew's repair actions. Therefore, this paper adopts a two-layer decision-making architecture for enhancing resilience, determining optimal routes for multiple crews to repair damage, and determining the optimal sequence for each crew to repair damaged nodes

and links.

In recent years, significant advancements have been made in the theory and practice of deep reinforcement learning (DRL) (Arulkumaran et al., 2017; Silver et al., 2017; Wu et al., 2021). DRL leverages deep neural networks (DNNs) to develop artificial agents capable of efficiently processing high-dimensional sensory inputs and leveraging past experiences to make informed decisions. One popular algorithm in this domain is the deep Q-network (DQN), which employs a value-function-based DRL approach to improve an agent's decision-making capabilities (Arulkumaran et al., 2017). For repair sequence decisions, the DQN method effectively utilizes degradation and damage information to support optimal strategies (Fan et al., 2022). In light of managing multiple crews, Pinciroli et al. (2021) proposed a novel sequential decision problem formulation for the operation and repair optimization of wind turbines over a long-term horizon via DRL based on proximal policy optimization (PPO). Additionally, methods such as deep deterministic policy gradient (DDPG) and double-DQN have been utilized to develop optimal maintenance schedules and plans for large-scale systems while minimizing costs (Ogunfowora and Najjaran, 2023). However, in the context of RO strategies, it is essential to consider both the repair sequence and repair routing decisions. The repair routing decision must consider the delayed rewards of future repair actions. Monte Carlo tree search (MCTS) (Browne et al., 2012) can estimate the repair action value for each damage state within a search tree by employing Monte Carlo rollouts. To calculate long-delay rewards accurately, the search tree expands significantly through rollout simulations.

To address the dual-layer decision-making characteristics of RO between multiple crews and a single crew, this paper proposes a DRL framework for IN restoration. Within this framework, an RO agent uses an actor-critic neural network (ACNN) to guide the MCTS. The actor head of the ACNN focuses the MCTS search on high-probability actions, whereas the critical head evaluates damage states within the search tree. The paper's motivations and contributions are as follows:

- (1) An RO problem involving multiple crews for IN restoration is proposed, which combines repair sequences and routing decisions.

- (2) The adoption of a DRL framework facilitates the global optimal RO strategy for resilience enhancement, considering the dual-layer decision-making between multiple crews and a single crew.

- (3) An actor-critic MCTS (AC-MCTS) method uses an ACNN to guide the MCTS search. The ACNN effectively utilizes IN damage information, allowing the MCTS to determine optimal repair routes and actions for multiple crews.

The subsequent sections of this paper are organized as follows: Section 2 presents the RO problem formulation.

Section 3 explains the DRL-based RO method. Section 4 presents a case study, and Section 5 concludes with closing remarks.

2 Problem formulation

2.1 Formulation of the infrastructure network and its damage and repair

2.1.1 Operation

An IN is represented as an undirected graph \mathbf{G} , consisting of N nodes and K edges. The graph \mathbf{G} is considered sparse [$K \ll N(N-1)/2$]. It is also assumed to be connected, meaning that there is at least one path connecting any two nodes via a finite number of steps. To describe this IN, two matrices are needed: an adjacency matrix \mathbf{A} and a matrix \mathbf{L} representing the physical distances. The $N \times N$ adjacency matrix \mathbf{A} has elements a_{ij} equal to 1 if there is an edge connecting node i to node j and 0 otherwise. The matrix \mathbf{L} contains elements l_{ij} representing the geographic distances between nodes i and j , even when there is no direct edge between i and j . The physical distance matrix \mathbf{L} can be calculated using the latitudes and longitudes of the IN nodes.

The shortest path length d_{ij} between nodes i and j is defined as the smallest sum of physical distances among all possible paths in the IN from node i to node j . Once the adjacency matrix \mathbf{A} and physical distance matrix \mathbf{L} are known, they can be used to compute the matrix \mathbf{D} representing the shortest path lengths. The efficiency ε_{ij} between nodes i and j can be expressed as follows:

$$\varepsilon_{ij} = w_i w_j \frac{1}{d_{ij}}, \quad (1)$$

where w_i is the weight of node i , and in this paper, the node degree is adopted as the weight. When there is no path in the IN between i and j under localized attacks, $d_{ij} = +\infty$ and $\varepsilon_{ij} = 0$. The average efficiency of \mathbf{G} can be expressed as follows:

$$E(\mathbf{G}) = \frac{1}{N(N-1)} \sum_{i \neq j \in \mathbf{G}} \varepsilon_{ij}. \quad (2)$$

2.1.2 Damage

Figure 1 shows an example of an IN and its corresponding graph \mathbf{G} . The damage center and radius are randomly generated. The localized attack area is highlighted in pink in Fig. 1. It is assumed that all nodes and edges within this pink area fail. The geographically localized edges in the IN are removed, and the pink nodes are isolated. Therefore, the state of the IN edges can be described by the current adjacency matrix \mathbf{A}_d after the localized attack,

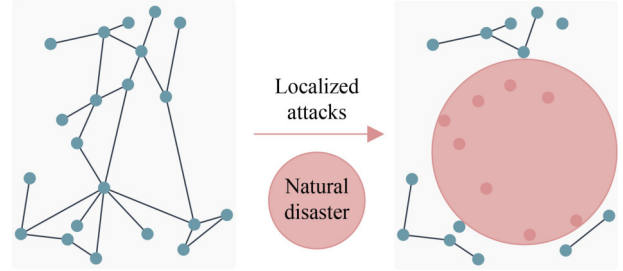


Fig. 1 Localized attacks.

and the state vector of the IN nodes can be described as follows:

$$\mathbf{S}_d^{\text{nodes}} = [s_1^{\text{node}}, s_2^{\text{node}}, \dots, s_i^{\text{node}}, \dots, s_j^{\text{node}}, \dots, s_N^{\text{node}}], \quad (3)$$

where s_i^{node} is 0 if node i is damaged and is 1 otherwise.

2.1.3 Repair

This paper proposes a comprehensive and actionable strategy to support repair activities in the IN. The strategy considers various factors, such as the repair team, repair time sequence, and repairing objects. The aim is to address the limitations of existing studies that focus primarily on repairing objects alone.

The repair activities are carried out in parallel by multiple repair crews. The mean time taken to repair a destroyed node i is denoted by $MTTR_i$. The time required to repair each edge can be calculated via the following equation:

$$RT_{ij} = l_{ij}/v_{\text{edge}}, \quad (4)$$

where v_{edge} denotes the repair speed of the damage edge. The time to move from i to j can be expressed as follows:

$$MT_{ij} = d_{ij}/v_{\text{move}}, \quad (5)$$

where v_{move} denotes the movement speed of the crew.

2.2 Resilience of infrastructure networks

Resilience is commonly used to assess the recovery performance of a system following localized attacks (Bai et al., 2021; Si et al., 2020; Das et al., 2020). This paper introduces the concept of resilience in INs to evaluate the effectiveness of restoration strategies, aiming to enhance network performance and reduce recovery time.

To quantitatively measure the resilience of an IN, it is vital to define a metric that represents the functional performance of the network. This metric is referred to as the figure of merit (FOM) within a widely adopted framework (Das et al., 2020). In this paper, the average efficiency $E(\mathbf{G})$ is normalized as follows to define the FOM at time τ :

$$FOM(\tau) = \frac{E_r(G)}{E^*(G)}, \quad (6)$$

where $E^*(G)$ is the average efficiency before the occurrence of a localized attack event and where $E_r(G)$ is the average efficiency at time τ . Furthermore, this definition of the FOM can also use other performance parameters of the IN.

