

## Fast Algorithm for N-2 Contingency Problem

K. S. Turitsyn, P. A. Kaplunovich  
 Department of Mechanical Engineering  
 Massachusetts Institute of Technology  
 Cambridge, USA  
 Email: turitsyn@mit.edu, pekap@mit.edu

### Abstract

*We present a novel selection algorithm for  $N-2$  contingency analysis problem. The algorithm is based on the iterative bounding of line outage distribution factors and successive pruning of the set of contingency pair candidates. The selection procedure is non-heuristic, and is certified to identify all events that lead to thermal constraints violations in DC approximation. The complexity of the algorithm is  $O(N^2)$  comparable to the complexity of  $N-1$  contingency problem. We validate and test the algorithm on the Polish grid network with around 3000 lines. For this test case two iterations of the pruning procedure reduce the total number of candidate pairs by a factor of almost 1000 from 5 millions line pairs to only 6128.*

### 1. INTRODUCTION

Maintaining reliable operation of the power system is of paramount importance for the power grid operators and society as a whole. This task will likely become even more challenging due to combination of multiple factors, that include shift toward intermittent renewable generation, electric transportation systems, deregulation of energy markets. The standards developed by North American Electric Reliability Corporation [1] necessitate the operators to ensure the system performance in the events of multiple outage contingencies. However, the problem of contingency identification remains computationally challenging due to combinatorial explosion of the total number of possible initiating events. This number grows approximately as  $N^k$  where  $N$  is the number of components (typically branches of the network) and  $k$  is the number of outaged elements.

Large number of algorithms have been developed to address the problem of computational complexity. The classical approaches towards contingency identification are based on ranking and selection approaches [2–7]. Within the ranking framework the candidate outage configurations are ranked according to heuristic performance index based on the line flow, capacity as well as the total number of lines in the network. Multiple variations of the method exist differing in the functional form of the performance index. The selection approach [3, 7] is based on the analysis of

power flow solutions and provide more accurate ranking at the expense of additional computational burden. A number of modifications to both methods have been proposed in the recent years that have significantly improved the efficiency of the ranking procedure. These include the approaches based on the network topology analysis [8–10], nonlinear optimization heuristics [11–13] and others. Our work is most closely related to the approaches based on the Line Outage Distribution Factors that have been recently explored in [14, 15].

In this paper we develop a new approach towards contingency selection problem that is based on iterative pruning of the contingency candidate set. Starting with a set of all possible 2 line outage pairs we exclude the pairs that are guaranteed to be “safe” from the contingency perspective. The corresponding guarantees can be shown using the analytic bounds for the line overload expression based on the Line Outage Distribution Factors computed within the stage of  $N-1$  contingency analysis. For realistic cases with small number of contingencies this pruning procedure allows one to filter out most of the line combinations leaving only few potentially dangerous ones. If the number of the final candidates is  $O(N)$  or less they can be analyzed directly with negligible computational overhead. Unlike most of the other approaches, our algorithm is not based on any uncontrollable heuristics. It is guaranteed to capture all the dangerous events without missing any pairs leading to violations. In this manuscript we describe the algorithm for  $N-2$  contingency analysis, its extensions to more general  $N-k$  problem will be reported elsewhere. The overall complexity of the algorithm depends on the efficiency of the power flow solution procedure and on the total amount of contingencies violating thermal constraints. In the relatively unstressed situations when the total number of contingencies is small the complexity can be estimated as  $O(RN)$  where  $R$  is the number of operations required to solve the linear power flow equations. The overall complexity is therefore comparable to the  $N-1$  contingency problem that is routinely solved by system operators.

The structure of this paper is the following. In section 2 we formally define the problem and derive the key relations necessary for the construction of the algorithm. In section 3

we describe the actual algorithm and discuss the issues of complexity, implementation and possible optimizations. Next, in section 4 we present the results of algorithm validation and various tests on the 3000 bus Polish grid model. Finally, the overview of the approach as well as possible extensions and research directions are presented in section 5.

