An Online Search Method for Representative Risky Fault Chains Based on Reinforcement Learning and Knowledge Transfer

Zhimei Zhang, Student Member, IEEE, Shaowei Huang, Member, IEEE, Ying Chen, Member, IEEE, Shengwei Mei, Fellow, IEEE, and Kai Sun, Senior Member, IEEE

Abstract—In the analysis of cascading outages and blackouts in power systems, risky cascading fault chains should be accurately identified in order to do further block or alleviate blackouts. However, the huge computational burden makes online analysis difficult. In this paper, an online search method for representative risky fault chains based on reinforcement learning and knowledge transfer is proposed. This method aims at promoting efficiency by exploiting similarities of adjacent power flow snapshots in operations. After the “representative risky fault chain” is defined, a framework of tree search based on Markov Decision Process and Q-learning is constructed. The knowledge in past runs is accumulated offline and then applied online, with a mechanism of knowledge transition and extension. The proposed learning based approach is verified on an illustrative 39-bus system with different loading levels, and simulations are carried out on a real-world 1000-bus power grid in China to show the effectiveness and efficiency of the proposed approach.

Index Terms—Blackout, cascading outages, fault chain, reinforcement learning, Markov Decision Process, Q-learning, knowledge transfer, tree search.

I. INTRODUCTION

More secure and reliable power system has been pursued by both academic and industrial entities perpetually. However, large-scale blackouts triggered by cascading outages [1]-[4] have been verified as major threats to the society. Therefore, it is important to study the cascading outages and then develop effective strategies that assist in blackout mitigation.

A cascading outage refers to a process during which sequential outages of individual components arise, weakening the power system or even collapsing it. Great efforts have been made on the analysis and mitigation of cascading outages. However, due to the low probability, rapid propagation and complex mechanism of cascading failures, the defense against blackout is still a quite challenging task.

A fault chain (FC) is a sequence of tripped components during the process of cascading outages [5], which captures the dependency in the outage propagation process. Blackouts usually do not happen abruptly [6]. In contrast, the failure propagates gradually, which eventually causes severe load loss. For example, the process of the 2003 Northeast America blackout lasted more than 2 hours [1]. So the process of cascading outages can be formulated as FCs. The FCs can be utilized for risk assessment [7]-[9], critical component identification [5], [10], [11], fault blocking [12] and other issues. Therefore, it is desirable to identify the FCs of interest in an accurate and efficient manner.

According to the state-of-the-art literature, related models can be classified into simplified ones (high-level statistical or topology-based) and detailed ones (power system modeled-based). The former group, including CASCADE [13] and branching process [14], aims at fast computation. These models extract some fundamental features of cascading failure, but they are less successful in representing the characteristics of the physical models. The latter group, represented by OPA [15] model and its derivatives [16], retains power flow models or even more detailed modeling of the components and processes. However, it is more computationally intensive and its online application is limited.

Various techniques have been proposed to enhance the efficiency for risky FC identification mainly based on detailed models. In [5], an index predicting power flow redistribution is exploited to generate risky FCs. Ref. [17] reports a randomly broad search method with later pruning to yield high-risk $N-k$ contingencies. Ref. [18] claims that historical data should be emphasized. Unnecessary transmission security constraints are eliminated to accelerate computation in [19]. Besides, an optimization-based bi-level method [20] is proposed to identify a series of worst $N-k$ contingencies. Both authors in [8] and [9] highlight the impact of different time scales. In [9], a remarkable synthesized index called risk estimation index (REI) is handled to search for risky chains with priority. While a sequential importance sampling (SIS) based strategy [21] is derived to acquire rare but critical FCs. Ref. [22] proposes an algorithm based on artificial intelligence (AI) that probes to FCs with maximal damages.

Even if the above acceleration methods are adopted, the identification of risky FCs is still very time-consuming and is hard to be applied online. When faced with constantly varying power flow snapshots, identifying FCs from scratch on each snapshot cannot meet the requirements of the online application. Fortunately, the state of power systems usually...
varies consecutively and mildly, so the FCs of two adjacent power flow snapshots may be different but similar. Then the problem arises: how to exploit this similarity so as to search for risky FCs efficiently in the online application scenario?

In this paper, a method of representative risky FC search based on reinforcement learning (RL) and knowledge transfer is proposed. The method first establishes the knowledge of each component’s vulnerability of the initial power flow snapshot from a cascading failure simulator or real data. After initialization, the knowledge is transferred to the power flow snapshot in the next dispatch cycle so that the search preference is swiftly updated. That is, a branch outage is more likely to be searched if the corresponding component proves to be riskier in the previous power flow snapshot. Meanwhile, the knowledge is also extended by additional search into uncharted branches. By utilizing known information from previous analysis instead of starting tabula rasa (from scratch) every time, the proposed method avoids low-risk FCs, which significantly saves time and thus has the potential to be implemented online.

