Fast and Reliable Screening of N-2 Contingencies

P. Kaplunovich and K. Turitsyn

***Abstract*—System Operators in many countries are required to maintain contingency plans for critical N-2 contingencies. Huge number of possible N-2 contingencies makes their direct assessment computationally prohibitive forcing the operators to rely on engineering judgement or uncontrollable heuristics. We present a novel algorithm for identification of critical N-2 contingencies that result in line overloads in post-contingency equilibrium. High computational efficiency of the algorithm is achieved via effective certification of safety for the majority of contingencies. Unlike many common heuristics, the algorithm is guaranteed to have zero missing rate in DC approximation models. Performance of the algorithm is validated by simulation of several IEEE case scenarios which demonstrate 30 – 1000 fold acceleration of contingency selection process in comparison with naïve brute-force approach. Various possible applications of the approach in the problems of security assessment, transmission topology control and planning are discussed in the end of the manuscript.**

***Index Terms*—Contingency screening, contingency analysis, linear sensitivities.**

I. INTRODUCTION

LARGE-SCALE power systems blackouts are some of the most catastrophic disasters in modern society that result in enormous economic damage of tens of billions per year for US economy alone [1]. To improve the reliability of power systems NERC and other regulatory agencies in the world enforce strict security standards that require power system operators to satisfy N-1 security constraint. Within N-1 constraint the system is required to continue normal operation after any single element failure. At the same time NERC requires the operators to maintain contingency plans [2] for all the critical N-2 contingencies where two elements fail within a short time-frame. List of critical N-2 contingencies and the corresponding contingency plans depend on the operating point and have to be updated on regular basis. So, as the levels of intermittent penetration increase the rate of contingency plans updates has to increase as well.

Selection of critical N-2 contingencies is a computationally challenging problem. “Brute force” enumeration of all possible component failure combinations and their effect on the system is realistically infeasible. Such analysis requires the operator to solve $N(N-1)/2$ instances of DC power flow. For a typical sized system model with 10000 lines the number of double line outages is almost $N_c = 50$ million. Even in DC approximation the number of operations required to re-solve equations for each simulation is at least $N_s = O(N^{1.2})$ [3]. So, the overall complexity of the “brute force” screening is at least $O(N_c N_s) = O(N^{3.2})$. Even in an optimistic scenario of optimized code capable of solving power flow equations in only 100 *ms* analysis of 50 million contingencies would take more than 1400 hours on a sequential computer. It would require a dedicated 1000 node cluster to reduce the analysis time to about an hour. Our study is motivated by the need in more efficient algorithms for analysis of N-2 contingencies.

The process of critical contingency list construction is called *contingency selection* and was originally introduced as a method of deciding which contingencies are important enough to be added to the list for online assessment [4]. In the initial work on the topic first-order performance index (PI) sensitivities were used to rank contingencies [4]. However, PI approach proved itself to be unreliable [5]. More effective approaches based on higher order performance indexes and DC power flow equations were developed in [6], [7]. In [7] Enns *et al.* were the first who noticed that one can significantly reduce the computational burden associated with contingency analysis by using *matrix inversion lemma* for small perturbations of the initial matrix to avoid inverting large matrices for each contingency. This research became a foundation for all following approaches built on the usage of *Line Outage Distribution Factors* (LODF) [8], [9]. Recent contingency screening and contingency analysis studies also include various techniques based on the network physical and electrical topology analysis [10], mixed integer and nonlinear optimization techniques [11]–[13] as well as statistical approaches like importance sampling [14]–[16] and other randomized algorithm based approaches [17], [18].

Despite the fact that plenty of algorithms for $N-k$ contingency screening were proposed in the last 3 decades, most of them are heuristic in a sense that they do not guarantee to find *all* critical contingencies. Lack of such guarantees introduces an additional risk factor to power systems operations and should be avoided in situations where security is critical. To address this problem we have developed [19] a fast and reliable algorithm for static N-2 contingency screening applicable for analysis of feasibility of post-contingency equilibrium in linearized power flow models. The algorithm reduces the time necessary to screen the contingency list by verifying that most contingencies are safe even without enumerating most of them. The ideology of the approach developed in this study is conceptually close to the bounding techniques introduced originally in the works of Galiana [20] and Brandwajn [21] and is inspired by the usage of LODFs for contingency screening reported in the recent works [9], [8].

The algorithm developed in this work is heuristic from a viewpoint of computational complexity. There are no mathematical theorems guaranteeing that it will be able to certify

Manuscript received January 06, 2015; revised May 20, 2015 and September 21, 2015; accepted November 16, 2015. Date of publication January 25, 2016; date of current version October 18, 2016. This work was supported in part by NSF and in part vt MIT/Skoltech initiative . Paper no. TPWRS-00026-2015.

K. Turitsyn is with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: turitsyn@mit.edu).

P. Kaplunovich resides in Cambridge, MA 02139 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRS.2015.2510586

safety for most of the contingencies thus resulting in a significant acceleration in comparison to the "brute force" approach. However, in all of our numerical experiments on realistic power systems we have observed that the algorithm can in fact improve the computational efficiency by a factor of $10^2 - 10^3$. We expect that this factor will be even higher for larger scale models. At the same time the key feature of our algorithm that distinguishes it from others is the mathematical guarantee of finding all critical contingencies, i.e., its *zero missing rate* . While the main limitation is its reliance on linearized power flow models. Hence, the algorithm is most appropriate for geographically small grids where thermal limits are dominating constraints and nonlinearities associated with angle differences or voltage drop can be ignored.

Major contributions of this work include a systematic description of the N-2 contingency selection algorithm and formal proof of its ability to identify all the critical contingencies. Several IEEE cases are used to validate the algorithm and assess its performance. The material is presented in the paper in the following order. After introducing in Section II the notations and formulating the problem in proper mathematical form, we continue in Section III by developing a general mathematical framework for contingency assessment in linearized power system models. The key algorithmic ideas of filtering the contingency set are presented in Section IV. The performance of algorithm is verified by simulations of several standard examples that are discussed in Section V. We conclude the paper by discussing applications and possible extensions of algorithm, mainly its extension to more realistic nonlinear models.

II. Model and Problem Formulation

In this study we consider a power grid with $N_B + 1$ buses and N_L branches. The topology of the grid is characterized by a directed graph $G = (V, E)$, $|V| = N_B + 1$, $|E| = N_L$, where each vertex $v_i \in V$ represents a bus and and each edge $e_{ij} = (v_i \to v_j) \in E$ represents a line connecting the bus v_i with the bus v_j. Following the notations from [9] we introduce the $N_L \times N_B$ incidence matrix $\boldsymbol{A}$ of the subgraph of G with the slack bus removed. Each row of matrix $\boldsymbol{A}$ has at most two non-zero elements: 1 on to the column corresponding to the "start" bus of line α and -1 on the column corresponding to the "end" bus of the line. We use the notation $\boldsymbol{a}_\alpha$ to refer to the row α of matrix $\mathbf{A}$. The reduced $N_B \times N_B$ nodal susceptance matrix is defined as $\boldsymbol{B} = \boldsymbol{A}^T \boldsymbol{X}^{-1} \boldsymbol{A}$, with $\boldsymbol{X}^{-1}$ being the diagonal $N_L \times N_L$ matrix of the branch susceptances, $\boldsymbol{X}^{-1} = \operatorname{diag}(b_1, \ldots, b_{N_L})$.

The algorithm developed in this manuscript is based on linearized models of power flow equations. The most important of those is the well-known DC approximation of the power flow equations that we use to explain the algorithm. However, the techniques can be straightforwardly extended to other linearized equations that may include both reactive power and voltage variations. Linearized models of power flow equations are commonly used for computationally intensive tasks, such as contingency screening, unit commitment and other problems. Although these models are not always accurate they may provide a reasonable approximation for the grids where the flows are mainly constrained by thermal limits, rather than angle differences or voltage limits.