## 2. PROBLEM SETTING

In this work we limit ourselves to DC approximation which is also used in most of the other  $N - k$  contingency studies. Although it's accuracy can be limited in some situations it is a reasonable model for an already challenging  $N - k$  contingency problem. Within this approximation the state of the power system is described by the vector of voltage phases  $\theta_k$  defined on every of the  $M$  buses in the system. The power flows are described by the linear dc power flow equations:

$$\mathbf{B}\theta = p \quad (1)$$

where  $\hat{B}$  is the  $M \times M$  nodal DC susceptance matrix and  $p$  is the vector of active power injections. The nodal DC susceptance matrix can be represented as  $\mathbf{B} = \mathbf{M}\mathbf{Y}\mathbf{M}^T$ , where  $\mathbf{Y}$  is the diagonal  $N \times N$  matrix of branch susceptances, and  $\mathbf{M}$  is the  $M \times N$  connection matrix with 1s indicating the beginning bus of every branch, and  $-1$  its end. The vector of power flows can be represented as  $f = \mathbf{Y}\mathbf{M}^T\theta = \mathbf{Y}\mathbf{M}^T\mathbf{B}^{-1}p$ .

Linear DC power flow admit a very simple and elegant analysis of the single and multiple line contingencies. There is conservation of total power flowing through the system, so whenever one or multiple line outage, the power that was flowing through them is distributed between the other lines in the system. Linear structure of the equations allows one to describe this distribution via linear mapping. The effect of the outage can be described by the matrix of so called Line Outage Distribution Factors (LODF) denoted as  $L_{yx}$  that relates the change of flow in a monitored line  $y$  that follows after the tripping of line  $x$  with original flow  $f_x$ . Formally one can write:

$$L_{yx} = \frac{f'_y - f_y}{f_x} \quad (2)$$

relates the change of the flow through line  $y$  from  $f_y$  to  $f'_y$  with the flow  $f_x$  through line  $x$  before the outage. The LODFs are extensively used for the  $N - 1$  contingency analysis. They can be computed in  $O(NK)$  operations, which is an acceptable overhead on top of the amount of calculations required to solve power flow equations. In the following discussion we assume that the matrix  $L_{xy}$  has been precomputed. As we will show, it is possible to express the overload effect of the double outage in terms of the expression for single outage LODF. This relation forms the basis of our algorithm that efficiently utilizes the information

available from  $N - 1$  contingency analysis to identify a tight set of double outage contingency candidates.

In order to find the relation between single and two line contingency LODFs we use the well-known expression for the LODF in general  $k$ -line contingency situation (see e.g. [6]):

$$\mathbf{L} = \mathbf{Y}\mathbf{M}^T\mathbf{B}^{-1}\tilde{\mathbf{M}}(\mathbf{1} - \tilde{\mathbf{Y}}\tilde{\mathbf{M}}^T\mathbf{B}^{-1}\tilde{\mathbf{M}})^{-1}, \quad (3)$$

where  $\tilde{\mathbf{M}}$  is the  $M \times k$  submatrix of  $\mathbf{M}$  corresponding to the outaged lines and similarly  $\tilde{\mathbf{Y}}$  is the  $k \times k$  outaged line submatrix of  $\mathbf{Y}$ . This expression is applicable both to single ( $n = 1$ ) and double  $n = 2$  line outage events. Direct comparison of these expressions allows us to relate the two LODF matrices. After straightforward but bulky calculations we arrive at the following expression for the effect of double outage:

$$f'_z - f_z = \frac{L_{zx}(f_x + L_{xy}f_y)}{1 - L_{yx}L_{xy}} + \frac{L_{zy}(f_y + L_{yx}f_x)}{1 - L_{yx}L_{xy}}. \quad (4)$$