The major contributions of this paper are as follows:

1) A reinforcement-learning-based framework of representative risky FC search is constructed. In the framework, the search process is modeled as a Markov Decision Process (MDP). Then a general search procedure based on the MDP is presented, where every FC is associated with a series of actions, and risky ones have positive values of actions.

2) A search strategy based on reinforcement learning within a power flow snapshot is introduced, including the exploitation and exploration of knowledge during the search. It not only helps find more representative risky FCs but also prepares knowledge for future similar snapshots.

3) The mechanism of knowledge transfer across power flow snapshots for representative risky FC identification is established. Then strategies of transition and extension for accelerating computation based on learned previous knowledge is proposed, which significantly enhances the performance of searching under the background of varying power flow.

The rest of this paper is organized as follows. Section II clarifies the target of the proposed search scheme. Section III constructs the framework of representative risky FC search based on reinforcement learning. Section IV presents the search strategies of offline initialization and online knowledge transition-extension. Case studies on two power systems are carried out in Section V. Finally, concluding remarks are drawn in Section VI.
can manufacture a mass of risky FCs far more productively after one risky FC is found, than explore other different FCs. But these similar FCs share the same core, and thus most of them are pointless.

To this end, a practical framework should be proposed to identify representative risky FCs. In this paper, the object of the search scheme is to maximize the number of obtained representative risky FCs in a given number $N_{\text{max}}$ of search trials.

III. FRAMEWORK OF REPRESENTATIVE RISKY FAULT CHAIN SEARCH

A. MDP Model of FC Search

According to [9], [21], the power system has the Markovian property, which means that given a state, information including power flow, generation re-dispatch and load shedding can be inferred deterministically by its previous state and the state transition. In order to find out representative risky FCs actively, this paper introduces a virtual agent (attacker) to simulate all the factors causing component outages. The agent aims to cause intense load loss through a series of actions (removing components), which can be regarded as a dynamic programming problem. Therefore, how the agent will take actions can be modeled as an MDP.

The MDP consists of a 4-tuple $(S, A, T, R)$, representing the state, action, probability of state transition, and reward, respectively.

1) State: the state of a system comprises its components' states. For a component, such as a transmission line, a transformer or a generator, its operating status is modeled as a binary variable which denotes normal operation or tripped. Then, the state of the whole system can be denoted by a sequence of tripped components. As different FCs may share the same initial outages, all FCs can be organized into a tree layout, shown in Fig. 2.

2) Action: at each step, the agent selects one available component and removes it, pushing the outages to develop. Although modern power systems conform to $N-1$ security, in some severe scenarios, it will be broken, meaning that the agent is able to act multiple times. A component cannot be removed twice, so it is labeled unavailable after removed. If the system steps into the fast process, the agent will let cascading outages propagate spontaneously. Then the system arrives at an absorbing state, which has no available action. Therefore, the sequence of removals comes to an end.

Here, the so-called “removals” may be actual physical/cyber attacks conducted by terrorists or just outages caused by hidden failure, catastrophic weather or other non-human factors. As Ref. [9] figured out, the time interval of two successive outages during the slow process is usually long, typically 3-20 minutes. So it is reasonable to assume that the agent removes components sequentially. That is, an action of removal is associated with one component, which can reduce the number of actions considerably.

3) State transition: after an action $a$, a new state is reached from the previous one $s$. During the slow process, the removal attempt has a probability $\gamma$ of success, so the distribution of the new state is

$$
\begin{align*}
\Pr(s'|s,a) &= \gamma \\
\Pr(s_{\text{fail}}|s,a) &= 1 - \gamma
\end{align*}
$$

where $s'$ is the state which has exactly one more component (the removed component) than $s$, and $s_{\text{fail}}$ is a variant of the absorbing state. If the system comes into $s_{\text{fail}}$, no new outage will occur and the cascading outage stops propagation.

However, after the fast process of cascading outages, the next state is an absorbing state. This transition is stochastic and hard to predict, due to the complicated mechanism of cascading failures.

4) Reward: the immediate reward $R(s'|s,a)$ is defined as the instant loss of the transition from the current state $s$ to a new state $s'$ by action $a$. In some cases, part of the load must be shed to maintain the equilibrium of the power system after removal of an element. According to (3), the difference of the load between these two states is called “instant loss”.

$$
\begin{align*}
R(s'|s,a) &= \text{Load}(s) - \text{Load}(s') \\
R(s_{\text{fail}}|s,a) &= 0
\end{align*}
$$

If the fast process of cascading tripping is triggered, state transition is random and therefore $s'$ is not deterministic, so the reward of this very last removal is calculated according to its expectation

$$
\begin{align*}
R(s_{n-1},a_n) &= \sum_{s_{ab}} R(s_{ab}|s_{n-1},a_n) \Pr(s_{ab}|s_{n-1},a_n) \\
&= \sum_{s_{ab}} (\text{Load}(s_{ab}) - \text{Load}(s_{ab})) \Pr(s_{ab}|s_{n-1},a_n)
\end{align*}
$$

where $n$ is the length of slow-process FC, $s_{n-1}$ is the last slow state before the fast process, $a_n$ is the very removal that triggers fast process, $s_{ab}$ is a possible absorbing state derived from $s_{n-1}$.

