Heuristics-Based Approach for Identifying Critical $N - k$ Contingencies in Power Systems

Saqib Hasan, Amin Ghafouri, Abhishek Dubey, Gabor Karsai, Xenofon Koutsoukos
Institute for Software Integrated Systems,
Vanderbilt University, Nashville, TN 37212, USA
Email:{saqib.hasan, amin.ghafouri, abhishek.dubey, gabor.karsai, xenofon.koutsoukos}@vanderbilt.edu

***Abstract*—Reliable operation of electrical power systems in the presence of multiple critical $N - k$ contingencies is an important challenge for the system operators. Identifying all the possible $N - k$ critical contingencies to design effective mitigation strategies is computationally infeasible due to the combinatorial explosion of the search space. This paper describes two heuristic algorithms based on the iterative pruning of the candidate contingency set to effectively and efficiently identify all the critical $N-k$ contingencies resulting in system failure. These algorithms are applied to the standard IEEE-14 bus system, IEEE-39 bus system, and IEEE-57 bus system to identify multiple critical $N - k$ contingencies. The algorithms are able to capture all the possible critical $N - k$ contingencies (where $1 \leq k \leq 9$) without missing any dangerous contingency.**

***Index Terms*—Blackouts, Cascading failures, $N-k$ contingency analysis, Heuristic algorithms, Resilience.**

I. Introduction

Power systems are complex electrical networks consisting of several physical (e.g., transmission lines, energy sources etc.) and computational (e.g., protection devices, PMU's etc.) components which are tightly coupled together. Reliable operation of these systems is of primary importance for the system operators. Based on the North American Electric Reliability Corporation (NERC) standards [1], these systems are generally operated according to the $N-1$ security criterion, where failure of any single component would not result in violation of branch flows, bus voltage or stability limits. Thus, system operators are routinely able to manage $N-1$ contingencies. However, it becomes challenging to deal with multiple simultaneous $N-k$ contingencies (where $k \geq 2$) which initiate severe cascading failures resulting in blackouts. Examples of such cases are Feb 2016 South Australia [2], Dec 2015 Ukraine [3], July 2012 India [4] and August 2003 North America [5] blackouts. Thus, NERC standards at present require the grid operators to operate power systems against cascading failures resulting from multiple contingencies [1].

Identifying all the possible critical $N - k$ contingencies is computationally infeasible for larger systems and higher values of k because of combinatorial explosion of the search space. For a specific power system, ignoring the sequence, $N - k$ contingency analysis requires $\frac{N!}{k!(N-k)!}$ number of simulations to identify all the critical contingencies. This number grows exponentially (N^k) as N increases. For instance, consider a power system with $N = 5000$, as the total number of components. To identify all the possible critical $N - 4$ contingencies causing cascading failures resulting in blackouts, a total of approximately 26×10^{12} simulations are needed to be performed. Hence, exhaustive search is infeasible.

Various approaches have been developed to reduce the computational complexity while identifying multiple critical contingencies [6]–[19]. Primitive contingency analysis techniques [6]–[10] are based on the ranking and selection of outages. As part of the ranking and selection techniques, contingencies are ranked and selected depending upon the performance index for voltage analysis, line flows, capacity, and power flow analysis. Event trees are used in identifying critical contingencies in [11]. A concept of delta centrality in [12] and line outage distribution factors in [13], [15] are used to identify groups of multiple $N - k$ contingencies. The work in [18] presented a fast $N - 2$ contingency analysis algorithm based on [13], [15], which performs pruning of the contingency set. The work in [14] provides a method based on iteratively selecting random subsets which are pruned to obtain collections of multiple contingencies causing system failure. In addition, small groups of severe multiple contingencies can be identified using the optimization algorithms proposed in [16], [17], [19].

In order to improve system reliability and resilience, efficient and effective ways to identify severe cascade causing contingencies are necessary. This paper presents a new approach towards identifying all possible critical $N - k$ contingencies causing cascading failures resulting in blackouts. The approach focuses on evaluating contingencies causing severe cascading failures. The contributions from this paper are:

- We present an algorithm (Algorithm I) that uses previously identified critical $N - k$ contingencies to identify the critical contingencies from the subsequent $N - k$ contingency analysis by pruning the current contingency candidate set. This reduces the computational burden accompanied for ranking contingencies based on the above mentioned approaches.
- We present an improved algorithm (Algorithm II) that uses the frequency distribution of the contingencies appearing in the candidate contingency set and combines it with Algorithm I to employ a 2-stage pruning process identifying all the critical $N-k$ contingencies. According to this distribution, most of the critical contingencies tend to fall within a specific region of the frequency distribution curve as shown in our evaluation section.

- We evaluate our approach using case studies on the standard IEEE-14 bus system [20], IEEE-39 bus system [21], and IEEE-57 bus system [22]. Our results show that the algorithms are able to capture all the critical $N-k$ contingencies without missing any dangerous system failure causing contingency. The approach largely reduces the computational effort and takes significantly less time to identify these critical contingencies as compared to the exhaustive search. Moreover, these algorithms are based on the iterative pruning of the search space which results in very few simulations.

The remainder of the paper is organized as follows. Section II presents the heuristic algorithms to identify critical $N-k$ contingencies. Section III discusses the cascade simulation framework used for simulating the power systems. Section IV demonstrates the results followed by the conclusions in Section V.