In this relation we denote the outage lines by  $x, y$  and consider the change of the flow on some arbitrary line  $z$ . The expressions  $L_{xy}$  correspond to the single line outage as defined in (2). Similar expression, although written in a different form has been recently derived in [15]. For some combinations of initially tripped lines  $x, y$  the denominator  $1 - L_{yx}L_{xy}$  can be zero. It was shown in [16] that such situations correspond to the islanding of the grid. After the grid is islanded the rank of the matrix  $\mathbf{B}$  in (1) is increased and it may not have a solution. This corresponds to the situation when individual islands do not have balanced generation and consumption. The restoration of the balance depends on the system operator policies and is not considered in this work. In our algorithm we substitute the corresponding elements of the matrix  $A_{xy}$  with zeros which automatically removes them from consideration. There are only few of such cases in the model of Polish Grid studied in this work. All of them correspond to islanding of single buses. The important property of (4) that is extensively exploited in our algorithm is the factorization of individual terms in (4). After introduction of  $A_{xy} = (1 + L_{xy}f_y/f_x)/(1 - L_{yx}L_{xy})$  the expression (4) can be rewritten as

$$f'_z - f_z = A_{xy}L_{zx}f_x + A_{yx}L_{zy}f_y \quad (5)$$

The contingency occurs whenever the absolute value of the flow at line  $z$  exceeds a critical value, i.e.  $f'_z > f_z^{\text{crit}}$  or  $f'_z < -f_z^{\text{crit}}$ . Both of these conditions can be rewritten in the form

$$A_{xy}B_{xc} + A_{yx}B_{yc} > 1 \quad (6)$$

where the  $c$  indicates one of the flow constraints, and there are two values of  $c$  associated with each line  $z$  with the matrix values given by  $B_{xc} = f_x L_{zx} / (f_z^{\text{crit}} \pm f_z)$ , where the  $+, -$  signs correspond to the conditions  $f'_z < -f_z^{\text{crit}}$  and  $f'_z > f_z^{\text{crit}}$  respectively. The form (6) is rather general,

and can be used for other types of linear constraints, such as voltage bus ones. Although these constraints are not discussed in this work, in the following we will assume that the sets of constraints and lines are separate and the elements of the matrix  $B_{xc}$  are not necessarily associated with individual line overloads. We denote the set of possible constraints  $c$  by  $\mathcal{C}$  and the set of all lines by  $\mathcal{E}$ . In these notations the problem is reduced to selection of all tuples  $(x, y)$  with  $x, y \in \mathcal{E}$  such that  $1 - L_{xy}L_{yx} \neq 0$  for which there exists at least one constraint  $c \in \mathcal{C}$  that satisfies the condition of line overload:

$$\Gamma_{xyc} + \Gamma_{yxc} > 1 \quad (7)$$

where  $\Gamma_{xyc} = A_{xy}B_{xc}$ . Brute force search of all such tuples requires in the worst case scenario requires at least  $O(N^2K)$  operations where  $N = |\mathcal{E}|$  is the number of branches and  $K = |\mathcal{C}|$  is the total number of constraints. If the only constraints are associated with line overloads  $K = 2N$ . The iterative pruning approach described dramatically lowers this estimate in practical situation when the total number of tuples is small. In this case the complexity of the algorithm can be estimated as  $O(NK) + O(N^2)$ .

### 3. ALGORITHM

Our algorithm is based on the simple idea of iterative pruning of the set of initiating line candidates. The algorithm exploits the algebraic structure of the overload condition (7). Although both of the terms  $\Gamma_{xyc}$  and  $\Gamma_{yxc}$  depend on three indices  $x, y, c$ , these dependence has a factorized form  $\Gamma_{xyc} = A_{xy}B_{xc}$ . This form admits a fast bounding procedure that results in an upper bound that depends only on two indices, for instance  $\Gamma_{xyc} \leq \Gamma_{xy*}^{\max}$ . This bound can be produced by finding the minimal  $B_{x*}^{\min}$  and maximal  $B_{x*}^{\max}$  values of  $B_{xc}$  for every value of  $z$ :  $B_{x*}^{\min} \leq B_{xc} \leq B_{x*}^{\max}$  and can be found by direct iteration over the matrix  $B_{xc}$  in only  $O(|\mathcal{E}| \cdot |\mathcal{C}|)$  operations. The expression for  $\Gamma_{xy*}^{\max}$  is given by