Suppose the agent will act $n$ times from a certain state $s_0$, then its aggregate reward under a certain strategy $\pi$ is

$$
\begin{align*}
V^\pi(s_0) &= \Pr(s_{\pi(s_0)}|s_0,\pi(s_0)) \left( R(s_{\pi(s_0)}|s_0,\pi(s_0)) + V^\pi(s_{\pi(s_0)}) \right) \\
&\quad + \Pr(s_{\text{fail}}|s_0,\pi(s_0)) R(s_{\text{fail}}|s_0,\pi(s_0))
\end{align*}
$$

Fig. 2. Tree layout of FCs
where \( \pi(s) \) is the action taken by the agent at the state \( s \) according to the strategy \( \pi \), \( s_{\pi(s)} \) is the next state after action \( \pi(s) \) is successfully taken, and \( s_t^n \) is the \( t \)-th state starting from \( s_0 \) of \( \pi \). According to (2) and (3), \( V^n(s_0) \) can be expanded recursively as
\[
V^n(s_0) = \sum_{t=0}^{n} \gamma^t R(s_{t+1}^n|s^n_t, \pi(s^n_t))
\]
(6)

Since the \( n \) actions constitute an FC, the aggregate reward \( V^n(s_0) \) is associated with the loss of the FC.

### B. RL-Based Solution

The proposed MDP is solved after every \( R \) and \( V \) is known. If so, all risky FCs (and including representative risky FCs) can be acquired by all the terminal \( R(s_{n-1}, a_n) \). Theoretically, the solution can be achieved by classical dynamic programming (DP) like Bellman iteration, which relies on enumerating. But it is difficult to solve directly because the number of FCs/states is numerous.

RL is a branch of machine learning, which concerns how an agent takes optimal actions in an environment to maximize its reward by self-study [23]. Compared with classical dynamic programming, RL is able to approach the optimal strategy without enumerating all possible branches. Thus it is advisable for relatively complex and time-varying systems [24], for example, the representative risky FC search.

In this paper, a technique based on the classical Q-learning is adopted to approach \( V \) iteratively and incrementally. This type of Q-learning has a clear structure, thus is suitable for solving the RL problem with MDP. The formation of the Q-learning is a value \( Q \) iteration update, using the weighted average of the old value and the new information as follows
\[
Q^{(k+1)}(s_t, a_t) = (1 - \alpha)Q^{(k)}(s_t, a_t) + \alpha R(s_{t+1}|s_t, a_t) + \gamma \max_{a_{t+1}} Q^{(k)}(s_{t+1}, a_{t+1})
\]
(7)

where the weight \( \alpha \in [0, 1] \) is the learning rate, determining to what extent newly acquired knowledge overrides old knowledge.

An episode (corresponding to a search trial) of the Q-learning algorithm ends up with an absorbing state. Then the Qs of related state-action pairs are updated by a backward traverse on the tree to speed up the iterative process, which is different from the traditional Q-learning method. This is because major loss occurs at the fast process (leaf nodes of the tree). For any absorbing state \( s_{ab} \), there is no available action, so \( Q(s_{ab}, a) \) is always zero. Therefore, if \( s_{t+1} \) is an absorbing state, \( Q \) is updated by (8), which is derived from (7):
\[
Q^{(k+1)}(s_t, a_t) = (1 - \alpha)Q^{(k)}(s_t, a_t) + \alpha R(s_{t+1}|s_t, a_t)
\]
(8)

\[
\alpha = \frac{1}{N(s_t, a_t)}
\]
(9)

where \( N(s_t, a_t) \) is the visit count, so the effect of (8) is calculating the average loss of \( N(s_t, a_t) \) distinct trials.