II. Contingency Analysis

In this section, we present two algorithms to identify critical $N-k$ contingencies in a power transmission system. We consider a power system $\mathcal{G}_p$ which consists of components such as buses, transmission lines, transformers, loads, and generators. The purpose of these components is to supply sufficient power from the generating stations to the loads. Failure(s) can occur in one or more component of the power system. We refer to these failures as $N-k$ contingencies, where the value of k defines the number of simultaneous multiple failures in a system consisting of N components. These $N-k$ contingencies may cause severe cascading outages resulting in a system failure, where system failure is defined by a user-supplied criterion that represents a blackout, e.g., power loss greater than or equal to 40% of the total power needed. Further, the system failure causing contingencies are referred to as critical $N-k$ contingencies. Finally, $N-k$ *contingency analysis* is defined as analyzing k simultaneous failures to understand their effects on the rest of the power network.

A. Algorithm I

First, we present Algorithm I which is based on the iterative pruning of the current candidate contingency set using the previously identified critical $N-k$ contingencies. For example, to identify critical components from $N-2$ analysis, we use the identified critical contingencies from $N-1$ analysis to prune the candidate contingency set for $N-2$ analysis. Let $\mathcal{U}$ represent the universal set of all the N possible component outages in a power system. Given a value of k, we denote by $\mathcal{S}_k$ the entire search space, defined by $\mathcal{S}_k = \{a \mid a \in 2^{\mathcal{U}}, |a| \leq k\}$. Further, we let C_f denote the system failure criterion.

Let $\mathcal{F} \in \mathcal{S}_{k'}$ be a contingency. If $\mathcal{F}$ causes a system failure, in the subsequent $N-k$ contingency analysis (where $k > k'$), we assume any other contingency $\mathcal{F}' \in \mathcal{S}_k$, satisfying $\mathcal{F} \subseteq \mathcal{F}'$, also causes a system failure. This assumption seems to hold true for most scenarios because, if $\mathcal{F}$ causes a system failure then intuitively $\mathcal{F}'$ will outage more number of components from the system. This will weaken the system more and result in larger damage. However, this assumption does not always hold true. That is, in some rare cases, even if $\mathcal{F}$ results in a system failure, $\mathcal{F}'$ will not cause a system failure (i.e., loss less than C_f) and eventually leads to a stable state. However, most of these $\mathcal{F}'$ still causes cascading failures that results in quite a significant loss within the system.

For example, consider a power system with universal set $\mathcal{U}$ containing transmission lines $tl_1, tl_2, \ldots, tl_m$. In $N-1$ contingency analysis, if an outage $\mathcal{F} = \{tl_a\}$ satisfies C_f, then $\mathcal{F}$ is marked as a critical contingency. Next, in $N-2$ contingency analysis, any contingency $\mathcal{F}' = \{tl_a, tl_i\}$, where $i \in \{1, \ldots, m\} - \{a\}$, is assumed to cause a system failure. Therefore, the candidate pairs are pruned from the search space $\mathcal{S}_2$ and are not considered for simulation.

Algorithm 1 Algorithm for Finding $N-k$ Contingencies

Input: $\mathcal{G}_p, \mathcal{U}, C_f, k$
Initialize: $\mathcal{T} \leftarrow \emptyset, \mathcal{R} \leftarrow \emptyset, c_{pre} \leftarrow 0$
for all $\mathcal{F} \in \mathcal{S}_1$ **do**
  $loss \leftarrow$ `simulate_contingency`$(\mathcal{G}_p, \mathcal{F})$
  if $loss \geq C_f$ **then**
    $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{F}$
  end if
end for
for $p = 2, \ldots, k$ **do**
  $\mathcal{P} \leftarrow \emptyset, \mathcal{R}_{cur} \leftarrow \emptyset$
  for all $\mathcal{F}' \in \mathcal{S}_p$ **do**
    for all $\mathcal{F} \in \mathcal{R}$ **do**
      if $\mathcal{F} \subseteq \mathcal{F}'$ **then**
        $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{F}'$
      end if
    end for
  end for
  $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{P}$
  $\hat{\mathcal{S}}_p \leftarrow \mathcal{S}_p \setminus \mathcal{P}$ ▷ prunes search space $\mathcal{S}_p$
  for all $\mathcal{F} \in \hat{\mathcal{S}}_p$ **do**
    $loss \leftarrow$ `simulate_contingency`$(\mathcal{G}_p, \mathcal{F})$
    if $loss \geq C_f$ **then**
      $\mathcal{R}_{cur} \leftarrow \mathcal{R}_{cur} \cup \mathcal{F}$
    end if
  end for
  $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}_{cur}$
  if $|\mathcal{R}_{cur}| \leq c_{pre}$ **then**
    break
  end if
  $c_{pre} \leftarrow |\mathcal{R}_{cur}|$
end for
return $\mathcal{T}$

The algorithm takes the power system model $\mathcal{G}_p$, the N possible component outage set $\mathcal{U}$, system failure criterion C_f, and contingency range k as inputs. Further, it identifies the total number of critical $N-k$ contingencies denoted by $\mathcal{T}$. The set of critical contingencies causing system failure that are identified through simulations is denoted by $\mathcal{R}$. The set of predicted $N-k$ contingencies resulting in system failure using the set $\mathcal{R}$ is denoted by $\mathcal{P}$. The algorithm evaluates each contingency denoted by $\mathcal{F}$ using the function `simulate_contingency`$(\mathcal{G}_p, \mathcal{F})$ and adds it to $\mathcal{R}$, if the loss due to $\mathcal{F}$ is greater than or equal to the system failure criterion C_f. The function `simulate_contingency`$(\mathcal{G}_p, \mathcal{F})$ is a contingency simulator described in Section III. Given a value of p ranging from 1 to k, the algorithm identifies the search

