Efficient algorithm for online $N - 2$ power grid contingency selection

by

Petr A. Kaplunovich

B.S., Applied Physics and Mathematics, Moscow Institute of Physics and Technology, 2012

Submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2014

Author ....
Department of Mechanical Engineering
December 13, 2013

Certified by..................................... ......
Konstantin Turitsyn
Assistant Professor of Mechanical Engineering
Thesis Supervisor

Accepted by ....
David E. Hardt
Chairman, Department Committee on Graduate Students
Department of Mechanical Engineering

Efficient algorithm for online $N-2$ power grid contingency selection

by

Petr A. Kaplunovich

Submitted to the Department of Mechanical Engineering
on December 13, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

Multiple element outages ($N-k$ contingencies) have caused some of the most massive blackouts and disturbances in the power grid. Such outages affect millions of people and cost the world economy billions of dollars annually. The impact of the $N-k$ contingencies is anticipated to grow as the electrical power grid becomes increasingly more loaded. As the result power system operators face the need for advanced techniques to select and mitigate high order contingencies. This study presents a novel algorithm for the fast $N-2$ contingency selection to address this problem. The developed algorithm identifies all potentially dangerous contingencies with zero missing rate. The complexity of the algorithm is shown to be of the same order as the complexity of $N-1$ contingency selection, which makes it much more efficient than brute force enumeration. The study first derives the equations describing the set of the dangerous $N-2$ contingencies in the symmetric form and presents an effective way to bound them. The derived bounding technique is then used to develop an iterative pruning algorithm. Next, the performance of the algorithm is validated using various grid cases under different load conditions. The efficiency of the algorithm is shown to be rather promising. For the Summer Polish grid case with more than 3500 lines it manages to reduce the size of the contingency candidates set by the factor of 1000 in just 2 iterations. Finally, the reasons behind the efficiency of the algorithm are discussed and intuition around the connection of its performance to the grid structure is provided.

Thesis Supervisor: Konstantin Turitsyn
Title: Assistant Professor

Acknowledgments

First, I would like to thank my adviser, Professor Kostya Turitsyn for all the great things I've learned from him directly and indirectly since the first day I joined his research group. He has provided me with the space and freedom to grow as individual researcher and always encouraged and guided me during my degree journey. I am extremely thankful to Kostya for teaching me how to think about complex problems and how to approach them even if the solution seems infeasible at first glance.

I would like to thank all my friends who have so far made my live journey wonderful. Especially a lot of thanks to Vahe who has been helping me to grow as a person during the nine amazing years of our friendship. I am confident that I would have become a totally different person without him and I can't overestimate the amount of his support and encouragement in every aspect of my life.

Last but not least, I would like to thank my family who have supported me by all possible means in every moment of my life. I thank my brother, Vanya, for subconsciously teaching me how to be a leader for somebody from the very first years of my live. I owe all the accomplishments and success I have had during my life to my parents Larisa and Alexander. Without their tireless work and love, I would never have the opportunities that I have today.

Contents

1 Introduction **11**

1.1 Motivation 11

1.2 Background 13

2 Model and problem formulation **15**

2.1 Grid model 15

2.2 DC power flow approximation 16

2.3 N-2 contingency selection problem 17

3 Contingency equations **19**

3.1 Single line outages. LODF. 19

3.2 Double line outages 21

3.3 Constraints 23

4 Fast N-2 contingency selection algorithm **25**

4.1 General idea 25

4.2 Bounding 26

4.3 Iterative algorithm 27

5 Simulation results **33**

5.1 Initial data 33

5.2 Polish power grid 35

5.2.1 Summer and Winter off-peak cases 35

5.2.2 Winter peak case 37

5.3 IEEE 300-bus test case under different stress conditions 38

6 Analysis of the results **43**

6.1 LODFs distribution 43

6.2 ξ^α and $\Gamma_{\alpha\beta}$ distributions 45

6.3 Distribution of the elements in the bounding matrices 47

7 Conclusion **51**

List of Figures

3-1 Single line outage islanding. $y_\alpha e_\alpha^T B^{-1} e_\alpha = 1$. 20

3-2 Double line outage islanding. $det(D_{\alpha\beta}) = 0$. 23

5-1 $|C_2^{filtered}|$, $|C_2^{final}|$ and completion times of the algorithm. 100 simulations with different values of the load coefficient $s \in [1; s^{max}]$ 40