$$\Gamma_{xy*}^{\max} = \begin{cases} A_{xy}B_{x*}^{\max}, & A_{xy} \geq 0 \\ A_{xy}B_{x*}^{\min}, & A_{xy} < 0 \end{cases} \quad (8)$$

that can be compactly written as  $\Gamma_{xy*}^{\max} = \max\{A_{xy}B_{x*}^{\max}, A_{xy}B_{x*}^{\min}\}$ . As the bound  $\Gamma_{xy*}^{\max}$  depends only on two indices, it can be used for fast pruning of the set  $\mathcal{A}$  of possible  $(x, y) \in \mathcal{A}$  tuple candidates. Whenever  $\Gamma_{xy*}^{\max} + \Gamma_{yx*}^{\max} \leq 1$ , the condition (7) can not be satisfied for any possible choice of  $z$ . Thus, the pruning of set  $\mathcal{A}$  can be accomplished in only  $O(|\mathcal{A}|)$  operations which is at most  $O(N^2)$ . Analogous upper bounds can be constructed for  $\Gamma_{x*c}^{\max}$  and  $\Gamma_{*yc}^{\max}$  to prune the set of pairs  $x, z$  that can be part of the triple satisfying (7). The detailed algorithm is presented in three listings below. The main function `findTuples` takes the set  $\mathcal{E}$  of possible initiating lines and set  $\mathcal{C}$  of all the relevant constraints as an input and returns the set of possible candidate tuples  $\mathcal{A}$  as the output.

The pruning happens in iterative fashion as each reduction of one set produces better bounds on the matrices  $A, B$  and allows extra pruning of the second set. In the step 3

---

#### Algorithm 1 `findTuples` ( $\mathcal{E}, \mathcal{C}$ )

---

```

1:  $\mathcal{A} \leftarrow \{(x, y) : x, y \in \mathcal{E}\}$ 
2:  $\mathcal{B} \leftarrow \{(x, c) : x \in \mathcal{E}, c \in \mathcal{C}\}$ 
3: repeat
4:   Calculate  $B_{*c}^{\max}, B_{*c}^{\min}$  ▷ Prune  $\mathcal{B}$ 
5:   Calculate  $A_{x*}^{\max}, A_{x*}^{\min}, A_{*y}^{\max}, A_{*y}^{\min}$ 
6:   for  $(x, c) \in \mathcal{B}$  do
7:      $\Gamma_{x+c}^{\max} \leftarrow \max\{A_{x*}^{\max}B_{xc}, A_{x*}^{\min}B_{xc}\}$ 
8:      $\Gamma_{*yc}^{\max} \leftarrow \max\{A_{*y}^{\max}B_{*c}^{\max}, A_{*y}^{\min}B_{*c}^{\min}\}$ 
9:   end for
10:   $\mathcal{B} \leftarrow \{(x, c) \in \mathcal{B} : \Gamma_{x+c}^{\max} + \Gamma_{*xc}^{\max} > 1\}$ 
11:  Calculate  $B_{x*}^{\min}, B_{x*}^{\max}$  ▷ Prune  $\mathcal{A}$ 
12:  for  $(x, y) \in \mathcal{A}$  do
13:     $\Gamma_{xy*}^{\max} \leftarrow \max\{A_{xy}B_{x*}^{\max}, A_{xy}B_{x*}^{\min}\}$ 
14:  end for
15:   $\mathcal{A} \leftarrow \{(x, y) \in \mathcal{A} : \Gamma_{xy*}^{\max} + \Gamma_{yx*}^{\max} > 1\}$ 
16: until  $\mathcal{A}$  stops changing
17: return  $\mathcal{A}$ 