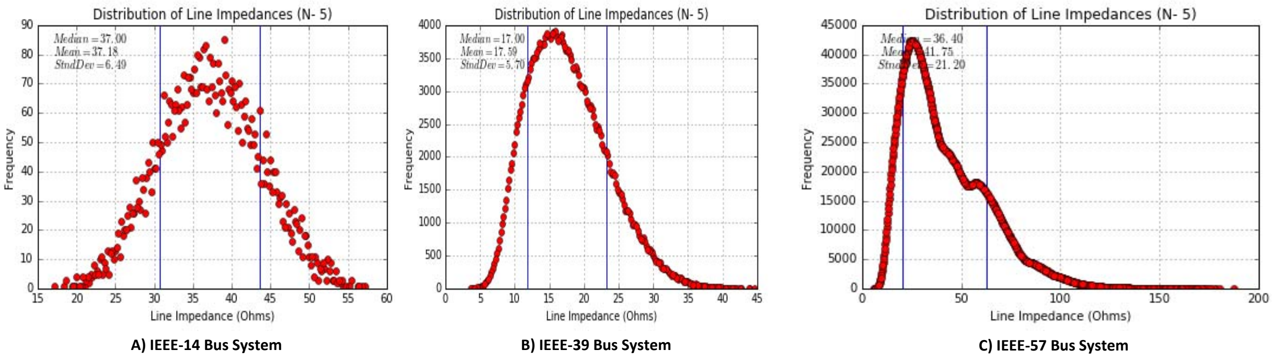

Fig. 1: Frequency distribution curves of the candidate contingency set ($\mathcal{S}_5$) for different standard power systems.

space $\mathcal{S}_p$ for the next iteration. In each iteration p, it evaluates if an element of $\mathcal{R}$ is a subset of an element in $\mathcal{S}_p$ depending upon which the elements are placed in $\mathcal{P}$. The set $\hat{\mathcal{S}}_p$ represents the pruned set of contingencies that are needed to be simulated using the function `simulate_contingency`$(\mathcal{G}_p, \mathcal{F})$ in order to identify critical $N-k$ contingencies that are not captured during the prediction stage. Using $\hat{\mathcal{S}_p}$, missed critical contingencies $\mathcal{F}$ that satisfied C_f are identified and $\mathcal{R}$ is updated accordingly. This further improves the pruning process in the subsequent iterations. The algorithm is terminated either after k iteration, or when the number of current identified critical contingencies obtained through simulations are less than or equal to the number of identified critical contingencies at the previous iteration ($|\mathcal{R}_{cur}| \leq c_{pre}$).

B. Algorithm II

In Algorithm II, we use the frequency distribution curve representing the frequency distribution of the candidate contingency set and the idea from Algorithm I to employ a 2-stage pruning of the candidate contingency set $\mathcal{S}_k$. This curve represents the frequency with which a contingency $\mathcal{F}$ with impedance $\mathcal{Z}(\mathcal{F})$ appear within the search space $\mathcal{S}_k$. The frequency distribution curve of different standard IEEE systems are shown in Figure 1. The x-axis represents the impedance of an individual contingency. Further, the y-axis describes the frequency with which individual contingency impedances appear within the search space $\mathcal{S}_k$. For any transmission line $a \in \mathcal{U}$, let z_a denote its impedance. Given a value of k and a contingency $\mathcal{F} \in \mathcal{S}_k$, the mean impedance $\mathcal{Z}(\mathcal{F})$ of the contingency is

$$\mathcal{Z}(\mathcal{F}) = \frac{\sum_{a \in \mathcal{F}} z_a}{|\mathcal{F}|} \tag{1}$$

The average impedance $\bar{\mathcal{Z}}_k$ of the frequency distribution curve for a search space $\mathcal{S}_k$ is

$$\bar{\mathcal{Z}}_k = \frac{\sum_{\mathcal{F} \in \mathcal{S}_k} \mathcal{Z}(\mathcal{F})}{|\mathcal{S}_k|} \tag{2}$$

where $|\mathcal{S}_k|$ is the total number of contingencies.

Further, the standard deviation $\sigma_{\mathcal{Z}}$ of the frequency distribution is defined by

$$\sigma_{\mathcal{Z}} = \sqrt{\frac{\sum_{\mathcal{F} \in \mathcal{S}_k} (\mathcal{Z}(\mathcal{F}) - \bar{\mathcal{Z}}_k)^2}{|\mathcal{S}_k|}} \tag{3}$$