5-2 The efficiency of the pruning loop $|C_2^{filtered}|/|C_2^{final}|$ and completion times of the pruning loop (Lines 11-16 in the Algorithm) and the full search (Lines 11-20 in the Algorithm). Simulations with different values of the load coefficient $s \in [1; s^{max}]$ 41

6-1 The distribution of Line Outage Distribution Factors $|d_\beta^\alpha|$ for the C_2^1 set for Polish off-peak Summer case. The horizontal axis shows the values of the elements. The vertical axis represent the number of the elements. 44

6-2 The distribution of the elements $\Gamma_{\alpha\beta}$ for the sets C_2^1, C_2^2 , C_2^3 for Polish off-peak Summer case. The horizontal axis shows the values of the elements. The vertical axis represents the number of the elements. . . 45

6-3 The distribution of the elements ξ_β^α for the sets C_2^1, C_2^2 , C_2^3 for Polish off-peak Summer case. The horizontal axis shows the values of the elements. The vertical axis represents the number of the elements. . . 46

6-4 The distribution of the elements in the bounding matrix U^{C_2} for the sets C_2^k for Polish off-peak Summer case. The horizontal axis shows the values of the elements. The vertical axis represents the number of the elements. The dotted line is drawn at 1 and depicts the pruning boundary 48

6-5 The distribution of the elements in the bounding matrix U^{S_2} for the sets S_2^k for Polish off-peak Summer case. The horizontal axis shows the values of the elements. The vertical axis represents the number of the elements. The dotted line is drawn at 1 and depicts the pruning boundary 49

Chapter 1

Introduction

1.1 Motivation

The electric power grid is a very complex system made of billions of components. Any of those elements tripped causes redistribution of power flows and in general changes the state of the grid. Therefore potentially dangerous element outages that could lead to sufficient load loss (blackout) should be anticipated for robust operation of the grid. "The interconnected power system shall be operated at all times so that general system instability, uncontrolled separation, cascading outages, or voltage collapse, will not occur as a result of any single contingency or multiple contingencies of sufficiently high likelihood" [1]

Power grids are generally operated according to $N-1$ security criterion, which assures that outage of any single component of the grid will not lead to violations of bus voltage, branch flow or stability limits. While such protection is enough if the probability of k multiple nearly simultaneous failures ($N-k$ contingencies) is negligibly small, $N-k$ contingencies do occur and can trigger severe cascading outages that result in massive blackouts, such as events of August 14, 2003 in North America [2], November 4, 2006 in Europe [3], July 30 and 31, 2012 in India [4]. Those blackouts have had an enormous impact on the economies, their damage is estimated by billions of dollars. Moreover, according to EATON Blackout Tracker there is a significant number of smaller blackouts that happen in the world on a daily basis. Specifically, only in 2012 there were more than 2800 outages that affected 25 million people in the U.S. [5]. By various estimations power outages and disturbances cost the U.S. economy between \$80 billion and \$188 billion per year.

Besides the notorious blackouts experience, shift towards intermittent renewable generation, electric transportation systems and deregulation of energy markets are among multiple factors that suggest that $N - k$ contingency screening problem will be increasingly more important in the future. Regulatory agencies are increasingly requiring utilities to perform analysis of $N - k$ contingencies. North American reliability standards require that, "Each Transmission Operator shall operate to protect against instability, uncontrolled separation, or cascading outages resulting from multiple outages" [6]. $N - k$ contingency screening allows grid operators to adjust control systems, apply protective actions to the most vulnerable components of the grid as well as to develop appropriate contingency management plans.