A localized attack event is assumed to occur at time τ_d , causing a significant degradation in the functional performance of the IN. τ_d is a random time, and the RO strategy should be developed on the basis of damage information and involve multiple crews (explained in Section 3). These crews execute the restoration strategy to recover the IN from the localized attack, after which the functional performance of the IN improves and reaches the FOM threshold FOM_{thr} at time τ_r . The following metric of resilience is adopted to quantify the loss incurred by the IN due to a localized attack.

$$RL = \int_{\tau_d}^{\tau_r} [FOM_{thr} - FOM(\tau)] d\tau. \quad (7)$$

2.3 Formulation of resilience optimization

The restoration of the IN under localized attacks is treated as an RO problem involving multiple crews. The recovery time and recovery level are the constraints, the resilience loss (RL) is the objective, and the repair actions are the basic decision variables. The goal of this RO problem is to minimize the RL with a finite recovery time and a predefined recovery level.

Thus, the RO problem with multiple crews can be expressed as follows:

$$\min RL, \quad (8)$$

$$\text{s.t. } a_{t_c}^* \in \Omega_c | c \in \mathcal{F}, t_c \in \mathcal{T}_c, \quad (9)$$

$$a_t^* \in \Omega | t \in \mathcal{T}, \quad (10)$$

$$\Omega_c \subset \Omega, \forall c \in \mathcal{F}, \quad (11)$$

$$\sum_{t_c \in \mathcal{T}_c} \tau(a_{t_c}^*) \leq \tau_r - \tau_d \leq \tau_{thr}, \forall c \in \mathcal{F}, \quad (12)$$

$$FOM(\tau_r) \geq FOM_{thr}, \quad (13)$$

where c is the crew number and $\mathcal{F} = \{1, 2, \dots, n_{cre}\}$ is the crew fleet set with n_{cre} crews, t_c is the repair time step of crew c , $\mathcal{T}_c = \{1, 2, \dots, T_c\}$ is the time step set during its own repair process, $a_{t_c}^*$ represents the repair action of time step t_c , and Ω_c is the action sequence set of crew c . In Constraint (10), t is the repair time step of this crew fleet, $\mathcal{T} = \{1, 2, \dots, T\}$ is the fleet time step set, a_t^* represents the repair action of time step t , and Ω is the fleet

action sequence set. Constraint (11) indicates the relationship between the fleet action sequence set Ω and each crew action sequence set Ω_c . In Constraint (12), $\tau(a_{t_c}^*)$ represents the time interval to execute the action $a_{t_c}^*$, and τ_{thr} is the recovery time threshold. Constraint (12) forces the total repair time of the crew fleet to be lower than the threshold recovery time, whereas Constraint (13) forces the IN performance to reach the FOM threshold at time τ_r .

This RO problem is a decision-making problem, and the RO strategy consists of a sequence of repair actions. The multiple crews execute repair actions in this RO strategy to improve the IN performance and reach the FOM threshold at time τ_r . Different RO strategies have different values of RL and τ_r . The recovery time τ_r can be calculated according to Section 2.1.3, whereas RL can be calculated according to Eq. (7). The optimal RO strategy aims to minimize the values of RL and τ_r , which depend on the repair sequence and routing decisions.

3 Deep-reinforcement learning-based resilience optimization method

3.1 Deep reinforcement learning framework for resilience optimization

For the RO problem, a DRL framework is developed in this paper, as shown in Fig. 2. The framework illustrates the multiple-crew repair process (MCRP), the multiple-crew time sequence, resilience and reward, and neural network training. The details of the four parts of the framework are explained in Sections 3.1.1–3.1.4.

3.1.1 Multiple crew repair process

Before a localized attack, the IN has an adjacency matrix A . Then, a localized attack is randomly initialized with a damage center and a damage radius at time τ_d . After that, the damage adjacency matrix A_d and node vector S_d^{nodes} are generated. $FOM(\tau_d)$ can be calculated according to Eq. (6).

To recover the IN, a crew fleet with n_{cre} crews executes a sequence of repair actions during an MCRP. A two-layer decision-making architecture is developed for this MCRP, consisting of a fleet-level action sequence set and n_{cre} crew-level action sequence sets. The fleet-level action sequence set is defined as:

$$\Omega = \{a_{t_c}^* | t = 1 \text{ to } T, c \in \mathcal{F}, t_c \in \mathcal{T}_c. \quad (14)$$

The crew-level action sequence set of crew c can be described as follows:

$$\Omega_c = \{a_{t_c}^* | t_c = 1 \text{ to } T_c, t \in \mathcal{T}. \quad (15)$$

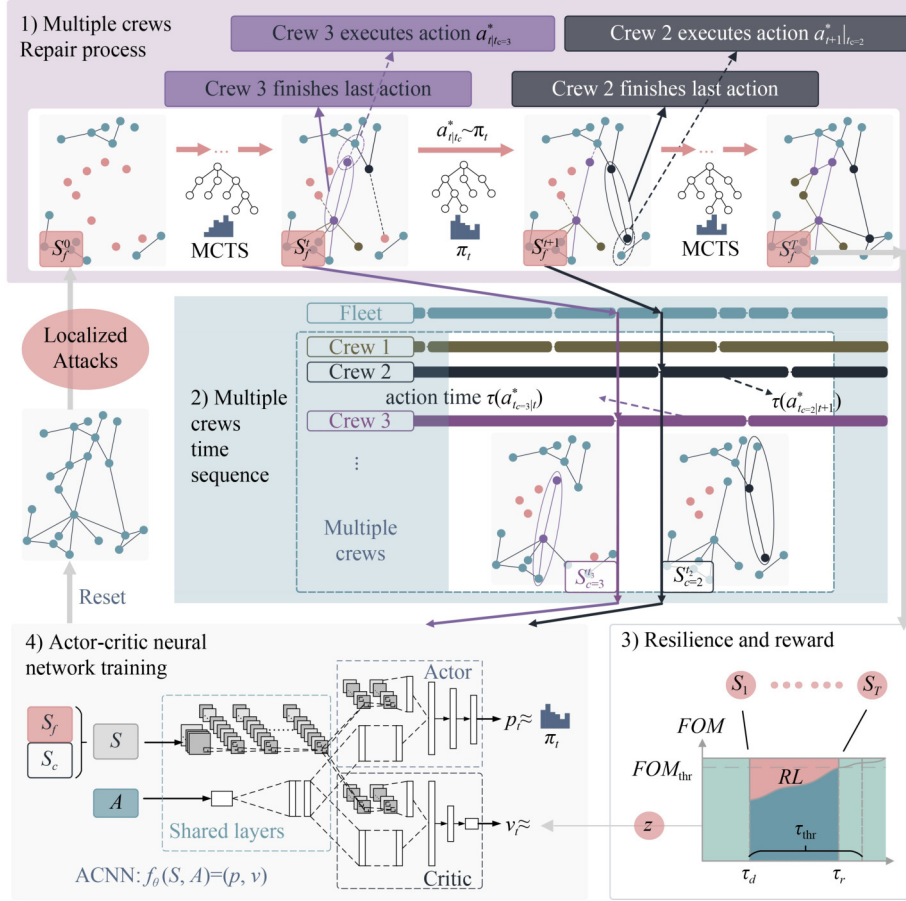


Fig. 2 Deep reinforcement learning framework for IN restoration.