Considering a window size within the frequency distribution

Algorithm 2 Algorithm for Finding $N-k$ Contingencies

Input: $\mathcal{G}_p, \mathcal{U}, C_f, \mathcal{Z}_w, k$
Initialize: $\mathcal{T} \leftarrow \emptyset, \mathcal{R} \leftarrow \emptyset$, $c_{pre} \leftarrow 0$
for all $\mathcal{F} \in \mathcal{S}_1$ **do**
  $loss \leftarrow$ `simulate_contingency`$(\mathcal{G}_p, \mathcal{F})$
  if $loss \geq C_f$ **then**
    $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{F}$
  end if
end for
for $p = 2, \ldots, k$ **do**
  $\mathcal{P} \leftarrow \emptyset, \mathcal{R}_{cur} \leftarrow \emptyset, \mathcal{S}_p^{'} \leftarrow \emptyset$
  for all $\mathcal{F} \in \mathcal{S}_p$ **do**
    if $\mathcal{Z}(\mathcal{F}) \notin \mathcal{Z}_w$ **then**
      $\mathcal{S}_p^{'} \leftarrow \mathcal{S}_p^{'} \cup \mathcal{F}$ ▷ prunes search space $\mathcal{S}_p$
    end if
  end for
  $\mathcal{T} \leftarrow \mathcal{T} \cup (\mathcal{S}_p \setminus \mathcal{S}_p^{'})$
  for all $\mathcal{F}' \in \mathcal{S}_p^{'}$ **do**
    for all $\mathcal{F} \in \mathcal{R}$ **do**
      if $\mathcal{F} \subseteq \mathcal{F}'$ **then**
        $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{F}'$
      end if
    end for
  end for
  $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{P}$
  $\hat{\mathcal{S}}_p \leftarrow \mathcal{S}_p^{'} \setminus \mathcal{P}$ ▷ prunes search space $\mathcal{S}_p^{'}$
  for all $\mathcal{F} \in \hat{\mathcal{S}}_p$ **do**
    $loss \leftarrow$ `simulate_contingency`$(\mathcal{G}_p, \mathcal{F})$
    if $loss \geq C_f$ **then**
      $\mathcal{R}_{cur} \leftarrow \mathcal{R}_{cur} \cup \mathcal{F}$
    end if
  end for
  $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}_{cur}$
  if $|\mathcal{R}_{cur}| \leq c_{pre}$ **then**
    break
  end if
  $c_{pre} \leftarrow |\mathcal{R}_{cur}|$
end for
return $\mathcal{T}$

curve (e.g., as shown by blue lines in Figure 1), most critical $N-k$ contingencies fall in this region. Hence, there is a higher probability of picking a critical contingency within this region. The window size is then obtained by

$$\mathcal{Z}_w = [\bar{\mathcal{Z}}_k - \sigma_{\mathcal{Z}}, \bar{\mathcal{Z}}_k + \sigma_{\mathcal{Z}}] \quad (4)$$

Based on the assumption from the frequency distribution curve, in Algorithm II, a contingency $\mathcal{F} \in \mathcal{S}_k$ that appears within $\mathcal{Z}_w$ is pruned from the search space $\mathcal{S}_k$. This is referred to as stage-1 prediction and pruning. After stage-1 pruning of $\mathcal{S}_k$, further pruning is done based on the same approach as Algorithm I. This provides a stage-2 prediction and pruning, which further improves the efficiency of our method.

The Algorithm takes the power system model $\mathcal{G}_p$, the N possible component outage set $\mathcal{U}$, system failure criterion C_f, frequency distribution curve window size denoted by $\mathcal{Z}_w$ and contingency range k as inputs. Further, it identifies the total number of critical $N-k$ contingencies denoted by $\mathcal{T}$. The set of critical contingencies causing system failure that are identified through simulations is denoted by $\mathcal{R}$. The set of predicted $N-k$ contingencies resulting in system failure using the set $\mathcal{R}$ is denoted by $\mathcal{P}$. The algorithm evaluates each contingency denoted by $\mathcal{F}$ using the function `simulate_contingency`$(\mathcal{G}_p, \mathcal{F})$ and adds it to $\mathcal{R}$, if the loss due to $\mathcal{F}$ is greater than or equal to the system failure criterion C_f. Given a value of p ranging from 1 to k, the algorithm identifies the search space $\mathcal{S}_p$ for the next iteration. In each iteration p, it evaluates if a contingency $\mathcal{F} \in \mathcal{S}_p$ does not exist within the specified region of the frequency distribution curve denoted by $\mathcal{Z}_w$, it is added to $\mathcal{S}_p^{'}$. This step defines the stage-1 pruning of the candidate contingency set $\mathcal{S}_p$.

Further, in the same iteration p, the algorithm evaluates if an element of $\mathcal{R}$ is a subset of an element in $\mathcal{S}_p^{'}$ depending upon which the elements are placed in $\mathcal{P}$. This step mark the stage-2 pruning of the search space $\mathcal{S}_p$. The set $\hat{\mathcal{S}_p}$ represents the pruned set of contingencies that are needed to be simulated in order to identify critical $N-k$ contingencies that are not captured during the prediction stage. Using $\hat{\mathcal{S}_p}$, missed critical contingencies $\mathcal{F}$ that satisfied C_f are identified and $\mathcal{R}$ is updated accordingly. This further improves the pruning process in the subsequent iterations. The algorithm is terminated either after k iteration, or when the number of current identified critical contingencies obtained through simulations are less than or equal to the number of identified critical contingencies at the previous iteration ($|\mathcal{R}_{cur}| \leq c_{pre}$).

Note, when the system size becomes too large, the two algorithms can be iteratively used over the subset of the search space $\mathcal{S}_k$ to identify critical $N-k$ contingencies. Another possible solution can be to run the algorithms on the subset of the search space $\mathcal{S}_k$ over a distributed computing platform. Moreover, the approach will still be able to capture all the possible critical contingencies without missing any dangerous contingency. In addition, we use simple data structures while implementing these heuristics. However, the efficiency of these algorithms can further be improved by making use of efficient data structures, such as trees etc., if needed. This will further improve the analysis results discussed in Section IV.