However, the computational complexity of finding dangerous $N - k$ contingencies makes brute force enumeration over the space of all contingencies infeasible. Such analysis requires $N!/(k!(N-k)!)$ simulations if the sequence of contingencies is ignored. Even for the case of average sized system with 3000 elements the number of double line outages ($k = 2$) is almost $N_c = 4.5$ million. This means that one needs to run N_c simulations where pairs of elements are tripped one by one, power flow recalculated and appropriate re-dispatching is done if necessary. Even in DC approximation (linearized form of full Kirchhoff equations) the number of computations required to re-solve equations for each simulation is at least $N_s = O(N^{1.2})$ [7]. This results in the complexity of $N - 2$ contingencies brute force screening to be at least $O(N_c N_s) = O(N^{3.2})$. The complexity grows substantially as generalization to full equations or to greater k's is considered. Thus, solving every contingency is quite computationally intensive - to the point of being infeasible for even modest systems, even when linear methods are used. The list of potentially dangerous contingencies should be reduced to make the contingency analysis tractable. Such *online contingency list* can be then used for brute force enumeration.

1.2 Background

The process of dangerous contingency list creation is called *contingency selection* and was developed as a method of determining which contingencies are important enough to be added to the list for online assessment [8]. In the initial work on the topic first-order performance index (PI) sensitivities were used to rank contingencies [8]. However, PI approach proved itself to be unreliable [9], as result approaches based on higher order performance indexes and DC power flow equations were developed [10, 11]. In [11] M. 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 *Outage Distribution Factors* (ODF) [12, 13]. Recent contingency screening and contingency analysis studies also include various techniques based on the network physical and electrical topology analysis [14, 15, 16, 17], mixed integer and nonlinear optimization techniques[18, 19, 20] as well as importance sampling [21, 22, 23] and stochastic approaches [24, 25].

Despite the fact that over the last 35 years there were proposed a plenty of algoritmhs for $N-k$ contingency screening, most of them are based on some heuristic and do not guarantee to find *all* high-risk contingencies. Moreover, major portion of those approaches do not exploit physical structure of the power grid, sometimes even ignoring power flow equations and considering only the topology of the grid. This study argues that such information can be efficiently exploited, and focuses on the developing a fast and reliable algorithm for static $N-2$ contingency screening in DC approximation that guarantees to include all potentially dangerous contingencies in the final online contingency list. The ideology of the approach that will be developed in this study is close to the bounding techniques in the works of F.D. Galiana [26] and V. Brandwajn [27] and is inspired by the usage of line outage distribution factors for contingency screening in the recent works [12, 13].

The results of this study will be presented in the following manner: Chapter 2 will explain the power grid model used in this study and give a formulation of the problem. Chapter 3 will derive equations governing $N-2$ contingencies and discuss how their structure can be exploited to develop an efficient screening algorithm. Chapter 4 will introduce the algorithm and discuss the ways it can be improved. Chapter 5 will present simulation results for different IEEE test cases from MATPOWER package. Chapter 6 will explain the reasons for the high efficiency of the algorithm. Conclusions will be summarized in Chapter 7.

Chapter 2

Model and problem formulation

This chapter will describe the problem being solved as well as initial assumptions and the model used. The complexity of the brute force approach to solve the problem in the given formulation will be also discussed which will serve as a primary motivation for the study.

2.1 Grid model

In this study we consider the grid with $N_B + 1$ buses and N_L branches. The grid is modeled as an undirected graph $G = (V, E)$, $|V| = N_B + 1$, $|E| = N_L$, where each vertex $v_i \in V$ represents a load, generation or just an empty bus and each edge $e_k = (v_i, v_j) \in E$ represents a branch connecting bus v_i with bus v_j. For such graph the arc-node incidence matrix A is defined as $N_L \times (N_B + 1)$ matrix where the jth column of A represents the jth vertex, v_j, and ith row represents the ith edge, e_i, in G. Each row has only two non-zeros at the columns that represent the vertices of the respective line. Formally, the a_{ij} entry of the matrix A is defined as follows:

$$a_{ij} = \begin{cases} 1, & \text{if } e_i = (v_j, \cdot) \\ -1, & \text{if } e_i = (\cdot, v_j) \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

Having the matrix A the $(N_B + 1) \times (N_B + 1)$ nodal DC susceptance matrix (the imaginary part of the admittance matrix for a general system) is defined as $B = A^T Y A$, where Y is the diagonal $N_L \times N_L$ matrix of the branch susceptances, $Y = diag(y_1, ..., y_{N_L})$. The matrix B defines the structure of the grid, containing all information about branches.