Within the DC approximation the state of the system is characterized by the $N_B \times 1$ vector of bus voltage angles $\boldsymbol{\theta}$. The DC power flow equations represent the relation between the angle vector and $N_B \times 1$ power injections vector $\mathbf{p}$:

$$\mathbf{B}\boldsymbol{\theta} = \mathbf{p}. \tag{1}$$

The $N_L \times 1$ branch flow vector representing the power flowing over every line of the system can be represented as

$$\mathbf{f} = \mathbf{X}^{-1}\mathbf{A}\boldsymbol{\theta} = \mathbf{X}^{-1}\mathbf{A}\mathbf{B}^{-1}\mathbf{p}. \tag{2}$$

The feasibility set is described by thermal limits on power flows formally defined via the following inequalities:

$$\underline{f_\alpha} < f_\alpha < \overline{f_\alpha}. \tag{3}$$

Note, that although in this paper we consider only line overload constraints, the algorithm can be straightforwardly extended to constraints defined by arbitrary linear systems 3 that may include both the voltage and angle constraints in case of more general linearized power flow models.

We consider contingencies corresponding to single and double line outages, and focus only on the analysis of the post-contingency equilibrium, assuming redistribution of powers as described by the equation (1). The generator and load outages are not considered in this work, although the generalization of the algorithm is straightforward as reported by the authors in [22]. As our algorithm is based on the idea of iterative safety certification of large number possible scenarios, it is convenient to describe the process via manipulation of the sets of critical contingency candidates.

First, we define the set $\mathcal{C}_1$ of single-element outage (N-1) contingencies corresponding to configurations of the grid with only one line removed. The size of this set is equal to the number of lines N_L. Similarly, we define the set of double-outage contingencies $\mathcal{C}_2 = \{(\alpha, \beta) : \alpha < \beta; \alpha, \beta \in 1 \ldots N_L, \}$, which formally represents the list of line labeled by indices α, β. The condition $\alpha < \beta$ ensures that each pair is present only once, and that the two lines in a pair are distinct. The total number of $N - 2$ contingencies is thus $N_L(N_L - 1)/2$. We use the term *islanding contingency* to refer to the subset of $\mathcal{C}_2$ corresponding to the configurations where the post-contingency power grid graph has multiple connected components. Each element in this subset corresponds to a pair of lines which form a cutset of the graph. Usually islanding of power system results in some load shedding, so contingency plans have to be established for each of these events. The non-islanding contingencies that result in the post-contingency state with at least one line being overloaded are referred to as *critical contingencies* . The objective of the N-2 contingency selection algorithm described in this work is to identify the full set of all islanding and critical contingencies in a computationally efficient way.

For the rest of the paper we assume that the base operating point described by the injection vector $\boldsymbol{p}^0$, and line flow vector $\boldsymbol{f}^0$ is feasible, and moreover N-1 secure, so removal of any single do not result in any islanding or overloads in the post-contingency equilibrium.

III. Post-Contingency Equilibrium

We start this section by reviewing the theoretical foundation behind the line outage distribution factors using the notation similar to the ones used in [9]. The reader may also consult [23], [24] for more detailed derivations of similar expressions.

Tripping of some set of lines changes the topology of the grid and its susceptance matrix. In the case of single line α outage the post-contingency susceptance matrix denoted as B^α for the grid can be represented as:

$$\mathbf{B}^\alpha = \mathbf{B} - \mathbf{a}_\alpha^\top b_\alpha \mathbf{a}_\alpha. \tag{4}$$

Here and in the following we use the upper indices to denote the contingency, whereas the lower indices enumerate the vector/matrix elements.

Whenever the line α is outaged, the power f_α^0 that was flowing through it in pre-contingency state is distributed among all the other lines in the system. The distribution of power flow results in a change of power flowing through line γ from the original f_γ^0 to the post-contingency one $f_\gamma^0 + \Delta f_\gamma^\alpha$. The expression for the distribution matrix Δf_γ^α can be easily derived using the well-known *matrix inversion lemma*:

$$\Delta f_\gamma^\alpha = d_\gamma^\alpha f_\alpha^0 \tag{5}$$

$$d_\gamma^\alpha = \left(1 - b_\alpha \mathbf{a}_\alpha \mathbf{B}^{-1} \mathbf{a}_\alpha^\top\right)^{-1} b_\gamma \left[\mathbf{A}\mathbf{B}^{-1}\mathbf{a}_\alpha^\top\right]_\gamma. \tag{6}$$

Matrix d_γ^α defined in (6) is the well-known matrix of Line Outage Distribution Factors. It plays an important role in the contingency selection studies, and represents linear sensitivities of the post-contingency line flows to the pre-contingency flow on the outaged line α [23].

This expression for the branch flow changes is not valid in situations when the denominator goes to zero, so $b_\alpha \mathbf{a}_\alpha \mathbf{B}^{-1} \mathbf{a}_\alpha^\top = 1$. These situations correspond to islanding of post-contingency grid. The post-contingency equilibrium analysis of islanded grids requires additional assumptions on generator primary and secondary response as well as load-shedding policies and is not discussed in this manuscript, although this problem was addressed by the authors in [22].

Multiple line outages can be analyzed in a similar manner. After an outage of the lines α and β the post-contingency susceptance matrix $\mathbf{B}^{\alpha\beta}$ is given by

$$\mathbf{B}^{\alpha\beta} = \mathbf{B} - \mathbf{a}_\alpha^\top b_\alpha \mathbf{a}_\alpha - \mathbf{a}_\beta^\top b_\beta \mathbf{a}_\beta. \tag{7}$$

The vector of the load flow distributions $\Delta f^{\alpha\beta}$ can be recovered using 2 and matrix inversion lemma resulting in the expression:

$$d_\gamma^{\alpha\beta} = \left[d_\gamma^\alpha \, d_\gamma^\beta\right] \mathbf{D}^{\alpha\beta} \begin{bmatrix} f_\alpha^0 \\ f_\beta^0 \end{bmatrix} \tag{8}$$

$$\mathbf{D}^{\alpha\beta} = \frac{1}{1 - d_\beta^\alpha d_\alpha^\beta} \begin{bmatrix} 1 & d_\beta^\alpha \\ d_\alpha^\beta & 1 \end{bmatrix}. \tag{9}$$

The equation (8) lies in the foundation of our algorithm. It provides a mathematically simple representation of double line outage effect in terms of single line outage distribution factors d_γ^α. So, the single line outage distribution factors that are calculated during the $N-1$ contingency analysis can be reused in $N-2$ analysis without any computational overhead.

As in the case of single-line contingencies, the special situation $d_\beta^\alpha d_\alpha^\beta = 1$ corresponds to islanded post-contingency grid.

The feasibility condition (3) for $N-2$ non-islanding contingencies can be expressed as a combination of the following inequalities

$$\left[d_\gamma^\alpha \, d_\gamma^\beta\right] \mathbf{D}^{\alpha\beta} \begin{bmatrix} f_\alpha^0 \\ f_\beta^0 \end{bmatrix} < \overline{f_\gamma} - f_\gamma^0, \tag{10}$$

$$\left[d_\gamma^\alpha \, d_\gamma^\beta\right] \mathbf{D}^{\alpha\beta} \begin{bmatrix} f_\alpha^0 \\ f_\beta^0 \end{bmatrix} > f_\gamma^0 - \underline{f_\gamma}. \tag{11}$$

The set of the overload constraints S_1 defined by the upper and lower limits for line flows consists of $2N_L$ elements. For the simplicity of the further analysis, it is convenient to aggregate all the constraints in a single vector and rewrite the system of constraints in the following form:

$$\xi_\gamma^\alpha \Gamma_{\alpha\beta} + \xi_\gamma^\beta \Gamma_{\beta\alpha} < 1 \tag{12}$$

$$\Gamma_{\alpha\beta} = \frac{1 + \frac{d_\beta^\alpha f_\beta^0}{f_\alpha^0}}{1 - d_\beta^\alpha d_\alpha^\beta}. \tag{13}$$

Where the $2N_L \times 1$ row vector $\boldsymbol{\xi}^\alpha = [\overline{\boldsymbol{\xi}^\alpha}, \underline{\boldsymbol{\xi}^\alpha}]$ composed of two components associated with the bounds on the flow defined in (10) and (11):

$$\overline{\xi_\gamma^\alpha} = \left(\overline{f_\gamma} - f_\gamma^0\right)^{-1} d_\gamma^\alpha f_\alpha^0 \tag{14}$$

$$\underline{\xi_\gamma^\alpha} = \left(f_\gamma^0 - \underline{f_\gamma}\right)^{-1} d_\gamma^\alpha f_\alpha^0. \tag{15}$$

The elements of vectors $\overline{\boldsymbol{\xi}^\alpha}, \underline{\boldsymbol{\xi}^\alpha}$ have a very natural interpretation. For the single line α outage contingency, the component $\overline{\xi_\gamma^\alpha}$ shows how close is the post-contingency state to violation of the upper-bound constraint on the flow through line γ. Whenever $\overline{\xi_\gamma^\alpha} > 1$ the post-contingency state violates the constraint upperbound. Naturally, whenever $\underline{\xi_\gamma^\alpha} > 1$ the lower bound limit on line γ is violated after the outage of line α. Similarly to line outage distribution factors d_γ^α the terms ξ_γ^α could be called *Line Outage Overload Factors* (LOOF).