```

---

we have omitted the definition of  $B_{x*}^{\min} = \min_{(x,c) \in \mathcal{B}} B_{xc}$  and its obvious counterparts for the sake of presentation simplicity. The sets  $\mathcal{A}, \mathcal{B}$  can be implemented via different data structures. The simplest, although not the most efficient choice is to simply use boolean masks for the matrices  $A_{xy}, B_{xc}$ . In this case both the iteration over the sets  $\mathcal{A}, \mathcal{B}$  in lines 6, 12 and the filtering operations in lines 10, 15 can be implemented as a direct loop over all possible values. In this implementation the total complexity of the algorithm will be given by  $O(INK) + O(IN^2)$  where  $I$  is the number of outer loop iterations. More sophisticated implementations of the sets can significantly reduce the number of inner loop iterations for small set cardinalities and thus improve the overall complexity. In general, we expect that the total number of outer loop iterations necessary for the algorithm to converge will be of order 2 – 4 for the realistic situations with small number of contingencies. This observation is supported by our numerical experiments, but its formal proof is far beyond the scope of our work.

Apart from various implementation possibilities there is also an additional degree of freedom related to the definition of the matrices  $A_{xy}$  and  $B_{xc}$ . The expression  $\Gamma_{xyc} = A_{xy}B_{xc}$  is invariant under the transformation  $A_{xy} \rightarrow s_x A_{xy}, B_{xc} \rightarrow s_x^{-1} B_{xc}$  for any non-zero values of  $s_x$ . This transformation affects the value of the bound  $\Gamma_{*xc}^{\max}$  on line 8 and can be used for improving the efficiency of the pruning process. Our preliminary results indicate that it is possible to reduce the size of the final set  $\mathcal{A}$  by a factor of 2 via careful choice of  $s_x$ . However, this reduction comes at the expense of substantial computational overhead. Nevertheless, this optimization may become important in situations where

the unoptimized pruning procedure is inefficient for some reasons.

It is also possible to improve the efficiency of the pruning procedure by appropriate subdivision of the constraint set  $\mathcal{C}$ . As the bounds  $B_{x^*}^{\max}$  and others are based on the analysis of the whole set of branches, few outliers in this set can significantly affect the value of the bounds. For example, a single line  $z$  with flow  $f_z$  very close to the capacity  $f_z^{\text{crit}}$  can inflate the values of  $B_{x^*}^{\max}$  for all initiating lines  $x$  and thus affect the efficiency of pruning. It is possible to mitigate this problem by subdivision of the constraint set  $\mathcal{C}$  and separate analysis of the outlier and all the other lines. We are currently exploring these possibilities and will report our findings in future publications.

#### 4. RESULTS

In order to validate and test the proposed algorithm we have used the Polish grid model available in MATPOWER package [17]. This grid consists of 3269 lines and 2737 buses. Our simulations have started with the base state found via solution of OPF problem. The results of  $N - 1$  contingency analysis for the base state indicate that there are 27 single line outage events that cause violations of one or more constraints with overall total of 37  $(x, c)$  event-overload pairs. In order to separate these contingencies we remove the corresponding  $(x, c)$  pairs from the original  $\mathcal{B}$  set after step 2 of the algorithm. In order to validate the pruning algorithm we have performed an exhaustive analysis of all possible 2 line contingencies and found 524 pairs of lines that result in overloads. Note, that this number is significantly less than the total number of  $N(N - 1)/2 \approx 5.3 * 10^6$  pairs and  $N(N - 1)(N - 2)/6 \approx 5.8 * 10^9$   $(x, y, c)$  triples that need to be analyzed with brute force approach.