2.2 DC power flow approximation

Significant number of power system applications rely on linear network models. Their properties have considerable analytical and computational appeal. Among those properties are reliable and unique solutions, minimal network data necessary for modeling as well as reasonably accurate MW flows results [28]. This study limits itself to a very popular, DC linear model of power flow equations which is widely used in various power system applications. The DC assumptions are the follwing:

- no resistive losses (in real power systems branch resistance is much smaller then its reactance);
- angle differences across lines are small (so that $sin(\Delta\theta) \approx \Delta\theta$; this assumption holds very well in most power systems);
- bus voltage amplitudes are controlled on each bus to be the same ($V_0 = 1\,p.u$).

With these assumptions the state of the power system is described by the $(N_B + 1) \times 1$ vector of voltage phases θ on every of the $N_B + 1$ buses in the system. Power flow equations in such system give a relationship between vector of voltage phases, structure of the grid (which is described by sucsepatance matrix B) and p - the $(N_B + 1) \times 1$ vector of power injections at each bus. In matrix form power flow equations are written as:

$$p = B\theta \tag{2.2}$$

Equation 2.2 is the main equation in the DC approximation contingency study. Given a power flow vector p and knowing the structure of the grid B one can find corresponding vector of phase angles θ by simply solving the system of linear equations. However, this system has infinite number of solutions since matrix B is singular ($det(B) = 0$). By denoting $N_B + 1$ bus as the reference bus, and deleting corresponding entries from the Jacobian matrix B and the vectors θ and p the system 2.2 can be constrained to have a unique solution for the vector of voltage phases θ. This solution can be used to find the $N_L \times 1$ vector f of power flows on all branches:

$$f = YA\theta = YAB^{-1}p \tag{2.3}$$

2.3 N-2 contingency selection problem

Given the DC power flow equations and the model of the grid as an undirected graph the contingency selection problem can be formulated as follows. The set C_1 is defined as the set of one-element outage ($N-1$) contingencies corresponding to configurations of the grid where one element has failed. Particularly in this work the set C_1 consists of the grid's branches, however in general this set can include any system components. Since the number of lines in the grid is N_L, the size of the set of $N-1$ contingencies is N_L as well. The set of double element outages can be formed as a Cartesan product $C_2 = C_1 \times C_1$ without duplicates. This set represents the pairs of failed elements. When the considered contingency elements are branches, the size of C_2 is $N_L(N_L - 1)/2$ with an assumption that the sequence of the contingencies is ignored (which is the case in the static contingency screening approaches). The occurrence of each contingency from the set C_2 can cause a violation of the system constraints, and if it does so such contingency is considered to be a *dangerous contingency*. The $N-2$ contingency selection algorithm that will be developed in this work aims to find a set C_2^{final} of those non-islanding double outages that are dangerous in the defined sense.

In general case the set of the system constraints is associated with power flows on branches and voltage levels on individual buses. However, this study limits itself only to the constraints associated with power flow limits, since DC approximation assumes a flat voltage profile and is unable to account for voltage violations and stability issues. Given the power flows through all branches f and the $N_L \times 1$ vector of power flow limits f^{max}, the set of system constraints can be described as following:

$$|f| < f^{max} \tag{2.4}$$

The set of constraints 2.4 gives a feasibility region for the problem being studied. Any dangerous contingency should cause a change of the state of the grid such that it will not be within the defined feasibility region.

The complexity of the brute force approach to solve the $N-2$ contingency selection problem defined in this section is rather high. As was noticed in Chapter 1, exhaustive enumeration of the set C_2 would require at least $O(N_L^{3.2})$ operations. The complexity grows very fast as the number of the grid elements increases, which serves as a primary motivation to develop a fast and reliable algorithm for obtaining C_2^{final}.