Similarly, the elements $\Gamma_{\alpha\beta}$ express the amount of "interference" between the single line outages in double outage contingency. Whenever line α is outaged some of its power gets redistributed to line β which increases or decreases the effect of its outage in comparison to single line outage of line β. Hence the effect of single line β outage on the loading of line γ expressed by the term ξ_γ^β is amplified by a factor of $\Gamma_{\beta\alpha}$ in (12). Whenever the lines α and β are far away from each other, so that the LODFs d_β^α are small enough, the factors $\Gamma_{\alpha\beta}, \Gamma_{\beta\alpha}$ become approximately one, and the expression (12) becomes a superposition of single line outage effects. Hence, we propose to call the factors $\Gamma_{\alpha\beta}$ the *Line Outage Interference Factors* (LOIF).

As discussed before, the total number of double line outages contingencies that need to be considered is equal to $N_L(N_L-1)/2$. For every contingency there are $2(N_L-2)$ constraints that need to be checked corresponding to overloads of all the lines except for the outaged ones. Before introducing the accelerated algorithm, we explain the naïve brute-force algorithm that can be used to screen contingencies.

1) Calculate the matrices $\overline{\xi_\gamma^\alpha}$ and $\underline{\xi_\gamma^\alpha}$ of single line outage overload factors. Total N_L operations involving solution of (1).
2) Calculate the matrix of line outage interference factors $\Gamma_{\alpha\beta}$. Total $N_L(N_L-1)/2$ basic operations.
3) For every triple α, β, γ check the condition (12). Designate triples that violate (12) as critical. Total $N_L(N_L-1)(N_L-2)$ operations.

In this work we assume that the single-line outage distribution factors calculated on step 1 are already available from the N-1 security analysis. Otherwise, the complexity of this step depends on the fill-in factor of the LU decomposition of matrix

B which tend to be rather low for realistic power systems. The total complexity of step 1 scales somewhere between N_L^2 and N_L^3 depending on the fill-in factor. In practice, for the models available in MATPOWER we observed that the computation time of the "brute-force" approach is dominated by step 3 with $N_L(N_L-1)(N_L-2)$ operations. The algorithm described in the next section allows to greatly accelerate the screening process by exploiting the special nature of the overloading conditions described by (12).

IV. Algorithm

The combinatorial complexity of direct enumeration and verification of all contingency-constraint combinations can be avoided with the help of filtering process. In our work this filtering process is organized via iterative construction and application of the so-called *safety certificates* . The concept of *safety certificate* refers to a computationally tractable mathematical condition that guarantees that some combinations of contingency (α, β) and constraint γ are safe, or in other words satisfy the inequality (12). The key idea of the iterative screening approach is to design safety certificates that one hand allow to screen out as large set of safe conditions as possible while on other being as computationally cheap as possible. The iterative nature of the algorithm relies on the idea that whenever some triples α, β, γ are certified to be safe and eliminated from the critical contingency candidate set, new better safety certificates can be constructed that can certify safety of even more critical contingency candidates.

The authors are not aware of any published applications of this approach to problems arising in power system context. However, it is one of the well-established and commonly used meta-heuristic in constraint programming [25] and more broadly combinatorial optimization problems. It is also conceptually similar to the widely used branch and bound technique in optimization, although the problem we consider is not of optimization nature, as we require to identify all the critical contingencies, not only the optimal ones. This meta-heuristic is well known to general audience familiar with Sudoku puzzles. The usual strategy of solving these puzzles is based on iterative elimination of puzzle solution candidates. The row, column and cell constraints serve as analogues of "safety certificates" that allow to certify that certain combinations of number positions do not satisfy the constraint and thus cannot be a valid solution and therefore are guaranteed to be "safe". After every elimination step the size of the set of the solution candidates is reduced quite dramatically, and new constraints can be applied to reduce it even further until a single solution is found. In this work we exploit the fact that the contingency selection problem is conceptually similar to the Sudoku puzzle, but instead of row, column and cell constraints the relevant constraints are represented by the inequalities (12). Critical contingencies are characterized by violation of at least one of the $N_L(N_L-1)(N_L-2)$ inequalities of the form (12).

The abstract flow of the algorithm is most naturally expressed using the language of sets. Formally, we define a set of all contingency – constraint triples $(\alpha, \beta, \gamma) \in \mathcal{D}^t$ that are not yet certified to be safe at algorithm step number t. This set can be visualized as three-dimensional grid of cells. Empty cells correspond to triples that are provably safe, whereas full cells correspond to potential contingency candidates. At the initial stage $t = 0$ of the algorithm, before any filtering took place, this set is simply a Cartesian product of the set of all possible double contingencies and all possible constraints: $\mathcal{D}^0 = \mathcal{C}_2 \times \mathcal{S}_1$. In other words, it consists of all combinations (α, β, γ) with $\alpha, \beta = 1 \ldots N_L$, with $\alpha < \beta$ and $\gamma = 1 \ldots 2N_L$. The three dimensional visualization of the set is completely full, i.e., all cells are full.

As the construction of this set in computer memory would require at least N_L^3 operations, practically this set is never constructed, and instead the algorithm relies on analysis of its "projections". A projection is simply a two-dimensional view of the three-dimensional cell grid. If at least one cell along the one-dimensional lines of sight is full, so is the cell in the projection. The empty cells in the projection correspond to transparent one-dimensional lines of sight with no full cells along each line.

Two important projections of this set are relevant in the context of our algorithm: one is the projection on the set of contingencies: $\mathcal{C}_2^t = \{(\alpha, \beta) | \exists\gamma : (\alpha, \beta, \gamma) \in \mathcal{D}^t\}$, and another on the outaged-line contingency pairs: $\mathcal{S}_2^t = \{(\alpha, \gamma) | \exists\beta : (\alpha, \beta, \gamma) \in \mathcal{D}^t\}$. These projections have a very simple interpretation. Whenever there exists at least one constraint γ that is not certified to be safe for the double outage of lines α, β, the element (α, β) belongs to the projection $\mathcal{C}_2^t$. In other words, whenever the pair (α, β) belongs to the set $\mathcal{C}_2^t$ the contingency (α, β) cannot be ruled out as non-critical on step t. Similarly, whenever a combination of at least one outaged line β with another line α is not certified to be safe with respect to violation of constraint γ the pair (α, γ) belongs to the set $\mathcal{S}_2^t$.

The sets $\mathcal{C}_2$ and $\mathcal{S}_2$ are geometrical projections of the full $\mathcal{D}^t$ set cell visualization on the (α, β) and (α, γ) planes.

These projections are crucial for our algorithm because they allow to filter the set $\mathcal{D}^t$ without direct enumeration of all its components. Whenever, for example, some pair $(\alpha^*, \beta^*) \in \mathcal{C}_2^t$ is certified to be safe, all triples $(\alpha^*, \beta^*, \gamma)$ with all possible values of γ can be safely removed from the set $\mathcal{D}^t$. So the key to efficient filtering is the ability to construct safety certificates $\mathrm{Cert}(\alpha^*, \beta^*)$ that can be checked without direct enumeration over a set of elements $(\alpha^*, \beta^*, \gamma)$ with varying values of γ.