The repair action is defined as $a_{t|t_c}^* = a_{t_c|t}^* = (i, j)$, which means that this crew c repairs the edge from the normal node i to node j . After repairing the edge, if node j is a damaged node, this crew then continues to repair this node; otherwise, it continues to execute its next action. At each time step t , an RO agent uses a search algorithm with an integrated ACNN to select a repair action $a_{t|t_c}^*$ for crew c , which has just finished its last action.

The decision-making of this agent (more details of which are given in Section 3.2) can be described as follows:

$$f_{\text{agent}}(\mathbf{S}, \mathbf{A}) = a^* = (i, j), \quad (16)$$

where $\mathbf{S} = [S_f, S_c]$ is the current repair state, with S_f representing the repair state executed by the crew fleet and S_c representing the repair state executed by crew c .

3.1.2 Multiple crew time sequences

For a fleet with n_{cre} crews, its two-layer decision-making architecture also has a fleet time sequence and n_{cre} crew time sequences during the MCRP. The time sequence set of crew c can be described as follows:

$$\phi_c = \{\tau_0, \tau_1, \dots, \tau_{t_c}, \dots, \tau_{T_c}\}, \quad (17)$$

where τ_{t_c} represents the time at which this crew finishes its action $a_{t_c|t}^* = (i, j)$ and $\tau_0 = \tau_d$. The parameter τ_{t_c} can be calculated as follows:

$$\tau_{t_c} = \tau_{t_c-1} + \tau(a_{t_c|t}^*), \quad (18)$$

where $\tau(a_{t_c|t}^*)$ represents the time interval to execute action $a_{t_c|t}^*$, which consists of four situations. In the first situation, the crew is at node i , and node j is normal. In the second situation, the crew is at node i , and node j is damaged. In the third situation, the crew is at node i' , and node j is normal. In the fourth situation, the crew is at node i' , and node j is damaged. According to Eqs. (4) and (5), $\tau(a_{t_c|t}^*)$ can be calculated as follows:

$$\tau(a_{t_c|t}^*) = \begin{cases} RT_{ij} \\ RT_{ij} + MTTR_j \\ MT_{i'i} + RT_{ij} \\ MT_{i'i} + RT_{ij} + MTTR_j \end{cases}. \quad (19)$$

The fleet time sequence can be described as follows:

$$\phi = \{\tau_0, \tau_1, \dots, \tau_r, \dots, \tau_T\}, \quad (20)$$

where $\tau_0 = \tau_d$ and where τ_t represents the time at which the crew fleet finishes action $a_{t|t_c}^* = a_{t_c|t}^* = (i, j)$.

3.1.3 Resilience and rewards

For action $a_{t|t_c}^* = a_{t_c|t}^* = (i, j)$, if node j is normal and edge (i, j) is damaged, then, after the repair of edge (i, j) , this edge is assumed to restart its operation. If both node j and edge (i, j) are damaged, then after the repair of edge (i, j) and node j , this edge and this node are assumed to restart their operation. Thus, $FOM(\tau)$ is a step function according to Eq. (6). After finishing an MCRP, the RL can be calculated according to Eq. (7). The reward of this MCRP is defined by normalizing RL as follows:

$$r = \frac{\int_{\tau_d}^{\tau_r} FOM_{thr} d\tau}{\int_{\tau_d}^{\tau_r} [FOM_{thr} - FOM(\tau)] d\tau}. \quad (21)$$

3.1.4 Actor-critic neural network training

For each time step t , the RO agent uses an ACNN $f_\theta(\mathbf{S}, \mathbf{A}) = (\mathbf{p}, \nu)$ to provide the search guide for a variant of the MCTS algorithm $f_s(\mathbf{S}, \mathbf{p}, \nu) = (\pi, a^*)$, where \mathbf{p} is a prior repair probability matrix to guide the tree search to select repair actions, whereas ν is a normalization parameter to evaluate the reward of the current repair action selection. The search algorithm uses the parameters \mathbf{p} and ν to output a search probability π to select a high-quality repair action a^* . Furthermore, the parameters \mathbf{p} and ν are generated by the ACNN, and more details of this ACNN architecture are given in Section 3.2.

The ACNN training data for each time step t are stored as $(\mathbf{S}_t, \mathbf{A}_t, \boldsymbol{\pi}_t, z_t)$, where $z_t = \alpha r + \beta \Delta FOM_t$ is the reward for the action $a_{t|t_c}^*$. α and β are empirical parameters, and $\Delta FOM_t = FOM(\tau_t) - FOM(\tau_{t-1})$. While the RO agent finishes an MCRP, one episode has also been finished, and the data $(\mathbf{S}, \mathbf{A}, \boldsymbol{\pi}, z)$ are sampled from all time steps of this MCRP. The DNN then uses the data $(\mathbf{S}, \mathbf{A}, \boldsymbol{\pi}, z)$ to adjust its parameters θ . The training targets maximize the similarity of the prior repair probability \mathbf{p} to the tree search probability $\boldsymbol{\pi}$ and minimize the error between the prior value ν and the reward z . Thus, the loss function is summed over the cross-entropy losses (CELs) and mean squared errors (MSEs) via gradient descent to adjust the parameters θ as follows:

$$f_{loss} = (z - \nu)^2 - \pi^T \log \mathbf{p} + c_{L2} \|\theta\|^2, \quad (22)$$

where c_{L2} is a constant that is used to control the L2 regularization level to prevent overfitting.

During each episode, a mini-batch of training samples is randomly selected from the data set containing all the episode data. The parameters θ are then adjusted via

batch training. After adjusting the parameters θ , the ACNN enhances the RO agent's decision-making capabilities for the next episode. The initial ACNN parameters θ are randomly generated. As the number of episodes increases, the trained ACNN provides stronger guidance for the tree search. After a sufficient number of episodes, the ACNN is well trained, and the MSEs and CELs stabilize. The RO agent can then effectively support decision-making to solve the RO problem for the IN under localized attacks. Furthermore, the ACNN does not need to be retrained during the IN's operation phase when a localized attack occurs.

3.2 Decision-making of the agent

At each time step t of the MCRP, the RO agent executes an AC-MCTS to select a repair action $a_{t|t_c}^*$, guided by an ACNN.

3.2.1 Actor-critic Monte Carlo tree search algorithm

The AC-MCTS algorithm generates a search tree. In this search tree, each tree node \mathbf{S} represents a repair state and contains tree edges (\mathbf{S}, a) . Each tree edge (\mathbf{S}, a) describes a legal repair action $a = (i, j)$ in the legal repair action space $\xi(\mathbf{S})$. Each crew at the current repair state has its own $\xi(\mathbf{S})$, which contains the repair actions for damaged IN nodes and edges, except for the repair actions that other crews are executing.

Each tree edge (\mathbf{S}, a) stores a set of statistics $\{P(\mathbf{S}, a), N(\mathbf{S}, a), W(\mathbf{S}, a), Q(\mathbf{S}, a)\}$, where $P(\mathbf{S}, a)$ represents the prior repair probability of selecting the tree edge (\mathbf{S}, a) , $N(\mathbf{S}, a)$ represents the visit count of the tree edge (\mathbf{S}, a) , $W(\mathbf{S}, a)$ represents the total repair action value and $Q(\mathbf{S}, a)$ represents the mean repair action value.

The AC-MCTS algorithm has four steps. Multiple simulations proceed via iteration over three steps, as shown in Fig. 3. step (a)–(c). The last step selects a repair action, as shown in Fig. 3. step (d).