The next chapter will derive the equations governing $N-2$ contingencies in the presented formulation.

Chapter 3

Contingency equations

In this chapter the equations governing $N-1$ and $N-2$ contingencies will be derived using LODFs. The structure of the equations will be discussed and the constraints for $N-2$ contingencies (2.4) will be rewriten in the form necessary for developing the fast contingency selection algorithm.

3.1 Single line outages. LODF.

Tripping of some set of lines changes the structure of the grid, which is described by the reduced susceptance matrix B in our model. In the case of one line α outage the corresponding susceptance matrix B_α for the grid can be represented as:

$$\begin{aligned} B_\alpha &= B + \Delta B_\alpha \\ \Delta B_\alpha &= -e_\alpha(y_\alpha e_\alpha)^T \end{aligned} \tag{3.1}$$

Where e_α is α's row of the matrix A, i.e. e_α has only two nonzero entries corresponding to the buses on the ends of the line α. With this new susceptance matrix B_α, the $N_L \times 1$ vector of active power flow changes Δf^α can be written using 2.3 as follows:

$$\Delta f^\alpha = f^\alpha - f = YA(B_\alpha^{-1} - B^{-1})p \tag{3.2}$$

The $(B_\alpha^{-1} - B^{-1})$ term in 3.2 can be readily simplified using *matrix inversion lemma*:

$$
\begin{aligned}
B_\alpha^{-1} - B^{-1} &= B^{-1}(e_\alpha q_\alpha) y_\alpha e_\alpha^T B^{-1} \\
q_\alpha &= \frac{1}{1 - y_\alpha e_\alpha^T B^{-1} e_\alpha}
\end{aligned}
\tag{3.3}
$$

Denominator of q_α is zero when $y_\alpha e_\alpha^T B^{-1} e_\alpha = 1$, which happens when the grid resistance between end points of line α is equal to the resistance of the line α. In its turn the later means that the tripping of the line leads to an islanding of the grid (Fig. 3-1). In this work a single line outage that leads to an islanding is considered to be a dangerous islanding $N-1$ contingency. It is reasonable to do so, because usually one of two islands separated by an outaged line contains only loads that are shed after such contingency. In general, both islands can contain generators and such case should be analysed separately re-dispatching the generation and resolving power flow equations for each island. We separate all single outage contingencies leading to an islanding into the set C_1^{isl} defined as follows:

$$
C_1^{isl} = \{\alpha \in C_1 : y_\alpha e_\alpha^T B^{-1} e_\alpha = 1\}
\tag{3.4}
$$

Considering all one line outages leading to an islanding of the grid as dangerous contingencies during the screening leaves us with only outages that have nonsingular values of q_α.

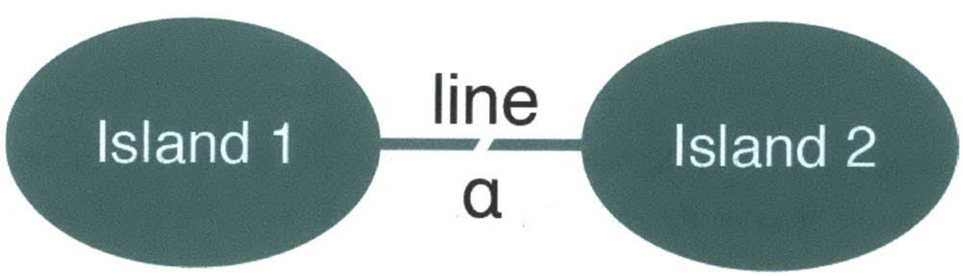

Figure 3-1: Single line outage islanding. $y_\alpha e_\alpha^T B^{-1} e_\alpha = 1$.