III. CONTINGENCY SIMULATOR

In this section, we describe our contingency simulator framework. There are various cascade simulation models and each of these models have their own assumptions, capabilities and limitations [23], [24], [25]. Among these models, there is no standard cascade simulation model for simulating cascading failures. In this work, we select a commonly used cascade simulation model, but it should be noted that the considered cascade model can be easily replaced by any other model while keeping the algorithms fixed.

We have developed a contingency simulator framework by integrating OpenDSS power system model and Cascade simulation model with the OpenDSS contingency simulator, which is a modified version of the simulator used in [26] for Simscape Models. The OpenDSS contingency simulator is an OpenDSS-based AC power flow solver for power systems [27]. The simulator allows us to capture critical $N-k$ contingencies causing severe cascading outages resulting in system failure. Further, the identified critical contingencies can help operators design effective mitigation strategies.

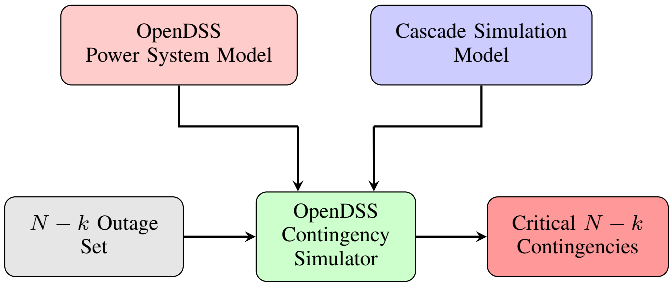

Fig. 2: Cascade Simulator Framework

The contingency simulator framework is shown in Figure 2, where the inputs to the simulator are the OpenDSS power system model, cascade simulation model, and $N-k$ outage set. The $N-k$ outage set is the set of contingencies that are needed to be simulated and analyzed. Further, the contingency simulator analyzes each contingency based on the cascade simulation model and identifies the set of critical $N-k$ contingencies.

The flowchart for the cascade simulation model is shown in Figure 3. Initial outages in the form of $N-k$ contingencies are given to the simulator. After the initial outages, the OpenDSS power system model is executed by solving power flow and the system is evaluated for overloads. If an overload is observed, identified branches are tripped and the system is evaluated for the system failure (i.e., blackout) criterion. Here, the system failure criterion is considered to be a load loss of 40% or more, which is one of the criterion in [28]. If the criterion is met then the contingency is marked as a critical $N-k$ contingency. However, if the criterion is not met and the system is still overloaded, then the overloaded branches are tripped until all the overloaded branches are eliminated, or the system satisfies the system failure criterion, or the system reaches a stable state (as described by Safe state in Figure 3). Moreover, if there are

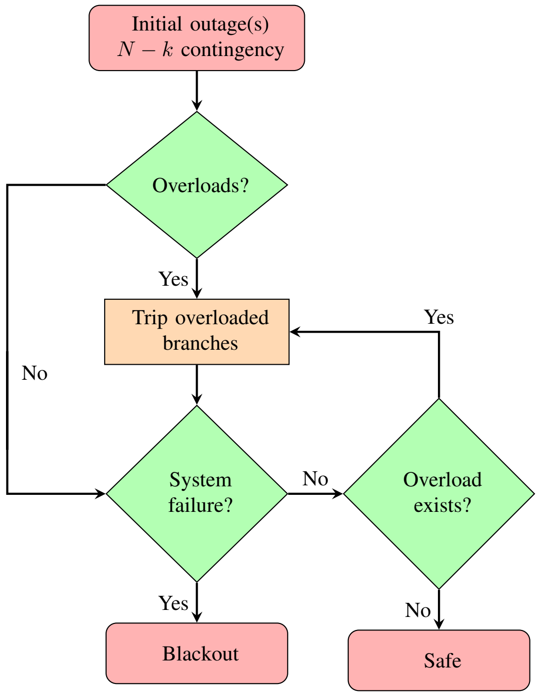

Fig. 3: Cascade Simulation Model

no overloads after the initial outages then the system is directly evaluated for the system failure criterion. If the criterion is not satisfied then the contingency is marked as Safe.

IV. Evaluation

To validate and test the developed algorithms, we apply them to the standard IEEE-14 bus system, IEEE-39 bus system, and IEEE-57 bus system. We noticed that the two algorithms are able to identify all the possible critical $N-k$ contingencies without missing any dangerous $N-k$ outage resulting in system failure. Furthermore, we observed that the heuristics are much more effective and efficient as compared to the exhaustive search and use significantly smaller number of simulations to predict the actual number of critical $N-k$ contingencies compared to exhaustive search.

A. Execution Time Analysis of the Algorithms

First, we compare the time complexity of the two algorithms with the exhaustive search. $N-9$, $N-5$, $N-4$ contingency analysis are performed on IEEE-14 bus system, IEEE-39 bus system and IEEE-57 bus system respectively. Figure 4 shows the time complexity results of these systems. In each figure, x-axis represents the value of k and the y-axis represents the time taken to perform the contingency analysis given a value of k. The red, green, and blue lines represent the execution time for exhaustive search, Algorithm I, and Algorithm II, respectively. Figure 4 clearly shows that Algorithm I and Algorithm II are much faster than the exhaustive search for all k values. Moreover, Algorithm II seems to be even faster than Algorithm I. This is because of the 2-stage pruning performed in each iteration of Algorithm II, which reduces the search space significantly and improves the algorithm's efficiency.