If \( s_{t+1} \) is not an absorbing state, the learning rate \( \alpha \) is 1. Then the \( Q(s_t, a_t) \) is the highest expected loss caused by \( a_t \) and future actions, as shown in Fig. 3. Since the probability and reward can be obtained according to (2) and (3), the value \( Q(s_t, a_t) \) can be decided by its descendent states. Thus, \( Q \) is updated by (10), which is also derived from (7):
\[
Q^{(k+1)}(s_t, a_t) = R(s_{t+1}|s_t, a_t) + \gamma \max_{a_{t+1}} Q^{(k+1)}(s_{t+1}, a_{t+1})
\]
(10)

### C. Overall Procedure of Representative Risky FC Search

In order to avoid getting too many non-representative risky FCs, discovered risky FCs are stored. At a certain step of a search, suppose the agent has taken some actions sequentially. Before taking the current action, actions having been taken \( (a_1, a_2, \ldots) \) combined with the selected action \( a_{ik} \) corresponds to an FC candidate, which is compared to the stored FCs. If a stored FC can represent the FC candidate, this action is also labeled unavailable for future searches, and the agent should select another action. For example, in Fig. 3, \( L_3 \) is unavailable after \( (L_1, L_2) \) tripped because FC \( (L_1, L_2) \) has been identified as risky. Moreover, an action is labeled unavailable when it makes the system move to a state without any available action.

When taking actions, the agent should make tradeoff between exploitation and exploration to avoid falling into local optimum. So the agent performs exploration with the probability \( \epsilon_1 \), which is focused on searching unexplored or less explored paths. Otherwise, the agent exploits the currently “well-behaved” options with larger Qs. The probability \( \epsilon_1 \) declines gradually as the search proceeds, so the agent is more likely to explore at the beginning and then tends to exploit its knowledge.

The overall procedure of RL-based representative risky FC search is shown in Algorithm 1. In the algorithm, the agent should always choose an available action according to prior or learned knowledge. This is crucial to the effectiveness of the Q-learning and will be presented in the next Section.
Algorithm 1 Representative risky FC search based on RL

Set initial value $Q = 0$, visit count $N = 0$ for all state-action pairs

for each episode (from 1 to $N_{\text{max}}$) do
  Reset power flow according to the initial state $s_0$
  for each step of the episode (before fast process) do
    Select an available action $a_i$ according to...
    if $\text{rand}(0,1) \leq \epsilon_1$ then
      exploration: ... to prior knowledge
    else
      exploitation: ... to current $Q$ (learned knowledge)
    end if
    Trip the selected branch
    $N(s, a_i) \leftarrow N(s, a_i) + 1$
    Update power flow
    if fast process or the length of the FC $\geq M$ then
      No more actions, evolve until absorbing state
    else
      Set $s \leftarrow s'$
    end if
    Calculate $R(s, a_i)$
  end for
  if $R > TH\%$ then
    Store the discovered risky FC
  end if
  Update $Q$ backward according to (8) or (10)
  Update the availability of actions backward
  Update (decrease) $\epsilon_1$
end for

Obtain representative risky FCs from the stored risky FCs

IV. RL-BASED OFFLINE AND ONLINE SEARCH STRATEGIES

A. Consecutive Learning

The operation status of real-world power systems varies with time. Therefore, a series of different power flow snapshots can be extracted. Each of them has a set of representative risky FCs, which should be identified online individually within the duration of the snapshot.

Considering the gradual changes in two adjacent power flow snapshots, the knowledge of representative risky FCs can be shared. Therefore, this paper proposes a knowledge transition process, which is inspired by the idea of transfer learning [26]. The process is divided into two parts: offline initialization and consecutive online transition-extension, as shown in Fig. 4.

The initialization is the acquisition of initial knowledge of FCs. At the beginning, which FCs of the system are risky is unknown. Therefore, high-risk states and actions need to be identified from scratch. This step may be very time-consuming but fortunately only needs to be executed once at the beginning.

Transition-extension is the application and continuous update of knowledge. It is performed after new power flow snapshots are obtained.

The transition prefers to exploit previously learned knowledge. Unless large disturbances occur, the difference between two adjacent dispatch snapshots is usually insignificant as the time interval is short, typically 15 minutes. So risky FCs in the previous snapshot may also remain risky in the upcoming snapshot.

In contrast, the extension tends to explore less “valuable” paths, because actions seemingly not dangerous may be actually risky but have not been discovered before. Moreover, because of the variation of power flow, some low-risk FCs may become risky in the new power flow snapshot. Therefore, moderate additional trials can be helpful complements to the historical experience exploitation and are needed to identify more risky FCs.

After the online refinement, the revised knowledge $Q$ is then applied to the next snapshot, which launches a new transition-extension process.