We finally represent the vector Δf^α using equations 3.3 as follows:

$$\begin{aligned}\Delta f^\alpha &= d^\alpha f_\alpha \\ d^\alpha &= YAB^{-1}(e_\alpha q_\alpha)\end{aligned} \tag{3.5}$$

Vector d^α defined in 3.5 is the vector of Line Outage Distribution Factors. It plays an important role in the contingency selection studies, and represents linear sensitivities of the line flows to the pre-outage flow on the outaged line α [29]. Using LODFs the flow on any line β after an outage of the line α is given by a simple relationship:

$$f_\beta^\alpha = f_\beta + d_\beta^\alpha f_\alpha \tag{3.6}$$

In the following the similar relationship will be established for the case of multiple line outages, and as will be shown in the next section, single line outage distribution factors defined in 3.6 will play a crucial role in that relationship. This brings up a question about the complexity of calculating all LODFs for a given grid, which is essentially the same as the complexity of solving $N-1$ contingency selection problem in the DC approximation. As was discussed earlier this complexity is about $O(N_L^{2.2})$.

3.2 Double line outages

The formula for multiple line outages will be derived in a very similar manner to the derivation of the relationship for single line contingencies in the previous section.

After an outage of the lines α and β the susceptance matrix $B_{\alpha\beta}$ for the corresponding grid is represented as:

$$\begin{aligned}B_{\alpha\beta} &= B + \Delta B_{\alpha\beta} \\ \Delta B_{\alpha\beta} &= \Delta B_\alpha + \Delta B_\beta \\ \Delta B_{\alpha\beta} &= -\begin{pmatrix} e_\alpha & e_\beta \end{pmatrix} \begin{pmatrix} y_\alpha e_\alpha^T \\ y_\beta e_\beta^T \end{pmatrix}\end{aligned} \tag{3.7}$$

For the sake of simplification of the derivation this matrix can be written in the equivalent form:

$$B_{\alpha\beta} = B + \begin{pmatrix} q_\alpha e_\alpha & q_\beta e_\beta \end{pmatrix} \begin{pmatrix} -\frac{1}{q_\alpha} & \\ & -\frac{1}{q_\beta} \end{pmatrix} \begin{pmatrix} y_\alpha e_\alpha^T \\ y_\beta e_\beta^T \end{pmatrix} \tag{3.8}$$

With this new susceptance matrix $B_{\alpha\beta}$, the vector of the flow changes $\Delta f^{\alpha\beta}$ can be written using 2.3 as follows:

$$\Delta f^{\alpha\beta} = f^{\alpha\beta} - f = YA(B_{\alpha\beta}^{-1}B - I)B^{-1}p \tag{3.9}$$

The $(B_{\alpha\beta}^{-1}B - I)$ term in 3.9 can be simplified using matrix inversion lemma:

$$\begin{aligned}
B_{\alpha\beta}^{-1}B - I &= B^{-1} \begin{pmatrix} q_\alpha e_\alpha & q_\beta e_\beta \end{pmatrix} D_{\alpha\beta}^{-1} \begin{pmatrix} y_\alpha e_\alpha^T \\ y_\beta e_\beta^T \end{pmatrix} \\
D_{\alpha\beta} &= \begin{pmatrix} q_\alpha(1 - y_\alpha e_\alpha^T B^{-1} e_\alpha) & -y_\alpha e_\alpha^T B^{-1} q_\beta e_\beta \\ -y_\beta e_\beta^T B^{-1} q_\alpha e_\alpha & q_\beta(1 - y_\beta e_\beta^T B^{-1} e_\beta) \end{pmatrix} \\
d_\beta^\alpha &= y_\beta e_\beta^T B^{-1} q_\alpha e_\alpha \qquad d_\alpha^\beta = y_\alpha e_\alpha^T B^{-1} q_\beta e_\beta \\
q_\alpha &= \frac{1}{1 - y_\alpha e_\alpha^T B^{-1} e_\alpha} \qquad q_\beta = \frac{1}{1 - y_\beta e_\beta^T B^{-1} e_\beta} \\
B_{\alpha\beta}^{-1}B - I &= B^{-1} \begin{pmatrix} q_\alpha e_\alpha & q_\beta e_\beta \end{pmatrix} \begin{pmatrix} 1 & -d_\alpha^\beta \\ -d_\beta^\alpha & 1 \end{pmatrix}^{-1} \begin{pmatrix} y_\alpha e_\alpha^T \\ y_\beta e_\beta^T \end{pmatrix}
\end{aligned} \tag{3.10}$$