ITERATION	$ \mathcal{A} $	$ \mathcal{B} $
0	5,341,546	10,683,092
1	17,928	322,365
2	6,128	188,761
3	5,816	163,788
4	5,750	156,807
5	5,750	155,813
6	5,750	155,813

Table I  
CANDIDATE SET  $\mathcal{A}$ ,  $\mathcal{B}$  SIZES EVOLUTION WITH ALGORITHM PROGRESSION.

Our algorithm has managed to reduce the number of  $(x, y)$  pair candidates from  $5.3 * 10^6$  to 6128 (that of course contain all 524 pairs that actually lead to overload) in only two steps. The subsequent outer loop iterations had marginal effect on the total number of pairs. Table I shows the evolution of the set  $\mathcal{A}$ ,  $\mathcal{B}$  sizes with each iteration. Note, that although there are a lot of elements in  $\mathcal{B}$  set, they don't affect the overall effectiveness of the approach, as the output of the

algorithm consists only of the initiating pairs  $(x, y)$  from the set  $\mathcal{A}$ . As one can see from the table, the algorithm converges after 6 iterations, but only the first two iterations lead to strong reductions in the  $\mathcal{A}$  set size, whereas the consequent iterations have diminishing returns.

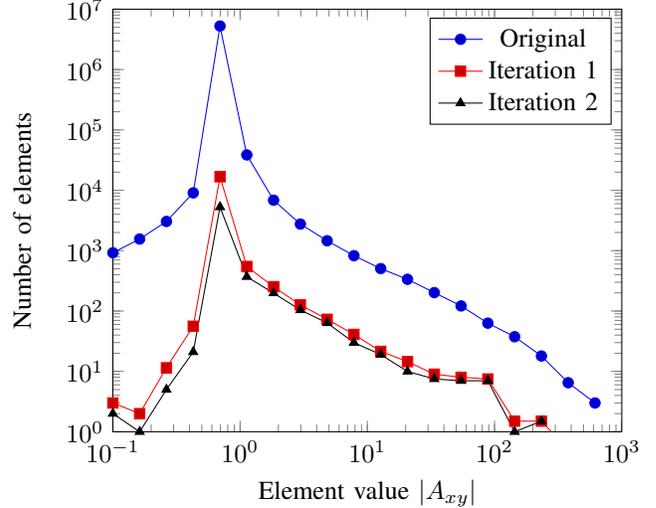


Figure 1. Histogram of  $A$  matrix elements distributions for the first two iterations.

In order to better understand the reason for the algorithm efficiency we have analyzed the distributions of the elements in the matrices  $A_{xy}$  and  $B_{xy}$ . As one can see from the figure 1 in the original system most of the elements of the matrix  $A$  are close to 1. This is because most of the lines do not affect each other after outages, so  $L_{xy}, L_{yx} \ll 1$ . Typically the flow from line  $x$  is distributed amongst its closest neighbors, whereas most of the lines  $y$  are not close in neither geographical nor electrical metrics. There are only about  $10^4$  pairs in the original network with value of  $A_{xy}$  larger than 1. As expected, the pruning operations have more significant effect on the left part of the distribution, as the corresponding pairs have lower chance of producing strong overflows. The third iteration of the algorithm has a seemingly minor effect on the distribution, but this is largely an artifact of the logarithmic scale of  $y$  axis, as the overall effect on the total number elements is quite significant as seen from the Table I.

The histogram 2 of the matrix  $B$  element has very different structure because the element  $B_{xc}$  is proportional to the line outage distribution factor  $L_{zx}$  that, as discussed previously, is very small for most of the pairs  $(x, z)$ . It is rather interesting that the distribution of  $B_{xz}$  and  $L_{xy}$  values (not shown) has an almost flat distribution in the log-scale, that points out to some self criticality in the network. We are not aware of any simple interpretations of this property. However, this property if shown to be universal for large scale power grids could be possibly linked to the power law distribution of