We construct this certificate by bounding the terms by exploiting the separable nature of the safety criterion (12) derived above. Formally, we define the quantity

$$U_{\alpha\beta}^t = \max_{\gamma \in \mathcal{Z}_{\alpha\beta}^t} [\xi_\gamma^\alpha \Gamma_{\alpha\beta} + \xi_\gamma^\beta \Gamma_{\beta\alpha}] \tag{16}$$

where the maximization takes place over all possible values of γ that are present in the projected set $\mathcal{S}_2^t$, so formally

$$\mathcal{Z}_{\alpha\beta}^t = \{\gamma \,|\, \exists\gamma : (\alpha, \gamma) \in \mathcal{S}_2^t \text{ or } (\beta, \gamma) \in \mathcal{S}_2^t\}. \tag{17}$$

The separable algebraic structure of the condition (12) ensures that all the values of $U_{\alpha\beta}^t$ can be evaluated in only $O(N_L^2)$ operations. Indeed this can be accomplished by precomputing all the values of $\max_\gamma \xi_\gamma^\alpha$ which would take $O(N_L^2)$ operations, and then calculating all $U_{\alpha\beta}^t$ in another $O(N_L^2)$ operations. The condition $U_{\alpha\beta}^t < 1$ is a safety certificate that guarantees that all of yet uncertified critical contingency candidates of the type (α, β, γ) are indeed safe. So, this condition can be used to filter the set of $\mathcal{C}_2^t$ of all possible outage line pairs:

$$\mathcal{C}_2^{t+1} = \{(\alpha, \beta) \in \mathcal{C}_2^t : \, U_{\alpha\beta}^t > 1\}. \tag{18}$$

Physically this condition can be interpreted as worst case scenario analysis. For a given contingency (α, β) instead of going through all possible constraints on post-contingency equilibrium we look only at the worst case scenario among the non-certified events. These scenarios corresponds to two pairs α, γ and β, γ with the highest values of LOOF. If the worst case scenario is certified to be safe, it automatically guarantees that all the non-worst case candidates are safe.

Completely analogous procedure can be constructed for filtering the projection set $\mathcal{S}_2^t$. The corresponding expressions are presented below:

$$V_{\alpha\gamma}^t = \max_{\beta \in \mathcal{B}_{\alpha\gamma}^t} [\xi_\gamma^\alpha \Gamma_{\alpha\beta} + \xi_\gamma^\beta \Gamma_{\beta\alpha}] \tag{19}$$

$$\mathcal{B}_{\alpha\gamma}^t = \{\beta \,|\, \exists \beta : (\alpha, \beta) \in \mathcal{C}_2^t\} \tag{20}$$

$$\mathcal{S}_2^{t+1} = \{(\alpha, \gamma) \in \mathcal{S}_2^t : \, V_{\alpha\gamma}^t > 1\}. \tag{21}$$

The filtering iterations reduce the set sizes on every step, so formally $\mathcal{C}_2^{t+1} \subseteq \mathcal{C}_2^t$ and $\mathcal{S}_2^{t+1} \subseteq \mathcal{S}_2^t$. This in turn reduces the set of all non-certified contingencies that can be reconstructed as $\mathcal{D}^t = \{(\alpha, \beta, \gamma) : \, (\alpha, \beta) \in \mathcal{C}_2^t, (\alpha, \gamma) \in \mathcal{S}_2^t\}$. Besides, as follows from (19) and (20) the filter for the set $\mathcal{S}_2^t$ depends on the composition of set $\mathcal{C}_2^t$, so the iterations may continue for several steps before the algorithm converges.

Assuming that the algorithm converges or is terminated after T steps the overall complexity can be estimated as $O(TN_L^2 + |\mathcal{D}^T|)$, where the last term accounts for the complexity of direct processing of all the non-certified contingencies. Whenever the total number of the critical contingencies is small enough one may hope that the iterative filtering procedure will significantly reduce the overall computational burden by filtering most of the safe candidates. Our numerical simulations presented in the sections below show that this indeed the case.

The full algorithm is summarized on the inset and described in plain words below. In addition to the filtering steps we also formally show how the initial conditions for the sets $\mathcal{C}_2^t$ and $\mathcal{S}_2^t$ are introduced. The expressions in the lines 0 and 2 as well as line 11 explain how our implementation deals with islanding $N - 1$ and $N - 2$ contingencies that appear in the actual IEEE models. As discussed in the previous section, the islanding contingencies result in singular single and double-line outage distribution factors. We separate them initially in the set $\mathcal{C}_1^{isl}$ of islanding $N - 1$ contingencies and set $\mathcal{C}_2^{isl}$ of islanding $N - 2$ contingencies. These sets are identified by checking the conditions $b_\alpha \mathbf{a}_\alpha \mathbf{B}^{-1} \mathbf{a}_\alpha^\top = 1$ and $d_\beta^\alpha d_\alpha^\beta = 1$ respectively. Both of the conditions correspond to zero denominator. Due to reliance on finite accuracy floating number arithmetic neither of the conditions is satisfied exactly, so numerical tolerance of 10^{-8} is used to check both of the conditions. On steps 1 and 2 we formally separate the set of islanding N-2 contingencies that is composed of all combination of N-1 islanding outages with all the other lines, as well as N-2 islanding contingencies described by the condition $d_\beta^\alpha d_\alpha^\beta = 1$. The sign $\times$ describes the cartesian product of the sets, while the signs $\setminus$ and $\cup$ refer to set subtraction and addition respectively.

The plain English interpretation of the steps of the algorithms described in the inset is presented below.

Lines 1 and 2. Start with the full candidate set consisting of all possible tripped line pairs and all lines being candidates for overload. Exclude the topological $N - 1$ and $N - 2$ "islanding" contingencies from the candidate set following the procedures from previous paragraph.

Line 4. Calculate the maximal LOOF factors $\max_\gamma \xi_\gamma^\alpha$ for a given N-1 contingency in accordance to equation (17). Use these values to determine the values of matrix U in equation (16). This matrix determines the upper bound (worst-case) impact of N-2 contingency corresponding to tripping of lines α and β.

Line 5. Matrix U^t from previous step used to filter the set C^t of possible initiating pairs. If the worst-case line is not overloaded is safe, so are all the other lines.

Line 6. Similar to step 4. Calculate $\max_\beta \xi_\gamma^\beta$ and $\max_\beta \Gamma_{\beta\alpha}$ corresponding to upper bound of the overloading impact of a given line γ.

Line 7. Use (19) and (21) to filter safe tripped lines from the contingency candidate set. If a given line cannot be overloaded in the worst case it should not be considered in the contingency analysis.

Line 8. Repeat the iterative candidate set pruning until it stops to decrease in size or the maximal iteration count is reached.

Line 10. Identify the truly critical contingencies by "brute-force" analysis of the filtered candidate set.

Line 11. Add the islanding N-1 and N-2 contingencies to the full list of critical contingencies.

Algorithm 1 $\mathtt{N} - 2$ contingency selection

1: $\mathcal{C}_2^1 \leftarrow \mathcal{C}_2 \setminus [\mathcal{C}_2^{isl} \cup (\mathcal{C}_1^{isl} \times \mathcal{C}_1)]$
2: $S_2^1 \leftarrow S_2 \setminus [\mathcal{C}_1^{isl} \times S_1]$
3: **repeat**
4: Find $U_{\alpha\beta}^t$ using Eq. (16)
5: Filter $\mathcal{C}_2^t$ using Eq. (18)
6: Find $V_{\alpha\beta}^t$ using Eq. (19)
7: Filter $\mathcal{S}_2^t$ using Eq. (21)
8: **until** $\mathcal{C}_2^t = \mathcal{C}_2^{t-1}$, $\mathcal{S}_2^t = \mathcal{S}_2^{t-1}$ or $t = t_{\max}$
9: $T \leftarrow t$
10: Filter $\mathcal{C}_2^T$ directly: $\mathcal{C}_2^T \rightarrow \mathcal{C}_2^{final}$
11: $\mathcal{C}_2^{critical} \leftarrow \mathcal{C}_2^{final} \cup [\mathcal{C}_2^{isl} \cup (\mathcal{C}_1^{isl} \times \mathcal{C}_1)]$
12: **return** $\mathcal{C}_2^{critical}$

We have published the open source MATLAB implementation of the algorithm at [26]. The implementation of the algorithm relies on the data sets and processing routines from the MATPOWER package [27] and can process all the cases available in the package.

V. Simulation Results

We have tested the performance of the proposed algorithm on all of the largest publicly available power grid models that are distributed with the MATPOWER package for MATLAB [27]. The testing procedure was organized as described below.