B. Initialization: Search without Previous Knowledge

In initialization, trials are rather random and blind. The agent in MDP cannot evaluate how much risk is caused by an outage of a transmission line, until the agent removes it and simulate the subsequent events. Therefore, the agent needs to use an index to estimate the severity of each outage and guide the search, which is called prior knowledge. According to experience, the heavier the power flow of a component is, the higher impact on the grid of its outage is imposed. Therefore, a power-flow-weighted (PF-weighted, or PFW for short) index $W_{PF}$ based on Softmax function [25] is proposed here to determine the probability of an action $a_s$

$$\Pr(a_s|s) = \frac{W_{PF}(C_j|s)}{\sum_{j=1}^{m} W_{PF}(C_j|s)} \quad (11)$$
where $PF(C_i|s)$ is component $C_i$’s power flow under state $s$, $N(s, a_i)$ is the visit count of action $a_i$ (removal of $C_i$) at the current state, and $k_s$ is an adjustable positive parameter. The parameter $k_s$ is necessary, because the value of $PF(C_i|s)$ may be very large to cause arithmetic overflow. In this paper, $k_s$ is set to be

$$k_s = \frac{\ln E}{\max_i(PF(C_i|s)/\sqrt{N(s, a_i)} + 1)}$$

where $E$ is a constant used to adjust the agent’s preference. A larger $E$ means that the agent is more likely to remove heavily loaded components. According to (13), the maximum value of $W_{PF}$ is $E$.

The probability $\epsilon_1$ is decided by the power flow and visit count of the initial state $s_0$

$$\epsilon_1 = \max_{i} \left( \frac{\sum_{i=1}^{m} (PF(C_i|s_0)/\sqrt{N(s_0, a_i)} + 1)}{\sum_{i=1}^{m} PF(C_i|s_0)}, \epsilon_{1m} \right)$$

(14)

where $\epsilon_{1m}$ is used to maintain the minimum level of exploitation.

If the agent would like to exploit obtained knowledge $Q$, it will select components with max weighted $Q$

$$a_i = \text{arg} \max_i \frac{Q(s_i, a_i)}{\sqrt{N(s_i, a_i)} + 1}$$

(15)

C. Transition-Extension: Search Based on Knowledge

After initialization, the knowledge of risky FCs of a certain power system is stored in $Q$. When the updated power flow snapshot is available, the agent can search risky FCs of the snapshot based on the previous knowledge and the new knowledge of the power flow snapshot.

1) Knowledge transition: the selection of actions can be guided by the previous knowledge $Q_0$, which is the $Q$ of the last snapshot. Therefore, in the transition stage, the probability of an action is determined as follows

$$\Pr(a_i|s) = \frac{W_{Q0}(a_i|s)}{\sum_{j=1}^{m} W_{Q0}(a_j|s)}$$

(16)

$$W_{Q0}(a_i|s) = \exp\left(\frac{Q_0(s_0, a_i)}{\sqrt{N(s, a_i)} + 1}\right) \times \text{sgn}(Q_0(s, a_i))$$

(17)

It is different from (15), because externally learned knowledge $Q_0$ is used instead of $Q$.

2) Knowledge extension: if the agent decides to extend its knowledge, it takes the same strategy as the initialization (11)-(12). Since the knowledge extension is used online, the limit for the number of search trials is much fewer than that of initialization.

3) Online mixed strategy: to further enhance the search efficiency, this paper proposes a mixed strategy to balance transition and extension. Similar to the $\epsilon$-greedy strategy in Algorithm 1, the mixed strategy uses another probability $\epsilon_2$ to transfer the previous knowledge $Q_0$: for a search, if a random number $\text{rand}(0,1) \leq \epsilon_2$, components are removed according to (16) by the principle of transition; if not, according to (11) by extension. The $\epsilon_2$ is also decided by the visit count of the initial state $s_0$

$$\epsilon_2 = \min(\epsilon_{20} \times \frac{\sum_{i}(Q(s_0, a_i)/\sqrt{N(s_0, a_i)} + 1)}{\sum_{i} Q(s_0, a_i)}, 1)$$

(18)

where $\epsilon_{20}$ is an initial constant. Like $\epsilon_1$, the $\epsilon_2$ also decreases gradually, so the agent inclines to transfer the previous knowledge at the early stage, then more likely to extend the knowledge as the number of search trials increases.

The online step actually has three strategies. Firstly, the agent has a probability $\epsilon_1$ of exploration. Then according to $\epsilon_2$, the agent will decide to whether explore according to previous knowledge (16) or power flow (11). The procedure is shown in Algorithm 2.