(1) Selection

The search tree executes the selection phase beginning at a root tree-node \mathbf{S}^0 and finishing at a leaf tree-node \mathbf{S}^L . The current repair state is taken as the root tree node \mathbf{S}^0 . At each time step $t = l \in [0, L]$ of this selection phase, the search tree selects a repair action as follows:

$$a^l = \operatorname{argmax}_{a^l \in \xi(\mathbf{S}^l)} [Q(\mathbf{S}^l, a^l) + U(\mathbf{S}^l, a^l)], \quad (23)$$

$$U(\mathbf{S}^l, a^l) = c_{\text{puct}} P(\mathbf{S}^l, a^l) \frac{\sqrt{\sum_b N(\mathbf{S}^l, b)}}{1 + N(\mathbf{S}^l, a^l)}, \quad (24)$$

where $U(\mathbf{S}^l, a^l)$ is an intermediate variable using a variant of the PUCT algorithm (Rosin, 2011), c_{puct} is a constant determining the selection level, $\sum_b N(\mathbf{S}^l, b)$ is the cumulative visit count of the tree node \mathbf{S}_l , and $N(\mathbf{S}^l, a^l)$ is the

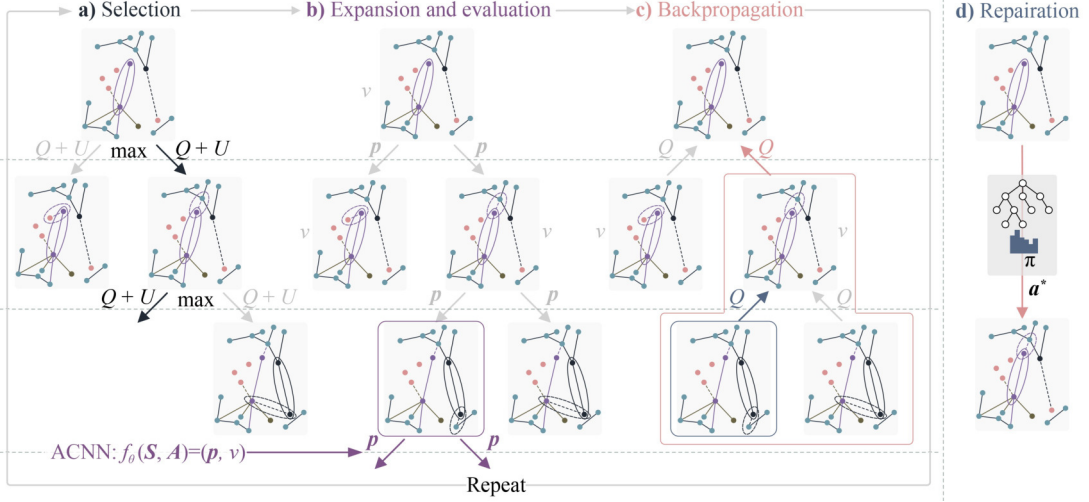


Fig. 3 Actor-critic Monte Carlo tree search algorithm for the RO agent.

cumulative visit count of the tree edge (S^l, a^l) .

(2) Expansion and evaluation

The search tree uses the ACNN $f_{\theta}(\mathbf{S}, \mathbf{A}) = (\mathbf{p}, v)$ to expand and evaluate the leaf tree-node S^l . This ACNN takes the leaf tree-node S^l and the adjacency matrix \mathbf{A} as inputs, and then the search tree expands the leaf tree-node S^l and stores the statistics according to the outputs of the ACNN. The prior repair probability matrix \mathbf{p} of the ACNN output is an $N \times N$ adjacency matrix whose element p_{ij} is the probability of selecting the repair action $a = (i, j)$. Generally, the search tree without the ACNN considers all of the legal tree edges (S^l, a^l) to expand this leaf tree node, whereas the search tree with the ACNN uses the prior repair probabilities to select the tree edges, as shown in Fig. 3. Step b. Each tree edge (S^l, a^l) of the leaf tree node S^l is initialized as follows:

$$\{N(S^l, a^l) = 0, W(S^l, a^l) = 0, Q(S^l, a^l) = 0, P(S^l, a^l) = p_{a^l}\}, \quad (25)$$

where p_{a^l} is the prior repair probability of selecting the tree edge (S^l, a^l) and is a matrix element of the ACNN output \mathbf{p} .

(3) Backpropagation

A backward pass is executed through each tree-node S^l in the above selection phase to update the tree-edge statistics. The statistics of the tree edge (S^l, a^l) are incremented as follows:

$$N(S^l, a^l) = N(S^l, a^l) + 1, \quad (26)$$

$$W(S^l, a^l) = W(S^l, a^l) + v, \quad (27)$$

$$Q(S^l, a^l) = \frac{W(S^l, a^l)}{N(S^l, a^l)}, \quad (28)$$

where v is the predicted value of the ACNN output and is

a normalization parameter used to evaluate the reward of the current repair action selection.

(4) Repair

After multiple simulations proceed by iterating over the above three steps, a higher-quality repair action a^* is selected by the fourth step, using a search probability π , which is an $N \times N$ adjacency matrix whose element is the probability of selecting the repair action $a = (i, j)$. This search probability is proportional to the exponentiated visit count, as follows:

$$\pi(a|S^0) = \frac{N(S^0, a)^{1/\varphi}}{\sum_b N(S^0, b)^{1/\varphi}}, \quad (29)$$

where φ is a temperature parameter used to control the exploration level. The repair action a^* is executed to transform at the root tree node S^0 .

For the next time step $t+1$ of the MCRP, the subsequent tree search is executed, and the search tree is reused: the new root tree node is the child tree node corresponding to the last high-quality repair action $a^*_{t|t}$. The search tree retains the subtree below this child tree node along with statistics and discards the remainder of the search tree. In summary, the tree search can be expressed as $f_{\text{is}}(\mathbf{S}, \mathbf{p}, v) = (\pi, a^*)$.

3.2.2 Actor-critic neural network

The ACNN $f_{\theta}(\mathbf{S}, \mathbf{A}) = (\mathbf{p}, v)$ is a double-input double-output neural network, as shown in Fig. 2. At each time step t , the inputs of ACNN f_{θ} are the current repair state $\mathbf{S}_t = [S^r_t, S^c_t]$ and adjacency matrix \mathbf{A} before damage. This ACNN processes the input features S_t via a residual module that consists of residual blocks (He et al., 2016) of convolutional layers with ReLU functions, whereas the input feature \mathbf{A} is a constant matrix and is processed by a multilayer module that consists of fully connected layers.

To capture the features in the two-layer decision-making architecture, the current repair state S_t consists of a fleet-level repair state S_f^t and a crew-level repair state S_c^t . S_f^t can be written as follows:

$$S_f^t = [A_f^t, A_f^{t-1}, \dots, A_f^{t-5}], \quad (30)$$

where A_f^t is the adjacency matrix at time step t of the MCRP, and if $t < 1$, then $A_f^t = A_f^1 = A_{\text{damage}}$. History adjacency matrices are necessary because the RO agent is not fully observable solely from the current adjacency matrix, since the MCRP is a sequence decision process. Moreover, S_c^t can be written as follows:

$$S_c^t = [A_c^t, A_c^{t-1}, A_c^{t-2}], \quad (31)$$

where A_c^t is the adjacency matrix at time step t_c of the crew c repair process, and if $t_c < 1$, then $A_c^t = A_c^1 = A_d$.