For every of the power grid model, we initialized the state with N-1 secure configuration obtained via solving the DC optimal power flow procedure (the details of the procedure can be found in [28]).Then the models were preprocessed to a form acceptable to our algorithm implementation. All the generators on

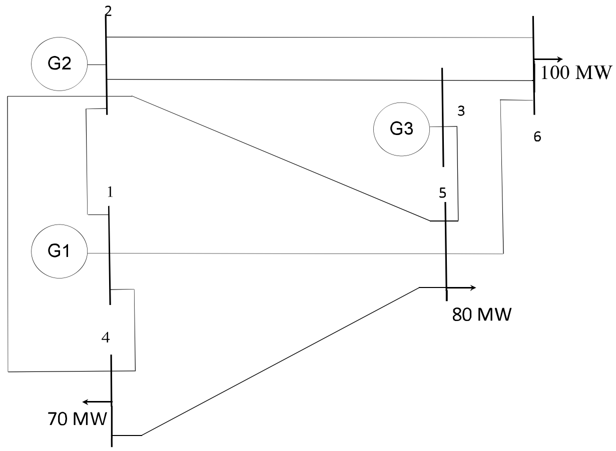

Fig. 1. 6 bus system from [23].

TABLE I
LINE DATA FOR 6 BUS SYSTEM

α	from	to	f_α	$\overline{f_\alpha}$	$\underline{f_\alpha}$
1	1	2	2.6	40	−40
2	1	4	26	60	−60
3	1	5	21.3	40	−40
4	2	3	−0.17	40	−40
5	2	4	46.9	60	−60
6	2	5	19.6	30	−30
7	2	6	24.3	90	−90
8	3	5	22.7	70	−70
9	3	6	49	80	−80
10	4	5	3	20	−20
11	5	6	−3.4	40	−40

the same buses were aggregated into a single generator and all parallel power lines were replaced with a single line of equivalent susceptance. The latter preprocessing step is not necessary, and was introduced historically to simplify the representation of the sets in MATLAB. It will be removed in the future versions of the algorithm. Then, all the single lined outage distribution factors defined by (5) were calculated and all the N-1 islanding contingencies were identified. The updated model and the information about LODF and islanding contingencies were passed to the N-2 contingency selection algorithm.

A. Illustrative 6-Bus Example

Before proceeding to the discussion of large-scale cases we provide an illustration of the algorithm and various matrices employed in it on a classical 6-bus example from [23] available in MATPOWER package. This system model illustrated on the Fig. 1 consist of 3 generators and 3 loads connected in total by 11 lines.

The detailed parameters of the system used in our simulation can be found in MATPOWER 4.0 *case6ww* datafile. On the Table I we show the key parameters of the lines. Table II shows the line outage distribution factors d_β^α. For example, from the first column one can see that whenever line 1 is outaged all the power going from bus 1 through this line is distributed among lines 2 and 3 also going from bus 1 in proportions of 59% and 41% respectively. One can easily notice that no element or pair of elements satisfies the islanding conditions, which is consistent with the topology of the grid illustrated on Fig. 1.

The matrix of line outage interference values $\Gamma_{\alpha\beta}$ is presented on Table III. One can immediately notice the anomalously high levels in row 4 associated with the small base flow $f_4^0 \approx -0.17$ through this line. Similarly the line outage overload matrix ξ_γ^α is presented on Table IV. Notably, all the elements in the Table IV have magnitude less than one, so the system is N-1 secure. The highest element is $\xi_2^5 \approx 0.85$ which corresponds to failure of line 2 connecting the buses 2 and 4. In pre-contingency scenario it is the highly loaded line transferring 49 *MW* of power. After the failure about 61% of the power is distributed on line 6 which becomes almost overloaded.

On the first step of algorithm the matrix $U_{\alpha\beta}^1$ is calculated and found to be equal to the one presented on Table V. Note, that quite a lot of values on this table have absolute values less than 1. This allows the algorithm to filter out a lot of contingencies. For example, the element is equal to $U_{17}^1 \approx 0.51$. This means that whenever the pair of lines 1 and 7 fail no other line can be overloaded. The algorithm was able to prove this fact without actually explicitly solving the power flow equations for post-contingency equilibrium with failed lines 1 and 7. In other words, single calculation of the element U_{17}^1 allowed to certify safety for 22 constraints associated with the post-contingency equilibrium. Ability to certify safety without actual consideration of all the constraints is what makes the algorithm so efficient on large power system cases. On the next step the algorithm performs a similar procedure using the matrix $V_{\alpha\gamma}^t$ not presented in the paper. The iterations converge after one step, and the final number of non-certified contingencies is found to be 23. For this simple system the reduction in number of candidates is not that impressive, however the effectiveness of the algorithm grows with system size as illustrated on examples below.

B. Summer and Winter Off-Peak Cases

The first set of simulations was performed using the Summer and Winter off-peak Polish power grid cases *case2737sop, case2746wop* from the MATPOWER package.

The Summer grid case consists of 2737 buses, 399 generators and 3506 lines, which suggests the size of the set $\mathcal{C}_2$ to be $(N_L - 1)N_L/2 \sim 6 * 10^6$. The algorithm reduced the size of the non-islanding double outage contingencies set from $|\mathcal{C}_2^1| = 3.5 * 10^6$ to $|\mathcal{C}_2^3| = 5346$ in just 2 iterations. The Table VI shows how the sizes of the candidate sets $\mathcal{C}_2^k$ and S_2^k changed with each iteration. The algorithm converged in 6 iterations, but the main reduction happened in the first two steps.

The simulations performed on the Winter off-peak case have shown very similar convergence results (Table VII). Compared to the Summer case, the Winter case has a higher number of elements and is slightly more stressed, which results in the larger number of critical non-islanding N-2 contingencies $|\mathcal{C}_2^{final}| = 928$ and slightly less effective performance of the filtering loop ($|\mathcal{C}_2^T|/|\mathcal{C}_2^{final}| \sim 30$).

This simulation shows the high efficiency of the algorithm is non-stressed conditions. In just two iterations the algorithm was able to filter out 99.9% and 99.1% of double outage non-islanding contingencies for the Summer and Winter cases correspondingly.

C. Winter Peak Case

The simulation results for the Winter peak Polish power grid case (*case2383wp* in MATPOWER package) are presented in the Table VIII. This peak case is much more loaded than the off-

TABLE II
LINE OUTAGE DISTRIBUTION FACTORS MATRIX d_β^α FOR 6 BUS SYSTEM

Lines	1	2	3	4	5	6	7	8	9	10	11
1	−1	0.64	0.54	−0.11	−0.5	−0.21	−0.12	−0.14	0.01	0.01	0.13
2	0.59	−1	0.46	−0.03	0.61	−0.06	−0.04	−0.04	0	−0.33	0.04
3	0.41	0.36	−1	0.15	−0.11	0.27	0.16	0.18	−0.02	0.32	−0.17
4	−0.1	−0.03	0.18	−1	0.12	0.23	0.47	−0.4	−0.53	0.17	0.13
5	−0.59	0.76	−0.17	0.16	−1	0.3	0.17	0.19	−0.02	−0.67	−0.19
6	−0.19	−0.06	0.33	0.22	0.23	−1	0.24	0.27	−0.03	0.31	−0.26
7	−0.12	−0.04	0.21	0.51	0.15	0.27	−1	−0.2	0.58	0.2	0.44
8	−0.12	−0.04	0.2	−0.38	0.14	0.26	−0.17	−1	0.47	0.19	−0.42
9	0.01	0	−0.03	−0.62	−0.02	−0.03	0.64	0.6	−1	−0.02	0.56
10	0.01	−0.24	0.29	0.13	−0.39	0.24	0.14	0.15	−0.02	−1	−0.15
11	0.11	0.03	−0.18	0.12	−0.13	−0.23	0.36	−0.4	0.42	−0.18	−1