<p>| TABLE I |</p>
<table>
<thead>
<tr>
<th>TEST PLATFORM AND SIMULATION PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>New England</td>
</tr>
<tr>
<td>Hardware</td>
</tr>
<tr>
<td>Software</td>
</tr>
<tr>
<td>PF Type</td>
</tr>
<tr>
<td>$\gamma$ in (7)</td>
</tr>
<tr>
<td>$M$ in Alg. 1</td>
</tr>
<tr>
<td>$TH%$ in Alg. 1</td>
</tr>
<tr>
<td>$E$ in (13)</td>
</tr>
<tr>
<td>$\epsilon_{1m}$ in (14)</td>
</tr>
<tr>
<td>$\epsilon_{20}$ in (18)</td>
</tr>
</tbody>
</table>

V. CASE STUDY

In this section, numerical simulations are performed to validate the proposed FC search framework based on RL and knowledge transfer. The New England 39-bus test system is used to study the feasibility and effectiveness of the proposed approach. Then a test case on a real-world provincial grid is used to test the effectiveness of the proposed consecutive learning. The DC power flow is adopted in this paper to simulate cascading outages. Other models compatible with Algorithm 1, including AC power flow model and others mentioned in [27] are also applicable to the proposed framework. The specifications of the test platform and the simulation parameters are listed in Table I, where the value of $TH\%$ is a specific percentage of the total load of the system.
TABLE II
Comparison of FCs

<table>
<thead>
<tr>
<th>Snapshot</th>
<th>Risky FCs</th>
<th>Shared FCs</th>
<th>Similar FCs</th>
<th>Unique FCs</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>3303</td>
<td>3244</td>
<td>50</td>
<td>9</td>
</tr>
<tr>
<td>S2</td>
<td>3383</td>
<td>3244</td>
<td>50</td>
<td>9</td>
</tr>
</tbody>
</table>

Fig. 5. 2-dimensional comparison of shared chains

A. IEEE 39-Bus New England Test System

There are 39 buses (including 10 generators) and 46 branches (including 12 transformers and 34 transmission lines) in the New England test system. In order to verify the feasibility, a security screening of $N - 2$ and $N - 3$ contingencies is conducted. As it is a small system, the computational cost is affordable.

1) Verification of the similarity of FCs between two similar snapshots: FCs from two similar but different power flow snapshots should be firstly compared, because the similarity is the premise of knowledge transfer. In order to illustrate the similarity and difference, FCs are compared and then their similarity is classified as follows:

1) Two FCs $F_1$ and $F_2$ are defined as shared FCs if their denotations are identical.

2) If they are not shared but $F_1$ is a subset or superset of $F_2$, they are called similar FCs. Similar FCs are also helpful for knowledge transfer. The number of similar FCs of two snapshots may be different because one FC may be similar to many FCs of the other snapshot.

3) Otherwise, they are unique FCs to each other. If an FC of snapshot $S_2$ is unique to every FC of snapshot $S_1$, it is unique to $S_1$. The number of unique FCs may also be unequal.

For example, FC ($L_1$, $L_2$) is similar to both ($L_1$, $L_2$, $L_3$) and ($L_1$, $L_4$, $L_2$), while the latter two are unique to each other.

Two power flow snapshots are constructed and compared. $S_1$ is the base case, and $S_2$ is constructed by amplifying the injecting power of all buses (generators and substations) of $S_1$ proportionally to 1.01 times (denoted as 1.01x). The result is shown in Table II. It can be seen that most of the FCs are shared, verifying the similarity of the two power flow snapshots and thus the rationality of the knowledge transfer.

The similarity is then verified in a broader range. Seven power flow snapshots are constructed in the same way as $S_2$, and the amplification rate $\lambda$ varies from 0.90x with an increment of 0.02x. Limited by $N - 1$ security, the maximum value of the amplification rate $\lambda_{\text{max}} = 1.013$, therefore the last snapshot is not 1.02x but 1.01x. Comparisons of 2-dimensional FC pairs are shown in Fig. 5 and Fig. 6.

The number of risky FCs increases with the growth of $\lambda$, which is expected. Snapshots with closer amplification rate $\lambda$ have more shared risky FCs. So the effect of knowledge transition may be better if the two power flow snapshots are more similar. However, no matter how close the two snapshots are, there are always some different (similar or unique) FCs, which also proves the necessity of knowledge extension proposed in section IV.

2) Verification of the efficiency of RL and knowledge transfer: firstly, the efficiency of the proposed RL mechanism is illustrated, compared to strategies without RL. Two kinds of prior knowledge are applied. One is purely random search (PR), where the agent always randomly selects an available component to remove with equal weight. The other one is PFW-weighted (PFW) search, where the agent selects component with weights that are determined by (11). Both two strategies are then combined with RL according to Algorithm 1, where the agent removes components according to prior or learned knowledge. Therefore, four strategies are tested and compared, denoted as PR, PFW, PR+RL, PFW+RL, respectively. The numbers of search attempts needed to cover 95% of all risky FCs are shown in Table III.

Methods with RL perform better than those without RL, meaning that they find more risky FCs than their non-RL counterparts at the same search trials. This is because RL...
In online application scenarios, the four methods in Fig. 7 can eventually identify all risky FCs, though later its rising speed becomes slower. Therefore, the PFW search does outperform the purely random search at the beginning. REI does at the same number of search trials. The PFW search has covered 95% of all risky FCs and 95.6% of the accumulated risk of the system. Meanwhile, the REI method only covers 65.8% of accumulated risk at the same search trials.