The exhaustive search to perform $N-4$ contingency analysis for IEEE-57 bus system identifies a total of 346,214 system failure contingencies from a total of 722,865 contingencies. Using Algorithm I, which requires performing only 24,469 simulations, 345,662 critical $N-k$ contingencies out of the 346,214 critical contingencies are identified. To identify the remaining 552 critical $N-4$ contingencies, Algorithm I uses 259,600 simulations in its final iteration. If the computational cost of running these 259,600 simulations to identify the remaining 552 critical $N-4$ contingencies is high, the algorithm can terminate prior to these simulations. This will significantly improve the execution time of Algorithm I, as shown by the green dotted line in Figure 4.C. The same approach can similarly be applied to the other considered systems but it is not shown due to figure clarity.

B. Reduction in the Total Number of Simulations

Now, we compare the total number of simulations needed to identify critical $N-k$ contingencies using exhaustive search, Algorithm I, and Algorithm II. The analysis on the standard IEEE systems shows that Algorithm I and Algorithm II effectively reduce the total number of simulations as compared to the exhaustive search and still are capable of identifying all the critical $N-k$ contingencies. Figure 5 shows the total number of simulations used by the exhaustive search, Algorithm I, and Algorithm II to identify all the critical $N-k$ contingencies. The y-axis represents the total number of simulations used by each algorithm. The red, green, and blue bars represent the number of simulations performed by the exhaustive search, Algorithm I, and Algorithm II, respectively. Figure 5.A shows that in IEEE-14 bus system, there are 89845, 3095, and 1734 number of simulations performed for the exhaustive search, Algorithm I, and Algorithm II to identify all the possible critical $N-k$ contingencies (where $k=9$). Further, Figure 5.B shows that in IEEE-39 bus system, there are 1676115, 117536, and 48046 number of simulations carried out for the exhaustive search, Algorithm I and Algorithm II to identify the circital $N-k$ contingencies (where $k=6$). As show in Figure 5, the numbers of performed simulations are significantly reduced by Algorithm I and Algorithm II as compared to the exhaustive search. In addition, Algorithm II performs much better than Algorithm I in terms of reducing the number of simulations.

The total number of simulations can be further reduced in both of our algorithms by avoiding running simulations for scenarios where large number of simulations results in identifying only a very few critical $N-k$ contingencies. In our case, for Algorithm I, if the 259,600 simulations that will capture only 552 critical contingencies are avoided, the total number of simulations will be reduced to only 24,469 simulations (marked by the black-colored region in Figure 5.C). Similarly, for Algorithm II, the total number of simulations can be reduced to only 7,413 simulations (marked by the orange-colored region in Figure 5.C). This is a significant reduction in the number of simulations if there is a leverage to identify most but not all critical contingencies.

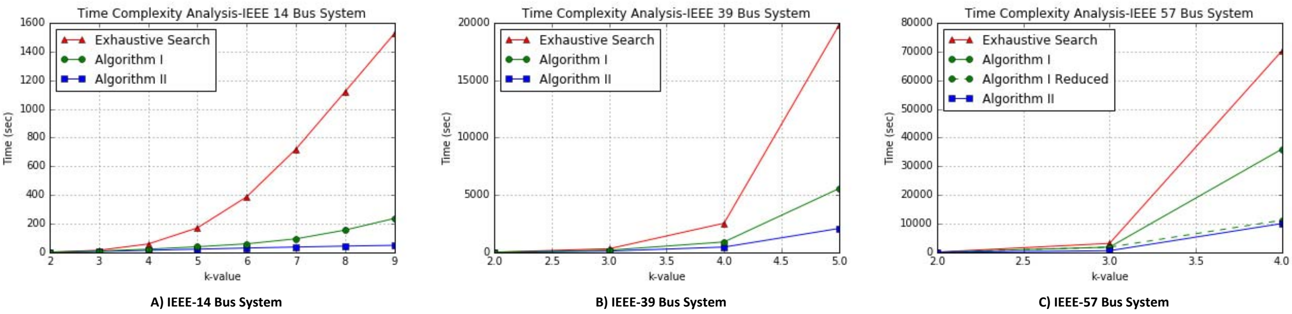

Fig. 4: Execution Time Analysis-Time taken by Exhaustive search, Algorithm I and Algorithm II to Identify Critical $N-k$ Contingencies

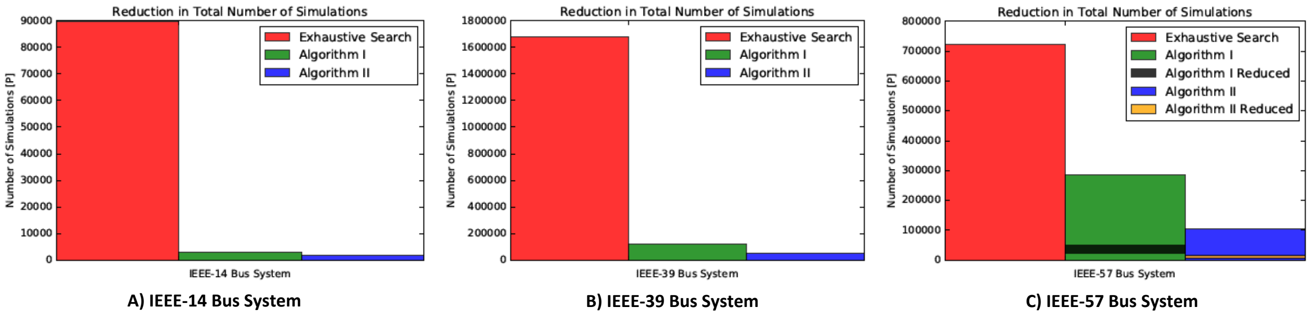

Fig. 5: Total Number of Simulations Run using Exhaustive Search, Algorithm I and Algorithm II to Identify Critical $N-k$ Contingencies