The residual and multilayer modules consist of shared layers and merge into two distinct modules: the actor head and the critic head. Both the actor head and the critic head receive inputs from the shared layers. The inputs from the residual module are processed by residual convolutional layers and a fully connected layer, whereas the inputs from the multilayer module are processed by fully connected layers. Then, the actor head and critic head output prior repair probabilities p_r and values v_r via a fully connected layer and a sigmoid function.

4 Case study

Python is used to simulate the case study for the post-disaster restoration of the IN. The power grid data with 228 nodes and 612 edges are chosen as an example for the IN (Molnar et al., 2021). The power systems are represented as nodes, and the transmission lines between these nodes are represented as edges to construct the graph G of the power grid. The case study then examines the resilience and restoration of the power grid under large-scale damage scenarios, and the optimal RO strategies are generated on the basis of the proposed DRL framework. Furthermore, the proposed framework can be applied to other real-world IN scenarios by analyzing the distributed geographical characteristics of nodes and edges.

4.1 Postdisaster restoration analysis

4.1.1 Damage examples

Figure 4 shows four damage cases of the power grid G , with randomly generated attack centers and damage radii. The damage nodes are highlighted in pink. The destruction information for each damage case is shown in Table 1.

The repair activities are conducted concurrently by

multiple repair crews with varying repair capabilities. For this section, we consider five repair crews with different values of v_{edge} and v_{move} . The nodes are categorized into three levels: first-level, second-level, and third-level nodes, with node weights w belonging to $(0, 2]$, $(2, 4]$, and $(4, +\infty)$, respectively. Each repair crew has different *MTTRs* for the three levels of nodes: *MTTR*^{1st}, *MTTR*^{2nd}, and *MTTR*^{3rd}. The repair capabilities of each repair crew are shown in Table 2. The repair capabilities can also be assigned on the basis of actual conditions. The four MCRPs in Fig. 5 are executed by a crew fleet with Crews 1, 2, and 3.

4.1.2 Repair strategies and resilience analysis

For each damage case depicted in Fig. 5, it is assumed that a localized attack event occurs at time $\tau_d = 200$ h, resulting in a significant degradation in the functional performance of the power grid. The power grid subsequently undergoes repair actions to recover from the localized attack, leading to an improvement in functional performance. The functional performance reaches the threshold $FOM_{\text{thr}} = 0.96$ at time τ_r . Generally, different MCRPs have different values of τ_r , and a resilience analysis of the four MCRPs is shown in Fig. 5. Notably, τ_d is an assumed time and can be set to any random number. The FOM threshold is an empirical parameter and can be set on the basis of the actual situation. To facilitate a clear comparison of different values of τ_r , a deterministic time τ_d is chosen for the case study. The *RLs* of the four MCRPs are shown in Fig. 5. Table 3 presents the *RL* and recovery time τ_r for each damage case.

4.1.3 Empirical analysis of agent training

The proposed method is applied to train the RO agent via a single machine equipped with one Intel i9 7980XE CPU and four RTX2080 TI-A11G GPUs. The training process begins with a completely random ACNN and continues for approximately 28 h without human intervention. During the training process, 10,000 MCRPs are executed, with 800 simulations for each MCTS, resulting in nearly 0.2 s of thinking time per repair action. The ACNN parameters are updated via mini-batches from the data set generated by the MCRPs.

Figure 5 shows the MSEs and CELs of the ACNN after each episode of the MCRP. The MSE measures the discrepancy between the actual outcome z and the ACNN value v , which predicts the actual optimization of the *RL*. The CEL indicates the similarity between the ACNN probabilities p and the MCTS policy π , which predicts the policy π . The hyperparameters used are as follows: $a = 2$, $b = 4$, $c_{L2} = 10^{-4}$, and $c_{\text{puct}} = 5$. Although the mathematical models of MSE and CEL differ, the ACNN is trained on the same training data set and loss function

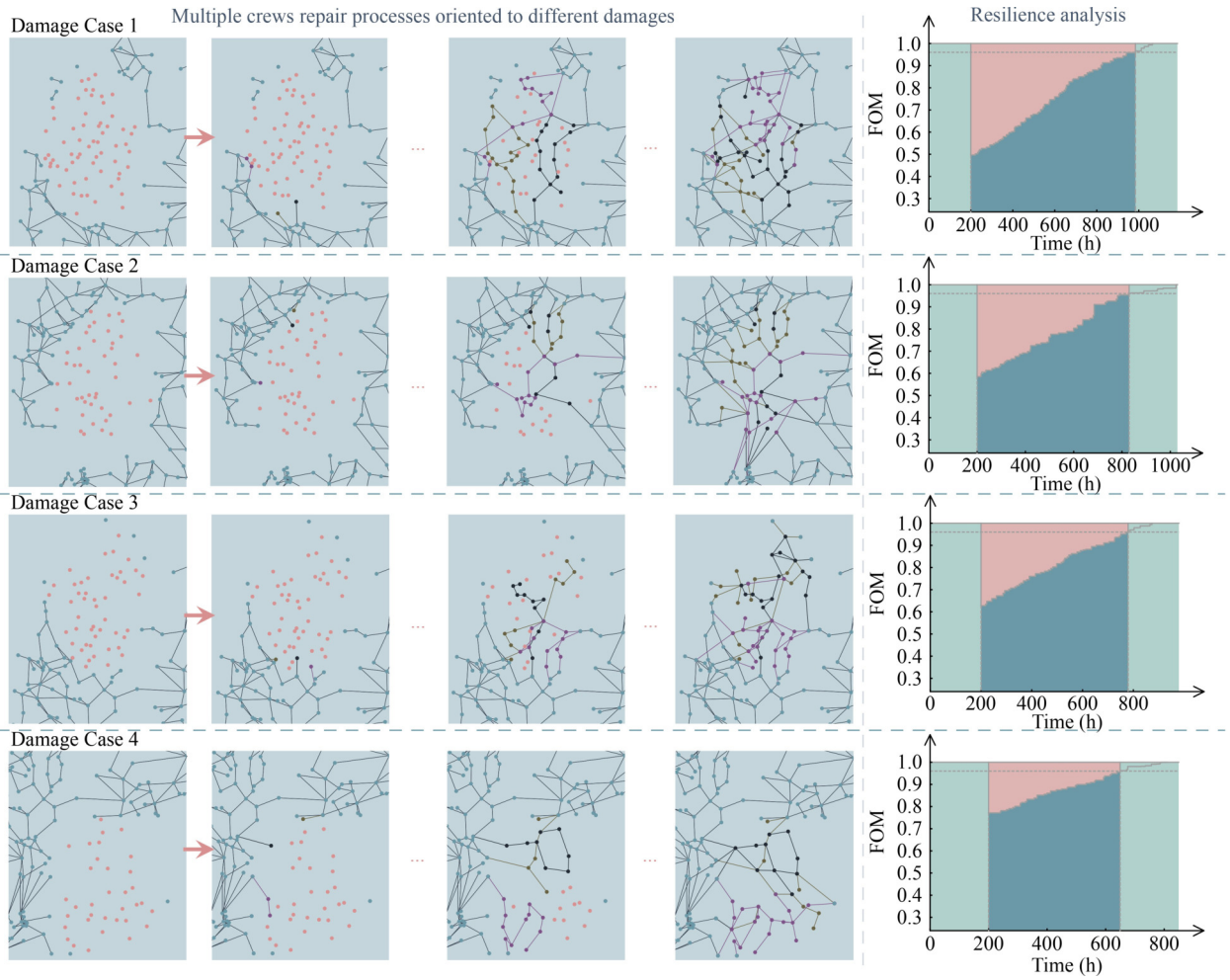


Fig. 4 Different repair processes with multiple crews.