Finally, the vector of the power flow changes $\Delta f^{\alpha\beta}$ can be written as:

$$\Delta f^{\alpha\beta} = \begin{pmatrix} d^\alpha & d^\beta \end{pmatrix} \begin{pmatrix} 1 & -d_\alpha^\beta \\ -d_\beta^\alpha & 1 \end{pmatrix}^{-1} \begin{pmatrix} f_\alpha \\ f_\beta \end{pmatrix} \tag{3.11}$$

Where $d^\alpha = YAB^{-1}(e_\alpha q_\alpha)$, $d^\beta = YAB^{-1}(e_\beta q_\beta)$ defined earlier were used.

The equation 3.11 establishes an important relationship that is in the core of the present work. One of its main features is the fact that the effect of the double line outage is represented through the effects of the single line outages (LODFs), which means that once calculated for $N-1$ contingency selection, these factors can be reused for $N-2$ contingency selection.

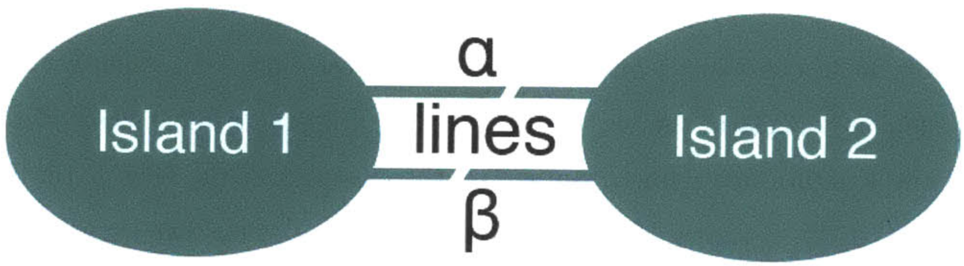

Figure 3-2: Double line outage islanding. $det(D_{\alpha\beta}) = 0$.

The matrix $D_{\alpha\beta}$ is singular if and only if the outage of lines α and β leads to an islanding of the power grid Fig. 3-2 (in other words, if and only if they form at least one minimal cutset of the corresponding graph) [13]. Similarly to the case of $N-1$ contingency, any $N-2$ contingency that leads to an islanding of the grid ($det(D_{\alpha\beta}) = 0$) will be considered as a dangerous islanding $N-2$ contingency. All such contingencies are separated into the set C_2^{isl}:

$$C_2^{isl} = \{(\alpha, \beta) \in C_2 : \det(D_{\alpha\beta}) = 0\} \tag{3.12}$$

3.3 Constraints

Having the relationship 3.11, a feasibility region 2.4 for $N-2$ contingency selection problem takes the form of:

$$|f + \begin{pmatrix} d^{\alpha} & d^{\beta} \end{pmatrix} D_{\alpha\beta}^{-1} \begin{pmatrix} f_{\alpha} \\ f_{\beta} \end{pmatrix} | < f^{max} \tag{3.13}$$

Which can be rewritten as the set of constraints:

$$\xi^\alpha \Gamma_{\alpha\beta} + \xi^\beta \Gamma_{\beta\alpha} < 1$$

$$\xi^\alpha = \begin{pmatrix} \operatorname{diag}(f^{max} - f)^{-1} f_\alpha d^\alpha \\ \operatorname{diag}(-f^{max} - f)^{-1} f_\alpha d^\alpha \end{pmatrix} \quad \Gamma_{\alpha\beta} = D^{-1}_{\alpha\beta} \begin{pmatrix} 1 \\ f_\beta / f_\alpha \end{pmatrix} \tag{3.14}$$

It is important to consider an interpretation of the vector ξ^α and the matrix Γ. Each pair of the components of the vector ξ^α, $(\xi^\alpha_z, \xi^\alpha_{z+N_L})$, contains information about the relative effect of the single line α outage on the closeness of the flow f_z on the line z to its limit f^{max}_z, it shows *how much the single outage of the line α influences the state of the line z.* Whereas each component of the matrix Γ, $\Gamma_{\alpha\beta}$, represents *the interference of the line β on the line α*, or *the relative value of the effective flow on the line α when both lines are tripped* [12].