C. Performance Accuracy of the Algorithms

First, the effectiveness of Algorithm II is shown using Figure 6. During the stage-1 prediction and pruning process in Algorithm II, we identified a total of 14,968 out of 19,778 and a total of 15,272 out of 21,879 critical contingencies for IEEE-14 bus system and IEEE-39 bus system respectively. These critical contingencies are identified without any simulations. Based on our heuristics for the frequency distribution curve, most critical $N-k$ contingencies are expected and do fall within the region shown by blue lines (Figure 6), which represents the window size defined by $\mathcal{Z}_w$ in Section II. Furthermore, most of the remaining critical contingencies that are not captured in stage-1 prediction and pruning process are identified using the stage-2 prediction and pruning process. In addition, only a very few critical contingencies are needed to be identified using simulations. Thus, all the critical $N-k$ contingencies are identified through minimum computational effort.

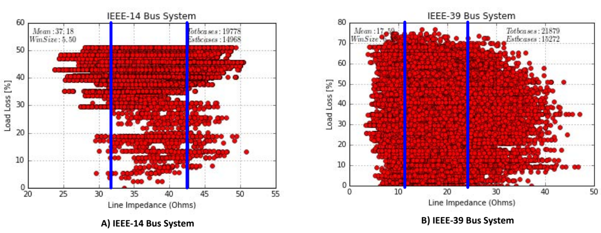

Fig. 6: Effectiveness of Stage-1 Prediction and Pruning Process of Algorithm II

Now, we compare the performance accuracy of the two algorithms. Performance accuracy is a measure of the ability of these algorithms to capture the number of critical $N-k$ contingencies when compared with the exhaustive search. In Figure 7, the x-axis represents the value of k and the y-axis represents the total number of identified critical $N-k$ contingencies. The red, green and blue lines represent the identified critical $N-k$ contingencies using exhaustive search, Algorithm I, and Algorithm II respectively. Figure 7 shows that Algorithm I and Algorithm II are nearly the same with respect to their performance accuracy, that is, they capture nearly the same number of critical contingencies. In IEEE-14 bus system, a total of 69749, 86733, and 88111 critical $N-k$ contingencies (where, $k=9$) are identified using exhaustive search, Algorithm I and, Algorithm II respectively. In IEEE-39 bus system, a total of 1068603, 1558545, and 1628069 $N-k$

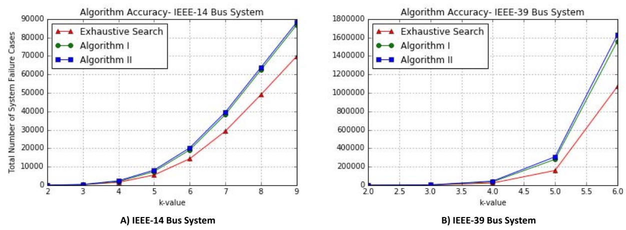

Fig. 7: Prediction Accuracy of Algorithm I and Algorithm II

contingencies (where, $k = 6$) resulting in system failure are identified using exhaustive search, Algorithm I and Algorithm II respectively. Performance accuracy results for IEEE-57 bus system are not shown but can be obtained similarly.

As shown in Figure 7, the two algorithms are capable of capturing all the critical $N - k$ contingencies that can be identified using exhaustive search. Moreover, they also capture and classify some non-critical contingencies as critical. But, most of these non-critical contingencies still cause cascading failures that result in significant losses within the system. Figure 6 shows most of the non-critical contingencies within the region marked by blue lines cause significant loss. If remained not addressed, these non-critical contingencies combined with another outage can instantly put the system in a non-recoverable state. In addition, these identified non-critical contingencies together with the critical contingencies from the algorithms can further help operators improve the system reliability and resilience.

V. Conclusions

Two heuristic algorithms were developed for $N - k$ contingency analysis problem. The idea for both these heuristics is based on the iterative pruning of the candidate contingency set $\mathcal{S}_k$. The pruning process is based on the previously identified critical $N - k$ contingencies and the information from the frequency distribution curve of the $N - k$ candidate contingency set. Even though the approach is based on the developed heuristics, it captures all the critical $N - k$ contingencies without missing any dangerous contingency resulting in system failure. The algorithms are validated and tested on the standard IEEE-14 bus system, IEEE-39 bus system and IEEE-57 bus system. The results described in Section IV demonstrate that these heuristics perform much better than the exhaustive search by reducing the computational time and minimizing the total number of simulations.

Although the results prove the effectiveness of the two heuristics, they can further be improved by using efficient data structures, such as trees etc. that could improve the overall efficiency of the algorithms. Another approach is to use the concept of distributed computing, where the algorithms can be run in parallel on multiple cores to optimize the approach further. Additionally, for very large candidate contingency set, a smaller critical subset can be used to identify critical $N-k$ contingencies. Furthermore, the identified critical $N-k$ contingencies can be used by system operators to design effective failure detection and mitigation strategies to improve system resilience and reliability [29]. As part of the future work, larger power transmission systems can be analyzed and the approach can be applied to radial distribution power networks to perform $N-k$ contingency analysis for identifying its performance.