Table 1 Information of each damage case

Damage case	Damage center	Damage radius (km)	Damage number		Current FOM
			Nodes	Edges	
Case 1	8.64810°W, 51.82235°N	188.59	62	186	0.496677
Case 2	9.19534°W, 50.28620°N	187.06	43	140	0.585140
Case 3	8.92320°W, 53.04213°N	182.39	47	140	0.627756
Case 4	11.22784°W, 48.82155°N	185.84	34	102	0.771439

Table 2 Information of each repair crew

Parameter	Crew 1	Crew 2	Crew 3	Crew 4	Crew 5
$MTTR^{1st}$ (h)	1	1.5	2	1.2	1.8
$MTTR^{2nd}$ (h)	3.2	4.1	4.8	3.7	4.5
$MTTR^{3rd}$ (h)	6.1	7.2	7.9	6.6	7.6
v_{edge} (km/h)	1.8	2	2.1	1.9	2
v_{move} (km/h)	50	55	60	52	58

(Eq. (17)). The trends of variation in the two curves are similar in some episodes of training. After approximately

8000 MCRP episodes, the MSE and CEL tend to stabilize, enabling the ACNN to efficiently guide the MCTS in selecting repair actions.

4.2 Discussion

In the context of IN restoration, the primary objective is to obtain an optimal feasible solution. Extensive analyses of the optimization solution are conducted to assess the influences of different factors.

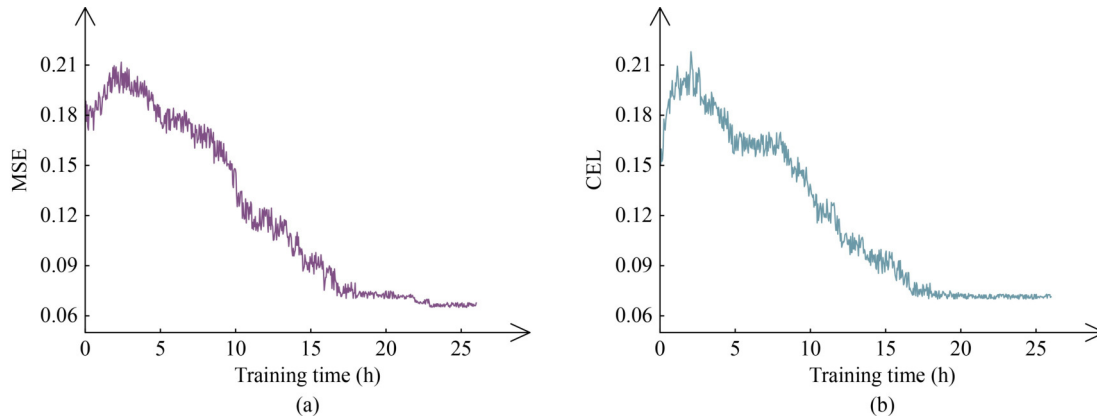


Fig. 5 MSEs (a) and CELs (b) of the ACNN.

Table 3 Resilience loss and recovery time for each damage case

Damage case	Case 1	Case 2	Case 3	Case 4
Resilience loss RL	202.4516	145.8626	114.4251	62.3091
Recovery time τ_r (h)	984.123	829.632	779.455	649.462

4.2.1 Different algorithms

An AC-MCTS method with high efficiency is proposed for IN postdisaster restoration in this paper. Additionally, a PPO method, a DDPG method, a double-DQN method, an ASC method, and a heuristic hybrid game (HHG) method (Feng et al., 2016) have been employed to address the same problem. Each of these six methods is analyzed separately by executing a Python program for each method to solve the four damage cases presented in Table 2. The AC-MCTS method outperforms the other methods in terms of RL , as illustrated in Fig. 6. In the initial MCRP stage for each damage case, the PPO method, the DDPG method, and the double-DQN method may achieve a higher FOM than the AC-MCTS method; however, these three methods may also lead to local optima, such as situations where multiple crews need to spend more time traveling during the middle and late MCRP stages. For certain cases, such as those depicted in Fig. 6(a) and 6(c), the PPO method, DDPG method, and double-DQN method achieve better RL than the HHG method and the ASC method do, but the HHG method and the ASC method are able to complete their MCRP sooner than these three methods are. According to the findings presented in Fig. 6, the AC-MCTS method results in superior RL results in less time than the other methods do because it uses the repair action value predicted by the ACNN and estimates the long-delayed reward of future repair actions in a search tree, thereby striving to achieve better RL for the entire MCRP. Therefore, the results demonstrate that the AC-MCTS method produces optimized strategies for postdisaster restoration.

The efficiencies of the methods are analyzed in Table 4. According to these results, the solution speeds of

Table 4 Running times (in Seconds) of different methods

Method	Case 1	Case 2	Case 3	Case 4
AC-MCTS	97.672	89.243	90.187	84.587
PPO	95.647	87.952	90.012	83.245
DDPG	94.313	87.115	89.584	81.313
Double-DQN	94.845	87.558	89.876	82.545
ASC	312.647	237.564	232.842	208.549
HHG	325.953	246.476	249.295	212.377

the AC-MCTS, PPO, DDPG, and double-DQN methods remain stable and do not exponentially decrease as the numbers of damaged nodes and edges increase. However, the solution speed of the ASC and HHG methods significantly decreases as the damage level increases.

4.2.2 Different numbers of crews

The same damage case (Damage Case 1 in Table 2) is maintained, and variations in crew fleet sizes of three, four, and five are examined via six different methods. The sensitivity results for a three-crew fleet (Crews 1, 2, and 3) are displayed in Fig. 6(a), whereas the results for a four-crew fleet (Crews 1, 2, 3, and 4) and a five-crew fleet (Crews 1, 2, 3, 4, and 5) are shown in Fig. 7(a) and 7(b), respectively. RL and recovery time for different crew numbers are documented in Table 5. As the number of crewmembers increases, the RL and total repair time decrease. The RL achieved by the RO agent is proven to be superior to that of other methods in terms of time efficiency. Thus, these findings demonstrate that the RO agent performs consistently in postdisaster restoration problems with varying crew sizes. Notably, the number of crews is generated randomly during the RO agent training process.