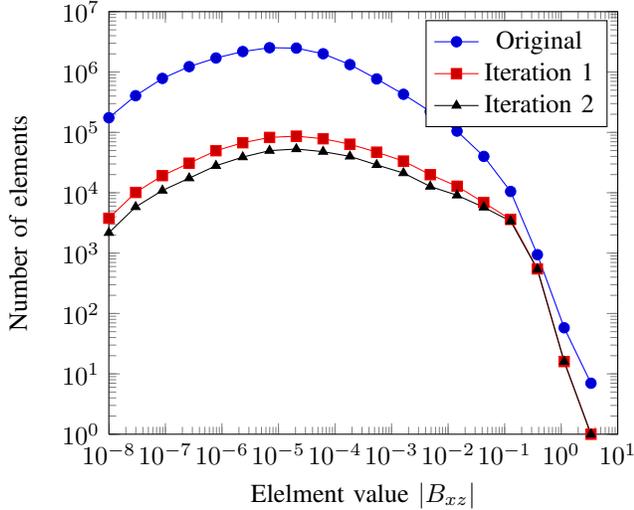


Figure 2. Histogram of  $B$  matrix elements distributions for first two algorithm iterations.

large blackout sizes [18–20] and potentially exploited for construction of fast contingency selection algorithms.

## 5. CONCLUSIONS

In conclusion, we have presented a novel algorithm for the  $N - 2$  contingency problem. The algorithm is based on the idea of iterative pruning of the possible candidate sets. Given the matrix of single line outage distribution factors only a small number of candidates can be identified in only  $O(N^2)$  operations, much smaller than the naive exhaustive search analysis that would require  $O(N^3)$  operations, therefore our algorithm decreases computational time by a factor of  $O(N)$  and its complexity is comparable with the complexity of usual  $N - 1$  contingency analysis. Unlike many other approaches our algorithm is not heuristic, and is certified to return all the double outage with violations. The algorithm has been validated and tested on the Polish grid example where the total number of double outage with violations was shown to be 524 via exhaustive search analysis. Our algorithm has reduced the set of all possible candidates from approximately 5000000 to about 6000 in just two iterations.

Although the effectiveness of the approach is impressive, there are several directions one can pursue to improve it even further. First, a number of additional optimizations are possible. Apart from the optimizations and implementation discussed briefly in the end of the section 3, there are a number of opportunities how this approach can be extended to more challenging settings. First, it is possible to apply the approach directly to  $N - k$  problems with  $k \geq 2$ . This would require accurate analysis of the expression (3) and derivation of relations similar to (4). Whenever only a small subset of possible  $k$ -line contingencies leads to violations, the proper bounding procedure should be able to filter out the safe

candidates. Another direction is associated with extension of our approach to AC power flows. As the approach is based on bounding various contributions to the line outage distribution factors, it might be feasible to extend to nonlinear systems without having to solve them in closed form. This is certainly a much more formidable task that necessitates a rather advanced nonlinear analysis approaches.

Another exciting opportunity lies in applying the proposed algorithm to the problem of analysis and mitigation of cascading failures in power grids [18, 21, 22]. The pruning approach can be used both for the development of efficient algorithms of assessing the probabilities of cascading outages, and for finding optimal decision choices for cascade prevention.

## ACKNOWLEDGEMENTS

This work was partially supported by NSF award ECCS - 1128437, MIT/SkTech seed funding grant and RGC (Russian government contract) 11.519.11.6018.