TABLE III
LINE OUTAGE INTERFERENCE FACTORS MATRIX $\Gamma_{\alpha\beta}$ FOR 6 BUS SYSTEM

Lines	1	2	3	4	5	6	7	8	9	10	11
1	0	11.81	6.97	1.02	−11.43	−0.6	−0.14	−0.2	1.25	1.01	0.84
2	1.7	0	1.65	1	3.95	0.96	0.97	0.97	1.01	1.04	1
3	1.35	1.73	0	1.03	0.77	1.37	1.22	1.23	0.96	1.15	1.06
4	2.86	6.75	−25.58	0	−39.47	−30.73	−99.95	74.04	262.76	−2.5	4.09
5	1.37	2.68	0.94	1.02	0	1.2	1.12	1.12	0.98	1.3	1.04
6	1.02	0.93	1.49	1.05	1.65	0	1.39	1.41	0.93	1.13	1.11
7	1	0.96	1.22	1.31	1.32	1.3	0	0.84	3.47	1.05	1.12
8	1	0.96	1.24	1.18	1.33	1.31	0.84	0	2.83	1.06	1.28
9	1	1	0.99	1.49	0.98	0.99	2.1	1.79	0	1	1.25
10	1.01	−1.16	3.37	1.02	−6.95	2.75	2.18	2.24	0.75	0	1.2
11	0.93	0.74	2.24	1.02	2.86	2.52	−1.92	4.47	−6.58	1.19	0

peak cases studied above. As a consequence the total number of critical non-islanding double outage contingencies is very high: $|\mathcal{C}_2^{final}| = 15882$. The pruning loop of the algorithm was able to filter out about 89% of candidates in two iterations yielding the output set of the size $|\mathcal{C}_2^T| = 267681$. The overall efficiency of the filtering measured as the ratio of false to real candidates $|\mathcal{C}_2^T|/|\mathcal{C}_2^{final}|$ is equal to ~ 17 which is even better than for the Winter off-peak case.

The summary of the results for the three Polish grid cases studied above are summarized in the Table IX. We add the completion time as observed on the standard Macbook Pro laptop. The completion time of the filtering loop is relatively low compared to the brute force filtering alternative for all three cases. This can be explained by the N_L quadratic complexity of the pruning loop. Meanwhile, the full search (Lines 3–11) completion time depends not only on the complexity of the pruning loop, but also on the complexity of the brute force enumeration over the $\mathcal{C}_2^T$. As the grid becomes more loaded the size of the $\mathcal{C}_2^T$ grows and hence the full search time grows as well. The overall performance of the filtering loop is rather high as even in the highest loaded case the 89% of the contingencies are filtered in just 0.3% of the time required by the brute force method.

D. Stress Analysis for IEEE 300-Bus Case

We analyzed the *IEEE 300-bus* test case to better understand the performance of the algorithm under varying loads. This case is of convenient intermediate size that has potentially large number of contingencies, while still small enough for extensive testing. *IEEE 300-bus* test case as presented in the MATPOWER 4.0 package used by our algorithm has extremely high limits on every line corresponding to 990000 *MW*. To generate more realistic line flow limits we first solved the DC-OPF for base load level and did the extensive analysis of all N-1 configurations. For every line we calculated the maximal flow observed in all N-1 contingencies and set the new limit to 1.25 of this maximal value. This way the system is guaranteed to be N-1 secure but has reasonably small safety margin, and is characterized by multiple N-2 contingencies. To test the performance of the algorithm we have varied the loading level from $s = 0$ to $s = 1.1$ of the base level by uniform rescaling of all load and generator values. For higher values of loading the MATPOWER package cannot find a valid AC power flow solution, so these values are assumed to be unrealistic. For all the loaded cases we tracked the sizes of actually critical non-islanding N-2 contingencies C_2^{final}, size of the candidate set identified by the algorithm C_2^T and the total algorithm running time.

The sizes of both the $\mathcal{C}_2^T$ and the $\mathcal{C}_2^{final}$ sets of critical non-islanding contingencies increase exponentially as the load coefficient grows as can be seen from Fig. 2. Although the efficiency of the filtering is relatively low, so many of the identified contingency candidates turn out to be safe, it does not degrade significantly with high values of loadings.

Completion time of the algorithm shown on Fig. 3 also shows an approximately exponential growth and is dominated by the "brute-force" assessment time of the filtered set C_2^T for high values of loadings. The total running time of the simple "brute-force" algorithm that does not do rely on any pruning was found to be about 80 s, demonstrating acceptable efficiency of the algorithm even for relatively small case. It should be noted that our heuristic way of generating the limits may have underestimated the thermal limits through weakly loaded lines and artificially increased the total number of N-2 contingencies.

TABLE IV
LINE OUTAGE OVERLOAD FACTORS MATRIX ξ_β^α FOR 6 BUS SYSTEM

Lines	1	2	3	4	5	6	7	8	9	10	11
1	0	0.4428	0.3096	0.0004	−0.6311	−0.1102	−0.0795	−0.0833	0.0176	0.0008	−0.0119
2	0.0457	0	0.2874	0.0001	0.846	−0.0357	−0.0257	−0.027	0.0057	−0.0286	−0.0038
3	0.0566	0.509	0	−0.0011	−0.274	0.2855	0.2059	0.2159	−0.0457	0.0504	0.0307
4	−0.0067	−0.021	0.0947	0	0.1451	0.1104	0.2826	−0.2264	−0.6416	0.0126	−0.0111
5	−0.0584	0.7576	−0.1385	−0.0009	0	0.2211	0.1594	0.1672	−0.0354	−0.0759	0.0238
6	−0.0362	−0.1137	0.5136	−0.0024	0.7865	0	0.4315	0.4524	−0.0958	0.0683	0.0644
7	−0.0048	−0.0151	0.0683	−0.0011	0.1046	0.0796	0	−0.069	0.4362	0.0091	−0.0227
8	−0.0065	−0.0204	0.0919	0.0012	0.1407	0.1071	−0.0886	0	0.4926	0.0122	0.0303
9	0.0012	0.0039	−0.0174	0.003	−0.0267	−0.0203	0.5015	0.4411	0	−0.0023	−0.0605
10	0.001	−0.3601	0.3587	−0.0011	−1.068	0.2703	0.1949	0.2044	−0.0433	0	0.0291
11	0.0064	0.0201	−0.0909	−0.0004	−0.1393	−0.106	0.203	−0.2105	0.4702	−0.0121	0
12	0	−0.3886	−0.2717	−0.0004	0.5538	0.0967	0.0697	0.0731	−0.0155	−0.0007	0.0104
13	−0.018	0	−0.1133	−0.0001	−0.3336	0.0141	0.0102	0.0106	−0.0023	0.0113	0.0015
14	−0.0172	−0.155	0	0.0003	0.0834	−0.0869	−0.0627	−0.0657	0.0139	−0.0154	−0.0094
15	0.0067	0.0211	−0.0954	0	−0.1461	−0.1112	−0.2846	0.228	0.6463	−0.0127	0.0112
16	0.0128	−0.1659	0.0303	0.0002	0	−0.0484	−0.0349	−0.0366	0.0078	0.0166	−0.0052
17	0.0093	0.0291	−0.1316	0.0006	−0.2015	0	−0.1106	−0.1159	0.0246	−0.0175	−0.0165
18	0.0028	0.0087	−0.0392	0.0007	−0.0601	−0.0457	0	0.0396	−0.2505	−0.0052	0.0131
19	0.0033	0.0104	−0.0468	−0.0006	−0.0717	−0.0546	0.0451	0	−0.251	−0.0062	−0.0154
20	−0.0003	−0.0009	0.0042	−0.0007	0.0064	0.0049	−0.1204	−0.1058	0	0.0006	0.0145
21	−0.0007	0.2671	−0.2661	0.0008	0.7921	−0.2005	−0.1446	−0.1516	0.0321	0	−0.0216
22	−0.0076	−0.0238	0.1077	0.0005	0.1649	0.1254	−0.2403	0.2492	−0.5566	0.0143	0

TABLE V
FIRST FILTERING MATRIX $U^1_{\alpha\beta}$ FOR 6 BUS SYSTEM

Lines	1	2	3	4	5	6	7	8	9	10	11
1	0	1.959	1.0855	0.0661	1.8293	0.325	0.5106	0.4651	0.663	0.126	0.1076
2	1.959	0	2.1405	0.7785	5.2596	0.9891	1.2149	1.166	1.257	0.878	0.8025
3	1.0855	2.1405	0	0.5882	1.1929	1.1283	1.2413	1.1924	0.9806	0.8199	0.6888
4	0.0661	0.7785	0.5882	0	0.9574	0.3739	0.8948	0.7526	1.512	0.0754	0.0778
5	1.8293	5.2596	1.1929	0.9574	0	1.4914	1.6052	1.5525	1.4123	1.6234	1.0625
6	0.325	0.9891	1.1283	0.3739	1.4914	0	1.0464	0.9967	0.7558	0.5107	0.4794
7	0.5106	1.2149	1.2413	0.8948	1.6052	1.0464	0	0.8048	2.7749	0.6772	0.6771
8	0.4651	1.166	1.1924	0.7526	1.5525	0.9967	0.8048	0	2.161	0.6312	0.8675
9	0.663	1.257	0.9806	1.512	1.4123	0.7558	2.7749	2.161	0	0.6581	1.0146
10	0.126	0.878	0.8199	0.0754	1.6234	0.5107	0.6772	0.6312	0.6581	0	0.1583
11	0.1076	0.8025	0.6888	0.0778	1.0625	0.4794	0.6771	0.8675	1.0146	0.1583	0