4.2.3 Different abilities of crews

In addition, focusing on the same damage case (Damage

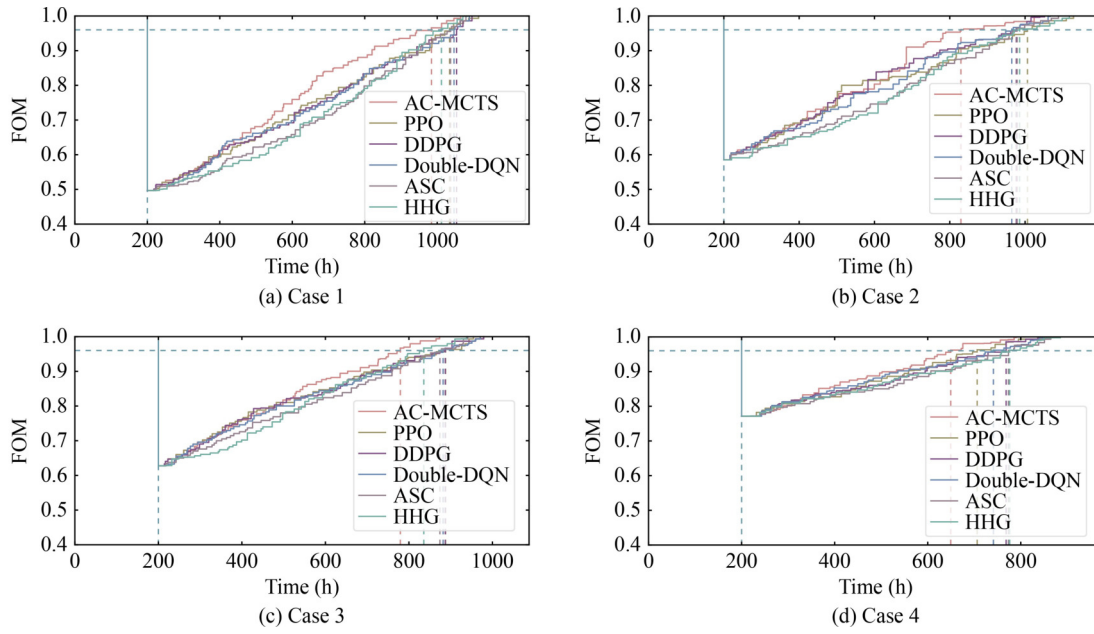


Fig. 6 Resilience loss of different methods for each damage case.

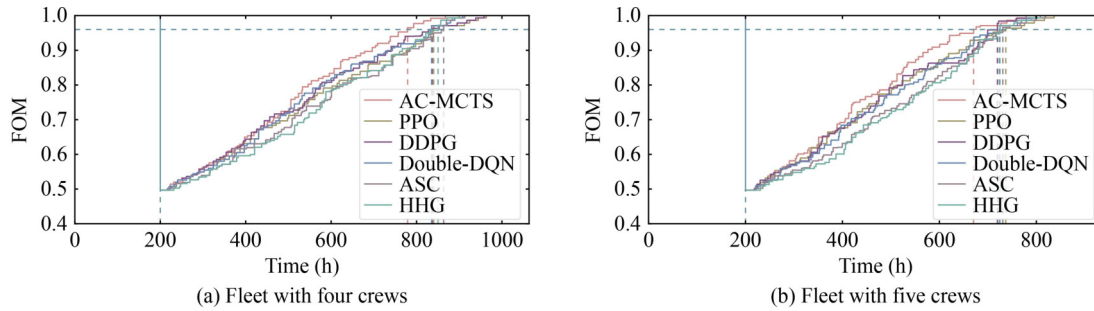


Fig. 7 Comparison of resilience with different numbers of crews

Table 5 Resilience loss and recovery time for different numbers of Crews

Crew fleet	Crews 1–3		Crews 1–4		Crews 1–5	
	RL	τ_r (h)	RL	τ_r (h)	RL	τ_r (h)
AC-MCTS	202.4516	984.123	157.4468	779.136	133.5588	670.447
PPO	232.2561	1034.058	178.4199	840.131	139.7974	736.981
DDPG	235.7050	1053.520	168.7603	836.657	140.8523	719.631
Double-DQN	236.7415	1046.361	171.4111	835.263	143.9193	723.802
ASC	258.3306	1036.819	189.6965	863.517	158.3478	730.855
HHG	256.4923	1011.631	190.8799	850.423	162.4286	724.925

Case 1 in Table 2) and employing the RO agent, we analyze performance differences when crew abilities vary. Resilience analyses for a fleet including Crews 1, 2, and 3 are shown in Fig. 4 (MCRP for damage case 1 and its resilience analysis), whereas analyses for fleets comprising Crews 2, 3, and 4 and Crews 3, 4, and 5 are illustrated in Fig. 8(a) and 8(b), respectively. RL and recovery time for varying crew abilities are summarized in Table 6. With the same RO agent, the optimal strategies achieve similar RL s with nearly identical total repair

times. These results demonstrate the consistent performance of the RO agents across crews with varying abilities.

5 Conclusions

The postdisaster restoration of INs has been a vital area of research in the recent past. In this paper, a well-formulated research problem is presented, taking into

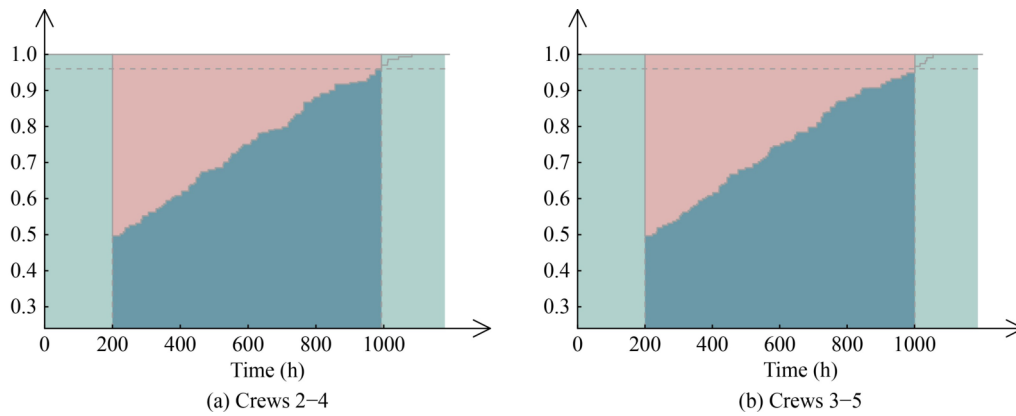


Fig. 8 Different abilities of crews: (a) Crews 2–4 (b) Crews 3–5.

Table 6 Resilience loss and recovery time for different abilities of Crews

Crew fleet	Crews 1–3	Crews 2–4	Crews 3–5
Resilience loss RL	202.4516	206.9847	208.4678
Recovery time τ_r (h)	984.123	993.253	1001.279

consideration the optimum sequence decision and optimum route selection issues for increasing the resilience of INs by engaging multiple crews. This paper presents a two-layer decision-making DRL framework of RO strategies with multiple crews, along with a solution methodology, namely, AC-MCTS, to effectively solve the problem. A case study is conducted in Python using the 228-node power grid as an exemplary IN to show that the proposed AC-MCTS method can generate a globally optimal RO strategy to minimize the overall RL for multiple crews. The AC-MCTS algorithm determines not only the optimum routes for repair but also the optimum decisions on the sequence of repairs. The results of the case study prove that, compared with other approaches, the AC-MCTS method yields better RL while reducing the time taken for repairs. Moreover, the proposed AC-MCTS method can support effective and consistent restoration decision-making in larger-scale INs. As the scale of the INs further increases, the training cost of the ACNN increases. Nevertheless, the proposed method can exploit distributed computing and parallel processing via languages such as C++, which can reduce training efficiency via several factors.

In the future, further research on optimal planning strategies for initial repair crews and resources is needed. Other important approaches to solve distributed policy optimization of the planning problem of multiple crews for IN restoration, such as multiagent DRL, are also worth investigating.

Competing Interests The authors declare that they have no competing interests.