Four more different power flow snapshot pairs are also tested. Results in Fig. 9 shows that the efficiency of the transition-extension process is influenced by the difference between the adjacent snapshots, which accords with the similarity comparison shown in Fig. 5. The nearly but not fully flat slope of 4 curves after turning points highlights the necessity of knowledge extension, especially when the loading level of the latter snapshot is higher than the previous snapshot.

B. 1000-Bus Sichuan Grid Model: Consecutive Learning

Different from the New England 39-node test system, there are backbone buses (500kV) as well as many peripheral buses with a lower voltage (≤220kV), which make the structure complicated. Sichuan Grid model has 1080 buses and 1628 branches. However, not all of them are available in each dispatch cycle (15 minutes) because of unit start-stop and switching actions. Therefore, the topology of Sichuan Grid may vary slightly and the concrete number of buses and branches of each snapshot are different. Meanwhile, the change in topology is mild and gradual. 923 (85.5%) of the buses remain unchanged during the day, which guarantees the similarity of snapshots of Sichuan Grid. It can be inferred from the number that there exists more than \(10^{18}\) possible fault combinations, which cannot be enumerated.

The fluctuation of the load (as shown in Fig. 10) varies asynchronously, which differs from the New England test system. Such a characteristic results in a more irregular tendency. In addition, the loading level of Sichuan Grid model is rather low, which means that the FCs able to trigger very fast process are longer but fewer. Therefore, FCs not longer than 6 branches are concerned in this case. The representativeness of risky FCs is...
important in Sichuan Grid model. For example, while the PFW method finds 397 and 903 risky FCs in snapshot 00:00 and 18:00 respectively, only 80 and 181 of them are representative, of which the proportions are both lower than 20%.

1) Efficiency of RL within a single snapshot: in each dispatch cycle, a power flow snapshot is retrieved from Energy Management System (EMS). In this part, two power flow snapshots of 00:00, and 18:00 are used for test, which represent lightly and heavily loaded scenarios respectively. As for search strategies, REI, PFW, and PFW+RL are performed with search trials of 50000 times on each of the three snapshots. Numbers of representative risky FCs identified by each method are shown in Fig. 11. The result of purely random strategy is not listed because it only finds very few risky FCs in Sichuan Grid model.

In this large grid, RL still has a moderate effect of acceleration. However, REI does not perform so well as in the small New England test system. It is because Sichuan Grid model is overall lightly loaded, making the secondary risk (the third part) of REI not distinguishable. So in the lightly loaded scenario of 00:00 snapshot, REI identifies even fewer representative risky FCs than PFW and PFW+RL. In the following part, when compared to the knowledge transfer method, only PFW is concerned.

2) Power of knowledge transfer during online consecutive learning: 96 real power flow snapshots of Sichuan Grid model on a certain day are used for online learning. When a new snapshot is obtained, the online transition-extension learning method is applied. Before the first 00:00 snapshot, the offline initialization is performed based on the 21:00 snapshot on the previous day.

The source snapshot is selected mainly because it is heavily loaded and thus has more risky FCs. A search of $9.6 \times 10^5$ times is performed, and then the knowledge is transferred to the 00:00 snapshot. These search trials guarantees that enough representative risky FC can be found while avoid overfitting. Every online transition-extension process searches 50000 times, which costs only 9-12 minutes and is shorter than the duration of a dispatch cycle, conforming to the requirement of online application. If an FC is not transferred successfully, its risk is discounted by multiplying $\gamma$, and then is passed to the next snapshot. For a fair comparison, the control group search each snapshot 60000 times according to (11) (PFW without RL or knowledge transfer), as if always treated as the initialization procedure. As a result, the two methods have the same amount of total search trials ($5.76 \times 10^6$) in a day and both cost about 19 hours. Therefore, the proposed mechanism of RL and knowledge transfer does not substantially affect the computational efficiency.

As shown in Fig. 12, the consecutive RL-based learning can always identify more representative risky FCs than the search without RL or T-E. On average, the proposed method finds 86.7% more than the control group. Test results also show that T-E finds more and more representative risky FCs than the control group as the time elapses, which reflects the effective accumulation of knowledge in the proposed method. During the former half of the day, the T-E finds 103 more representative risky FCs than its rival on average, while the number grows to 130 during the afternoon.

Fig. 13 shows the search process of discovered representative risky FCs by the two methods. The proposed T-E approach first prefers to transfer learned external knowledge, which is the major contributor to the effective identification of risky FCs reflected by the sharp rise of the 3 blue solid curves. Then it turns to extension and exploitation of its own learned knowledge, resulting in a slower but still lasting increase, which is the same as Fig. 7 and Fig. 9.