TABLE VI
SUMMER OFF-PEAK POLISH POWER GRID. $\mathcal{C}_2^{final} = 463$

t	Size of the set $\mathcal{C}_2^t$	Size of the set S_2^t
1	3,455,928	10,643,906
2	20,269	321,333
3	5,346	201,883
4	5,026	179,553
5	5,011	171,470
6	4,997	170,116
7	4,997	170,091

TABLE VII
WINTER OFF-PEAK POLISH POWER GRID. $\mathcal{C}_2^{final} = 928$.

t	Size of the set $\mathcal{C}_2^t$	Size of the set S_2^t
1	3,602,175	10,880,102
2	54,776	731,023
3	31,727	529,545
4	30,841	490,714
5	30,610	482,124
6	30,599	481,059
7	30,599	480,912

TABLE VIII
WINTER PEAK POLISH POWER GRID. $\mathcal{C}_2^{final} = 15882$

t	Size of the set $\mathcal{C}_2^t$	Size of the set S_2^t
1	2,496,178	8,326,110
2	312,349	1,643,246
3	270,370	1,528,152
4	268,280	1,508,559
5	267,681	1,503,257
6	267,681	1,503,090
7	267,681	1,503,066

VI. DISCUSSION

The computational complexity of filtering the sets $\mathcal{C}_2^t$ and S_2^t and updating bounding matrices decreases with each iteration since it depends on the size of the sets. In the worst case scenario each filtering happens in $O(N_L^2)$ computations, giving the computational complexity of the main loop to be $O(KN_L^2)$ as was shown before. Thus, the computational complexity to find $\mathcal{C}_2^{critical}$ largely depends on three factors. The first factor is the number of elements in the set $\mathcal{C}_1$ which is equal to the number of lines N_L in our case. The second one is the number of iterations T necessary for the main loop to complete. In the performed

TABLE IX
THE SIMULATION RESULTS FOR THE POLISH GRID CASES. TABLE PRESENTS APPROXIMATE SETS SIZES AND COMPLETION TIMES OF THE DEVELOPED ALGORITHM (ALGORITHM 1) AND THE DIRECT BRUTE FORCE ENUMERATION

Case	$\lvert\mathcal{C}_2^1\rvert$	$\lvert\mathcal{C}_2^T\rvert$	$\lvert\mathcal{C}_2^{final}\rvert$	Lines 3-8 completion time, sec	Lines 3-11 completion time, sec	Brute force completion time, sec
Summer off-peak	$3.5 \cdot 10^6$	$5.0 \cdot 10^3$	$4.6 \cdot 10^2$	75	90	$2.8 \cdot 10^4$
Winter off-peak	$3.6 \cdot 10^6$	$3.1 \cdot 10^4$	$9.3 \cdot 10^2$	79	227	$2.9 \cdot 10^4$
Winter peak	$2.5 \cdot 10^6$	$2.7 \cdot 10^5$	$1.6 \cdot 10^4$	59	1130	$1.9 \cdot 10^4$

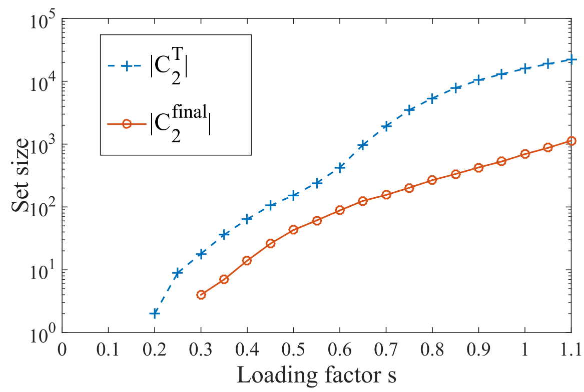

Fig. 2. Sizes of the C_2^T and C_2^{final} sets versus the loading factor s. All contingencies were proved to be safe for $s < 0.3$.

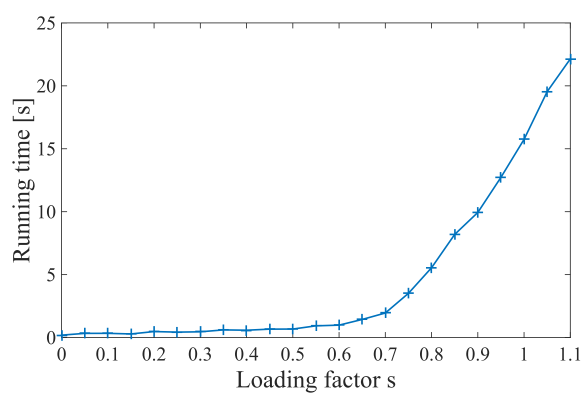

Fig. 3. Total algorithm running time for different values of loading factor s.

simulations K was of order $5 - 7$. Finally, the complexity depends on the size of the filtered set C_2^{final}, which usually was of the same order as N_L in the performed simulations. Taking into account all the factors with their empiric estimations, the total complexity of the algorithm presented in this section can be estimated as $O(N_L^2)$ (assuming that the LODF matrix is given and that the pruning loop is effective enough).

The algorithm poses a trade-off between the size of the set $\mathcal{C}_2^{final}$ and the number of operations necessary to filter the set $\mathcal{C}_2^1$ down to this size. There is a number of possibilities to affect both sides of this trade-off. Firstly, the size of the filtered set can be decreased by *divide and conquer* approach. The input set $\mathcal{C}_2^1$ can be divided into subsets W_1 and W_2. After such a division the problem is decomposed into three subproblems:

- Contingency selection when both outages are in the set W_1
- Contingency selection when both outages are in the set W_2
- Contingency selection when each subset has a single outage

Each of these subproblems will have its own bounding matrices, and the main iteration loop will have to be separately performed for each subset. Using an appropriate choice of the dividing technique, the size of the output set may theoretically be decreased. However, the number of the operations required by this approach is usually substantially higher then the complexity of the original algorithm. One of the cases when the *divide and conquer* technique could be useful is online security assessment of the power grid in situations when the operating point doesn't change substantially over time. In such case the results of a contingency selection from the previous time step can be used to separate the lines that participated in potentially critical contingencies more often than others into a subset. Such lines would typically inflate the bounding matrices and their separation and analysis, hence, this approach can substantially increase efficiency of the contingency selection in the remainder part of the grid.

Another degree of freedom that was not exploited in the presented algorithm is related to the formal definition of the components in (12). The expression $\xi_\gamma^\alpha \Gamma_{\alpha\beta}$ is invariant under the diagonal matrix transformations $\xi_\gamma^\alpha \leftarrow r_\alpha \xi^\alpha$, $\Gamma_{\alpha\beta} \leftarrow r_\alpha^{-1} \Gamma_{\alpha\beta}$ for any non-singular matrix $R = diag(r_1, \ldots, r_{N_L})$. This transformation affects bounding matrices for sets $\mathcal{C}_2^k$, $\mathcal{S}_2^k$, essentially tightening some boundaries and loosening others, and can be used to decrease the size of the output set. The simulation results indicate a reduction of the output set size by the factor of 2 in the best case. However, this reduction comes at the expense of substantial computational overhead, since the appropriate transformation has to be carefully searched for. Nevertheless, such degree of freedom may become important in situations where the original algorithm is not efficient for some reasons.

The main limitation of the algorithm is its reliance on linearized models of power flows. This restriction undermines its applicability in grids where the nonlinear effects are important. As our algorithm is based on bounding techniques, one natural way to extend it to nonlinear models is to bound the effect of nonlinearity can be bounded as well. Examples of such bounds have been recently developed in [29] where explicit relations were given for the maximal error produced by linearization of power flow equations.