Acknowledgment

This work is funded in part by the National Science Foundation under the award number CNS-1329803 and the FORCES project under the award number CNS-1238959.

References

[1] S. NERC, "Top-004–2: Transmission operations," *North American Electric Reliability Corporation*, 2007.
[2] A. Reese, "Blackouts cast Australia's green energy in dim light," 2017.
[3] T. Pultarova, "Cyber security-Ukraine grid hack is wake-up call for network operators [news briefing]," *Engineering & Technology*, vol. 11, no. 1, pp. 12–13, 2016.
[4] J. J. Romero, "Blackouts illuminate India's power problems," *IEEE spectrum*, vol. 49, no. 10, 2012.
[5] U.-C. Force, "Final report on the august 14th blackout in the United States and Canada," *Department of Energy and National Resources Canada*, 2004.
[6] G. Ejebe and B. Wollenberg, "Automatic contingency selection," *IEEE Transactions on Power Apparatus and Systems*, no. 1, pp. 97–109, 1979.
[7] T. Mikolinnas and B. Wollenberg, "An advanced contingency selection algorithm," *IEEE Transactions on Power Apparatus and Systems*, no. 2, pp. 608–617, 1981.
[8] G. Irisarri and A. Sasson, "An automatic contingency selection method for on-line security analysis," *IEEE transactions on power apparatus and systems*, no. 4, pp. 1838–1844, 1981.
[9] M. K. Enns, J. J. Quada, and B. Sackett, "Fast linear contingency analysis," *IEEE Transactions on Power Apparatus and Systems*, no. 4, pp. 783–791, 1982.
[10] G. Ejebe, H. Van Meeteren, and B. Wollenberg, "Fast contingency screening and evaluation for voltage security analysis," *IEEE Transactions on Power Systems*, vol. 3, no. 4, pp. 1582–1590, 1988.
[11] Q. Chen and J. D. McCalley, "Identifying high risk nk contingencies for online security assessment," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 823–834, 2005.
[12] R. D. Dosano, H. Song, and B. Lee, "Network centrality based nk contingency scenario generation," in *Transmission & Distribution Conference & Exposition: Asia and Pacific, 2009*. IEEE, 2009, pp. 1–4.
[13] C. M. Davis and T. J. Overbye, "Multiple element contingency screening," *IEEE Transactions on Power Systems*, vol. 26, no. 3, 2011.
[14] M. J. Eppstein and P. D. Hines, "A random chemistry algorithm for identifying collections of multiple contingencies that initiate cascading failure," *IEEE Transactions on Power Systems*, vol. 27, no. 3, 2012.
[15] C. Davis and T. Overbye, "Linear analysis of multiple outage interaction," in *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*. IEEE, 2009, pp. 1–8.
[16] V. Donde, V. López, B. Lesieutre, A. Pinar, C. Yang, and J. Meza, "Severe multiple contingency screening in electric power systems," *IEEE Transactions on Power Systems*, vol. 23, no. 2, pp. 406–417, 2008.
[17] D. Bienstock and A. Verma, "The nk problem in power grids: New models, formulations, and numerical experiments," *SIAM Journal on Optimization*, vol. 20, no. 5, 2010.
[18] K. S. Turitsyn and P. Kaplunovich, "Fast algorithm for n-2 contingency problem," in *System Sciences (HICSS), 2013 46th Hawaii International Conference on*. IEEE, 2013, pp. 2161–2166.
[19] C. M. Rocco, J. E. Ramirez-Marquez, D. E. Salazar, and C. Yajure, "Assessing the vulnerability of a power system through a multiple objective contingency screening approach," *IEEE Transactions on Reliability*, vol. 60, no. 2, pp. 394–403, 2011.
[20] http://icseg.iti.illinois.edu/ieee-14-bus-system/, IEEE-14 Bus System.
[21] http://icseg.iti.illinois.edu/ieee-39-bus-system/, IEEE-39 Bus System.
[22] http://icseg.iti.illinois.edu/ieee-57-bus-system/, IEEE-57 Bus System.
[23] Y. Koç, T. Verma, N. A. Araujo, and M. Warnier, "Matcasc: A tool to analyse cascading line outages in power grids," in *Intelligent Energy Systems (IWIES), 2013 IEEE International Workshop on*. IEEE, 2013.
[24] B. A. Carreras, D. E. Newman, I. Dobson, and N. S. Degala, "Validating opa with wecc data." in *HICSS*, 2013, pp. 2197–2204.
[25] D. P. Nedic, I. Dobson, D. S. Kirschen, B. A. Carreras, and V. E. Lynch, "Criticality in a cascading failure blackout model," *International Journal of Electrical Power & Energy Systems*, vol. 28, no. 9, pp. 627–633, 2006.
[26] S. Hasan, A. Chhokra, A. Dubey, N. Mahadevan, and G. Karsai, "A simulation testbed for cascade analysis," in *IEEE PES Innovative Smart Grid Technologies (ISGT)*, 2017.
[27] O. P. Model and O. S. Element, "Opendss manual," *EPRI,[Online] Available at: http://sourceforge. net/apps/mediawiki/electricdss/index. php*.
[28] http://sites.ieee.org/pes-cascading/presentations/, PES General Meeting.
[29] A. Ghafouri, A. Laszka, A. Dubey, and X. Koutsoukos, "Optimal detection of faulty sensors used in route planning," in *Science of Smart City Operations and Platforms Engineering (SCOPE)*, 2017.