3) Discussions on power flow models: To prove that the proposed Algorithm 1 is compatible with AC power flow, a preliminary test is applied to the 00:00 snapshot of Sichuan Grid in this part. This test uses AC power flow based on the process of physical simulation proposed by [28], of which the results are shown in Table IV.

According to the results, the time consumed by a search trial
with AC power flow model is 10.4 times that with DC power flow on average. In order to meet the requirement of the online application, the computational time for the identification of online representative risky FCs should be less than a dispatch cycle. Therefore, fewer search trials are performed. When searching 6000 times, the identification of representative risky FCs under AC power flow model consumes 15 minutes, which is equal to the dispatch cycle. Meanwhile, these 6000 search trials find 83 representative risky FCs combined with the proposed PFW+RL technique, which is close to the number of DC power flow. Although the number of search trials of AC power flow is less than that of DC power flow, the AC power flow model is able to find out risky FCs that cannot be identified by DC power flow by considering the voltage and reactive power. According to this preliminary result, both AC and DC power flow models are acceptable and practical for online application. The authors will do further research based on AC power flow in the future.

VI. CONCLUSION

To identify representative risky fault chains more efficiently, this paper proposes an online search method based on reinforcement learning and knowledge transfer. The search for fault chains is modeled as a Markov Decision Process, and then a Q-learning-based method is applied to the search, highlighting the utilization of historical and current knowledge. After the mechanism of knowledge transition and extension established, strategies for offline training and online application are presented. Test results of New England test system and Sichuan Grid model demonstrate the effectiveness and the efficiency of the proposed approach, which is promising for online deployment. Based on the proposed approach, the authors also plan to exploit the identified fault chains to block or mitigate cascading outages in future studies.

However, it should be noted that producing fault chains is just the first step for blackout mitigation. The authors would exploit fault chains to block or mitigate cascading outages in the future.

ACKNOWLEDGMENT

The authors would like to thank Kun Song, Boda Li, Dr. Yue Xia and Dr. Junjian Qi, for their valuable suggestions.
during the writing and revision of the paper.

REFERENCES


Zhimei Zhang (S’16) received the B.E. degree in electrical engineering from Tsinghua University, Beijing, China, in 2006, and the Ph.D. degree in electrical engineering in 2013, where he is currently working toward the Ph.D. degree in electrical engineering. His research interest includes computational analysis of power systems.

Shaowei Huang received the B.S. and Ph.D. degrees from the Department of Electrical Engineering, Tsinghua University, Beijing, China, in July 2006 and June 2011, respectively. From 2011 to 2013, he worked as a post-doctoral in the Department of Electrical Engineering, Tsinghua University. He is currently an Associate Professor in the Department of Electrical Engineering, Tsinghua University. His research interests include power systems modeling and simulation, power system parallel and distributed computing, complex systems and its application in power systems, artificial intelligence.

Ying Chen (M’07) received the B.E. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 2001 and 2006, respectively, both in electrical engineering. He is currently an associate professor with the Department of Electrical Engineering, Tsinghua University. His research interests include parallel and distributed computing, electromagnetic transient simulation, cyber-physical system modeling, and cyber security of the smart grid.

Shengwei Mei (SM’06–F’14) received the B.S. degree in mathematics from Xinjiang University, Urumuqi, China, in 1984, the M.S. degree in operations research from Tsinghua University, Beijing, China, in 1989, and the Ph.D. degree in automatic control from the Chinese Academy of Sciences, Beijing, China, in 2001. He is currently a Professor with the Department of Electrical Engineering, Tsinghua University. His research interests include power system analysis and control, game theory and its application in power systems.
Rui Yao (S’12–M’17) received the B.S. degree (with distinction) in 2011 and Ph.D. degree in 2016 in electrical engineering at Tsinghua University, Beijing, China. He was a postdoctoral research associate at the University of Tennessee, Knoxville during 2016–2018. He is currently a postdoctoral appointee at Argonne National Laboratory. His research interests include power system modeling and stability analysis, resilience modeling and assessment, and high-performance computational methodologies.

Kai Sun (M’06–SM’13) received the B.S. degree in automation and the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China, in 1999 and 2004, respectively. He conducted postdoctoral studies at Arizona State University (2006-2007) and the University of Western Ontario (2005). He was a Project Manager in grid operations and planning with EPRI, Palo Alto, CA, USA, from 2007 to 2012. He is currently an Associate Professor with the Department of EECS, the University of Tennessee, Knoxville, TN, USA.

His research interests include stability, dynamics and control of power grids, and other complex systems. Prof. Sun serves on the Editorial Boards of the IEEE Transactions on Smart Grid, IEEE ACCESS, and IET Generation, Transmission and Distribution.