VII. CONCLUSIONS

We have presented a novel algorithm for efficient selection of critical N-2 contingencies that result in line overloads in post-contingency equilibrium. The key feature of the algorithm is its zero missing rate: it is guaranteed to identify all the contingencies that are critical in DC approximation. At the same time its running time is several orders of magnitude less than that of the brute-force enumeration alternatives. The performance of the algorithm was illustrated on several case studies of the largest power grid models available in public domain. In all the cases the algorithm performed 30 – 1000 times better than the brute-force enumeration alternative. The relative performance will likely be even better in larger grid models operating in non-extreme loading conditions. The total complexity in this case is expected to scale as $O(N_L^2)$, where N_L is the number of power lines in the system. This complexity is comparable to the standard N-1 security assessment procedure.

Most naturally, the algorithm can be incorporated in real-time security and risk assessment tools. Currently, the selection of N-2 scenarios in operations and planning is based on engineering judgement or heuristic algorithms with uncontrollable performance. Fast screening using the technique described in this paper is a viable alternative to these practices, that may allow updating of the contingency list every 15 minutes or even faster if necessary. At the same time, security assessment is not the only application of the algorithm. Another promising application which would benefit from fast contingency screening algorithm is the problem of transmission topology control [30], [31]. Lines that do not participate in any of the critical N-2 contingencies are natural candidates for line switching actions, as switching of those lines does not violate N-1 security. Finally, it is worth noting that our recent studies of the critical contingency sets [32] have demonstrated that frequency of line participation in critical contingency sets is very non-uniform, with only 0.1% of all lines being overloaded in about half of all N-2 contingencies. Identification of those lines followed by some de-loading can significantly increase the reliability of the system, and can accomplished in real-time operation without significant computational overhead.

References

[1] K. H. LaCommare and J. H. Eto, "Cost of power interruptions to electricity consumers in the United States (US)," *Energy*, vol. 31, no. 12, pp. 1845–1855, 2006.

[2] NERC, Standard TOP-004-2: Transmission Operations, North American Electric Reliability Corporation Std., 2007.

[3] F. F. Alvarado, "Computational complexity in power systems," *IEEE Trans. Power App. Syst.*, vol. PAS-95, no. 4, pp. 1028–1037, Jul. 1976.

[4] G. Ejebe and B. Wollenberg, "Automatic contingency selection," *IEEE Trans. Power App. Syst.*, vol. PAS-98, no. 1, pp. 97–109, Jan. 1979.

[5] G. Irisarri and A. Sasson, "An automatic contingency selection method for on-line security analysis," *IEEE Trans. Power App. Syst.*, vol. PAS-100, no. 4, pp. 1838–1844, Apr. 1981.

[6] B. Stott, O. Alsac, and F. Alvarado, "Analytical and computational improvements in performance-index ranking algorithms for networks," *Int. J. Electr. Power Energy Syst.*, vol. 7, no. 3, pp. 154–160, Jul. 1985.

[7] M. Enns, J. Quada, and B. Sackett, "Fast linear contingency analysis," *IEEE Trans. Power App. Syst.*, vol. PAS-101, no. 4, pp. 783–791, Apr. 1982.

[8] T. Guler and G. Gross, "Detection of island formation and identification of causal factors under multiple line outages," *IEEE Trans. Power Syst.*, vol. 22, no. 2, pp. 505–513, May 2007.

[9] C. M. Davis and T. J. Overbye, "Multiple element contingency screening," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1294–1301, Aug. 2011.

[10] Q. Chen, S. Member, and J. J. D. Mccalley, "Identifying high risk N − k contingencies for online security assessment," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 823–834, May 2005.

[11] H. Mori and Y. Goto, "A tabu search based approach to (n − k) static contingency selection in power systems," in *Proc. 2001 IEEE Int. Conf. Systems, Man, and Cybernetics*, 2001, vol. 3, pp. 1954–1959.

[12] V. Donde, V. Lopez, B. Lesieutre, A. Pinar, and J. Meza, "Severe multiple contingency screening in electric power systems," *IEEE Trans. Power Syst.*, vol. 23, no. 2, pp. 406–417, May 2008.

[13] D. Bienstock and A. Verma, "The n − k problem in power grids: New models, formulations, and numerical experiments," *SIAM J. Optimiz.*, vol. 20, no. 5, pp. 2352–2380, Jun. 2010.

[14] F. Fonteneau-Belmudes, D. Ernst, and L. Wehenkel, "A rare event approach to build security analysis tools when n − k (k > 1) analyses are needed (as they are in large scale power systems)," in *Proc. 2009 IEEE Bucharest PowerTech*, 2009, pp. 1–8.

[15] J. Thorp, "An importance sampling application: 179 bus WSCC system under voltage based hidden failures and relay misoperations," in *Proc. 31st Hawaii Int. Conf. System Sciences*, 1998, vol. 3, pp. 39–46.

[16] Q. Chen and L. Mili, "Risk-based composite power system vulnerability evaluation to cascading failures using importance sampling," in *Proc. 2011 IEEE Power and Energy Soc. General Meeting*, 2011, pp. 1–6.

[17] M. J. M. Eppstein and P. D. H. Hines, "A "random chemistry" algorithm for identifying multiple contingencies that initiate cascading failure," *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1698–1705, Aug. 2012.

[18] P. D. Hines, I. Dobson, E. Cotilla-Sanchez, and M. Eppstein, "Dual graph and random chemistry methods for cascading failure analysis," in *Proc. 2013 46th Hawaii Int. Conf. System Sciences (HICSS)*, 2013, pp. 2141–2150.

[19] K. Turitsyn and P. Kaplunovich, "Fast algorithm for n-2 contingency problem," in *Proc. 2013 46th Hawaii Int. Conf. System Sciences (HICSS)*, 2013, pp. 2161–2166.

[20] F. Galiana, "Bound estimates of the severity of line outages in power system contingency analysis and ranking," *IEEE Trans. Power App. Syst.*, vol. PAS-103, no. 9, pp. 2612–2624, 1984.

[21] V. Brandwajn, "Efficient bounding method for linear contingency analysis," *IEEE Trans. Power Syst.*, vol. 3, no. 1, pp. 38–43, Feb. 1988.

[22] P. Kaplunovich and K. Turitsyn, "Fast selection of n − 2 contingencies for online security assessment," in *Proc. 2013 IEEE Power and Energy Soc. General Meeting (PES)*, 2013, pp. 1–5.

[23] A. Wood and B. Wollenberg, *Power Generation Operation and Control*. New York, NY, USA: Wiley, 1996.

[24] J. Guo, Y. Fu, Z. Li, and M. Shahidehpour, "Direct calculation of line outage distribution factors," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1633–1634, Aug. 2009.

[25] P. Van Hentenryck, *Constraint Satisfaction in Logic Programming*. Cambridge, MA, USA: MIT Press, 1989, vol. 5.

[26] K. S. Turitsyn, N − 2 Contingency Screening Algorithm, 2014 [Online]. Available: https://github.com/pekap/N_2_contingency_analysis

[27] 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," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[28] P. Kaplunovich, "Efficient algorithm for online n − 2 power grid contingency selection," Master's thesis, Massachusetts Inst. Technol., Cambridge, MA, USA, 2014.

[29] S. Bolognani and S. Zampieri, "On the existence and linear approximation of the power flow solution in power distribution networks," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 163–172, Jan. 2016.

[30] K. W. Hedman, R. P. O'Neill, E. B. Fisher, and S. S. Oren, "Optimal transmission switching-sensitivity analysis and extensions," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1469–1479, Aug. 2008.

[31] P. A. Ruiz, A. Rudkevich, M. C. Caramanis, E. Goldis, E. Ntakou, and C. R. Philbrick, "Reduced MIP formulation for transmission topology control," in *Proc. 2012 50th Annu. Allerton Conf. Commun., Control, and Comput. (Allerton)*, 2012, pp. 1073–1079.

[32] P. Kaplunovich and K. Turitsyn, "Statistical properties and classification of n − 2 contingencies in large scale power grids," in *Proc. 2014 47th Hawaii Int. Conf. System Sciences (HICSS)*, 2014, pp. 2517–2526.

P. A. Kaplunovich, photograph and biography unavailable at the time of publication.

K. Turitsyn, photograph and biography unavailable at the time of publication.