References

- Ahmadian N, Lim G J, Cho J, Bora S (2020). A quantitative approach for assessment and improvement of network resilience. *Reliability Engineering & System Safety*, 200: 106977
- Allal A, Sahnoun M, Adjoudj R, Benslimane S M, Mazar M (2021). Multi-agent based simulation-optimization of maintenance routing in offshore wind farms. *Computers & Industrial Engineering*, 157: 107342
- Arulkumar K, Deisenroth M P, Brundage M, Bharath A A (2017). Deep reinforcement learning a brief survey. *IEEE Signal Processing Magazine*, 34(6): 26–38
- Bai G H, Wang H, Zheng X Q, Dui H Y, Xie M (2021). Improved resilience measure for component recovery priority in power grids. *Frontiers of Engineering Management*, 8(4): 545–556
- Browne C B, Powley E, Whitehouse D, Lucas S M, Cowling P I, Rohlfshagen P, Tavener S, Perez D, Samothrakis S, Colton S (2012). A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1): 1–43
- Carnevali L, Tarani F, Vicario E (2020). Performability evaluation of water distribution systems during maintenance procedures. *IEEE Transactions on Systems, Man, and Cybernetics. Systems*, 50(5): 1704–1720
- Das L, Munikoti S, Natarajan B, Srinivasan B (2020). Measuring smart grid resilience: Methods, challenges and opportunities. *Renewable & Sustainable Energy Reviews*, 130: 109918
- Diaz-Ramírez J, Huertas J I, Trigos F (2014). Aircraft maintenance, routing, and crew scheduling planning for airlines with a single fleet and a single maintenance and crew base. *Computers & Industrial Engineering*, 75: 68–78
- Du Y, Wu D (2022). Deep reinforcement learning from demonstrations to assist service restoration in islanded microgrids. *IEEE Transactions on Sustainable Energy*, 13(2): 1062–1072
- Dui H Y, Dong X H, Wu X M, Chen L W, Bai G H (2024a). IoT-enabled risk warning and maintenance strategy optimization for tunnel-induced ground settlement. *IEEE Internet of Things Journal*, 11(13): 22966–22981
- Dui H Y, Liu M, Song J Y, Wu S M (2023). Importance measure-based resilience management: Review, methodology and perspectives on maintenance. *Reliability Engineering & System Safety*,

- 237: 109383
- Dui H Y, Wu X M, Wu S M, Xie M (2024b). Importance measure-based maintenance strategy optimization: Fundamentals, applications and future directions in AI and IoT. *Frontiers of Engineering Management*, 11(3): 542–567
- Dui H Y, Zhang H Q, Dong X H, Zhang S R (2024c). Cascading failure and resilience optimization of unmanned vehicle distribution networks in IoT. *Reliability Engineering & System Safety*, 246: 110071
- Dui H Y, Zhu Y W, Tao J Y (2024d). Multi-phased resilience methodology of urban sewage treatment network based on the phase and node recovery importance in IoT. *Reliability Engineering & System Safety*, 247: 110130
- Fan X D, Zhang X J, Yu X (2022). A graph convolution network-deep reinforcement learning model for resilient water distribution network repair decisions. *Computer-Aided Civil and Infrastructure Engineering*, 37(12): 1547–1565
- Feng Q, Bi X, Zhao X J, Chen Y R, Sun B (2017). Heuristic hybrid game approach for fleet condition-based maintenance planning. *Reliability Engineering & System Safety*, 157: 166–176
- Fu J F, Nunez A, De Schutter B (2022). Real-time UAV routing strategy for monitoring and inspection for postdisaster restoration of distribution networks. *IEEE Transactions on Industrial Informatics*, 18(4): 2582–2592
- Ganganath N, Wang J V, Xu X, Cheng C, Tse C K (2018). Agglomerative clustering-based network partitioning for parallel power system restoration. *IEEE Transactions on Industrial Informatics*, 14(8): 3325–3333
- He K M, Zhang X Y, Ren S Q, Sun J (2016). Deep residual learning for image recognition. In: *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Conference on Computer Vision and Pattern Recognition. Seattle, CA: IEEE, 770–778
- Liu Z X, Wang L F (2021). Leveraging network topology optimization to strengthen power grid resilience against cyber-physical attacks. *IEEE Transactions on Smart Grid*, 12(2): 1552–1564
- Lu L Q, Wang X, Ouyang Y F, Roningen J, Myers N, Calfas G (2018). Vulnerability of interdependent urban infrastructure networks: equilibrium after failure propagation and cascading impacts. *Computer-Aided Civil and Infrastructure Engineering*, 33(4): 300–315
- Molnar F, Nishikawa T, Motter A E (2021). Asymmetry underlies stability in power grids. *Nature Communications*, 12(1): 1457
- Ogunfowora O, Najjaran H (2023). Reinforcement and deep reinforcement learning-based solutions for machine maintenance planning, scheduling policies, and optimization. *Journal of Manufacturing Systems*, 70: 244–263
- Pincirolu L, Baraldi P, Ballabio G, Compare M, Zio E (2021). Deep reinforcement learning based on proximal policy optimization for the maintenance of a wind farm with multiple crews. *Energies*, 14(20): 6743
- Rosin C D (2011). Multi-armed bandits with episode context. *Annals of Mathematics and Artificial Intelligence*, 61(3): 203–230
- Ruan J H, Wang Z X, Chan F T S, Patnaik S, Tiwari M K (2021). A reinforcement learning-based algorithm for the aircraft maintenance routing problem. *Expert Systems with Applications*, 169: 114399
- Si S B, Zhao J B, Cai Z Q, Dui H Y (2020). Recent advances in system reliability optimization driven by importance measures. *Frontiers of Engineering Management*, 7(3): 335–358
- Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A, Chen Y T, Lillicrap T, Hui F, Sifre L, van den Driessche G, Graepel T, Hassabis D (2017). Mastering the game of Go without human knowledge. *Nature*, 550(7676): 354–359
- Wang P J, Wen G H, Yu X H, Yu W W, Wan Y (2021). Synchronization of resilient complex networks under attacks. *IEEE Transactions on Systems, Man, and Cybernetics. Systems*, 51(2): 1116–1127
- Wu Q L, Feng Q, Ren Y, Xia Q, Wang Z L, Cai B P (2021). An intelligent preventive maintenance method based on reinforcement learning for battery energy storage systems. *IEEE Transactions on Industrial Informatics*, 17(12): 8254–8264
- Wu Z R, Xu W W, Li C C, Meng X F (2022). A new approach for generator startup sequence online decision making with a heuristic search algorithm and graph theory. *Energy Reports*, 8: 678–686
- Xie H N, Jiang M H, Zhang D D, Goh H H, Ahmad T, Liu H, Liu T H, Wang S Y, Wu T (2023). IntelliSense technology in the new power systems. *Renewable & Sustainable Energy Reviews*, 177: 113229
- Xu M, Ouyang M, Hong L, Mao Z J, Xu X L (2022). Resilience-driven repair sequencing decision under uncertainty for critical infrastructure systems. *Reliability Engineering & System Safety*, 221: 108378
- Xu Z Z, Chopra S S (2022). Network-based assessment of metro infrastructure with a spatial-temporal resilience cycle framework. *Reliability Engineering & System Safety*, 223: 108434
- Yu S, Zhang L, Lu H H, Fernando T, Wong K P (2017). A DSE-based power system frequency restoration strategy for PV-integrated power systems considering solar irradiance variations. *IEEE Transactions on Industrial Informatics*, 13(5): 2511–2518
- Zhang Z Y, Ji T T, Wei H H (2022). Dynamic emergency inspection routing and restoration scheduling to enhance the post-earthquake resilience of a highway-bridge network. *Reliability Engineering & System Safety*, 220: 108282
- Zuo M J (2021). System reliability and system resilience. *Frontiers of Engineering Management*, 8(4): 615–619