## REFERENCES

- [1] NERC, *Standard fac-011-2 – system operating limits methodology for the operations horizon*, 2009.
- [2] G. Ejebe and B. Wollenberg, “Automatic contingency selection,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-98, no. 1, pp. 97–109, Jan. 1979.
- [3] T. Mikolinnas and B. Wollenberg, “An Advanced Contingency Selection Algorithm,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, no. 2, pp. 608–617, Feb. 1981.
- [4] G. Irisarri and A. Sasson, “An Automatic Contingency Selection Method for On-Line Security Analysis,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, no. 4, pp. 1838–1844, Apr. 1981.
- [5] M. Enns, J. Quada, and B. Sackett, “Fast Linear Contingency Analysis,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-101, no. 4, pp. 783–791, Apr. 1982.
- [6] B. Stott, O. Alsac, and F. Alvarado, “Analytical and computational improvements in performance-index ranking algorithms for networks,” *International Journal of Electrical Power & Energy Systems*, vol. 7, no. 3, pp. 154–160, Jul. 1985.
- [7] G. Ejebe, H. Van Meeteren, B. Wollenberg, and H. P. V. Meeteren, “Fast contingency screening and evaluation for voltage security analysis,” *IEEE Transactions on Power Systems*, vol. 3, no. 4, pp. 1582–1590, 1988.
- [8] Q. Chen, S. Member, and J. J. D. McCalley, “Identifying High Risk N-k Contingencies for Online Security Assessment,” *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 823–834, May 2005.

- [9] T. Guler and G. Gross, "Detection of island formation and identification of causal factors under multiple line outages," *IEEE Transactions on Power Systems*, vol. 22, no. 2, pp. 505–513, May 2007.
- [10] R. D. Dosano, H. Song, and B. Lee, "Network centrality based N-k contingency scenario generation," in *2009 Transmission & Distribution Conference & Exposition: Asia and Pacific*, IEEE, Oct. 2009, pp. 1–4.
- [11] H. Mori and Y. Goto, "A tabu search based approach to (N-k) static contingency selection in power systems," in *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236)*, vol. 3, IEEE, 2001, pp. 1954–1959.
- [12] V. Donde, V. Lopez, B. Lesieutre, A. Pinar, and J. Meza, "Severe Multiple Contingency Screening in Electric Power Systems," *IEEE Transactions on Power Systems*, vol. 23, no. 2, pp. 406–417, May 2008.
- [13] M. J. M. Eppstein and P. D. H. Hines, "A Random Chemistry Algorithm for Identifying Multiple Contingencies that Initiate Cascading Failure," *Power Systems, IEEE Transactions on*, no. 99, 2011.
- [14] C. Davis and T. Overbye, "Linear Analysis of Multiple Outage Interaction," in *2009 42nd Hawaii International Conference on System Sciences*, IEEE, 2009, pp. 1–8.
- [15] C. M. Davis and T. J. Overbye, "Multiple Element Contingency Screening," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1294–1301, Aug. 2011.
- [16] T. Guler and G. Gross, "Detection of island formation and identification of causal factors under multiple line outages," *IEEE Transactions on Power Systems*, vol. 22, no. 2, pp. 505–513, May 2007.
- [17] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, *MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education*, 2011.
- [18] I. Dobson, B. a. Carreras, V. E. Lynch, and D. E. Newman, "Complex systems analysis of series of blackouts: cascading failure, critical points, and self-organization.," *Chaos (Woodbury, N.Y.)*, vol. 17, no. 2, p. 026 103, Jun. 2007.
- [19] P. Hines, K. Balasubramaniam, and E. Sanchez, "Cascading failures in power grids," *IEEE Potentials*, vol. 28, no. 5, pp. 24–30, Sep. 2009.
- [20] P. Hines, J. Apt, and S. Talukdar, "Large blackouts in North America: Historical trends and policy implications," *Energy Policy*, vol. 37, no. 12, pp. 5249–5259, Dec. 2009.
- [21] I. Dobson, J. Kim, and K. R. Wierzbicki, "Testing branching process estimators of cascading failure with data from a simulation of transmission line outages.," *Risk analysis : an official publication of the Society for Risk Analysis*, vol. 30, no. 4, pp. 650–62, Apr. 2010.
- [22] M. Koenig, P. Duggan, J. Wong, M. Y. Vaiman, M. M. Vaiman, and M. Povolotskiy, "Prevention of cascading outages in Con Edison's network," in *IEEE PES T&D 2010*, IEEE, 2010, pp. 1–7.