There are $2N_L$ constraints for each double line outage contingency to be verified in order to certify a contingency to be either safe or dangerous using the brute force approach. Therefore 3.14 itself doesn't decrease the complexity of the contingency selection yet. However, the form of 3.14 can be utilized in order to develop an effective algorithm.

The next chapter will focus on the developing a fast algorithm for $N-2$ contingency selection.

Chapter 4

Fast N-2 contingency selection algorithm

The main goal of any fast contingency selection algorithm is to avoid direct enumeration over the set of contingencies/constraints. This chapter will present an efficient algorithm that exploits the structure of equations 3.14 to iteratively prune the set of potentially dangerous contingencies in the time compared to the $N-1$ contingency selection. The presented algorithm will avoid complete enumeration using smart bounding of different factors in the equations 3.14.

4.1 General idea

The problem of finding the set of all dangerous non-islanding contingencies C_2^{final} from the set C_2 is essentially the problem of finding all pairs (α, β) such that for each pair there exists at least one constraint z from the set of constraints $S_1 = \{i = 1..2N_L\}$ that is violated. Mathematically it can be written as:

$$C_2^{final} = \{(\alpha, \beta) \in C_2 : \exists z \in S_1,\ \xi_z^{\alpha}\Gamma_{\alpha\beta} + \xi_z^{\beta}\Gamma_{\beta\alpha} > 1\} \tag{4.1}$$

However, instead of enumerating the large sets C_2 and S_1 to find C_2^{final} itself, it is appealing to first filter C_2 and S_1 to decrease the number of candidate contingencies and constraints. If it is possible to obtain small enough filtered sets $C_2^{filtered}$ and $S_1^{filtered}$ then the complexity of finding C_2^{final} will be substantially decreased:

$$C_2^{final} = \{(\alpha, \beta) \in C_2^{filtered} : \exists z \in S_1^{filtered},\ \xi_z^{\alpha}\Gamma_{\alpha\beta} + \xi_z^{\beta}\Gamma_{\beta\alpha} > 1\} \tag{4.2}$$

4.2 Bounding

The set S_2 of contingency-constraint pairs (α, z) is defined as a Cartesan product $S_2 = C_1 \times S_1$, and the set S_2^{final} as follows:

$$S_2^{final} = \{(\alpha, z) \in S_2 : \exists \beta \in C_1,\ \xi_z^{\alpha}\Gamma_{\alpha\beta} + \xi_z^{\beta}\Gamma_{\beta\alpha} > 1\} \tag{4.3}$$

Using 4.1 and 4.3 the following conditions can be readily obtained:

$$\begin{aligned}
(\alpha, \beta) \in C_2 &: U_{\alpha\beta}^{C_2} \overset{def}{=} \max_z(\xi_z^{\alpha}\Gamma_{\alpha\beta} + \xi_z^{\beta}\Gamma_{\beta\alpha}) < 1 \quad \Rightarrow \quad (\alpha, \beta) \notin C_2^{final} \\
(\alpha, z) \in S_2 &: U_{\alpha z}^{S_2} \overset{def}{=} \max_{\beta}(\xi_z^{\alpha}\Gamma_{\alpha\beta} + \xi_z^{\beta}\Gamma_{\beta\alpha}) < 1 \quad \Rightarrow \quad (\alpha, z) \notin S_2^{final}
\end{aligned} \tag{4.4}$$

In other words, any pair of lines (α, β) from the set C_2, for which the upper bound $U_{\alpha\beta}^{C_2}$ defined in 4.4 is less then 1, is certified to be safe and can be removed from the search space C_2. Meanwhile, any pair (α, z) from the set S_2 can be removed from the search space S_2 if its corresponding bound $U_{\alpha z}^{S_2} < 1$. This admits to an idea of calculating bounding matrices U^{C_2}, U^{S_2} to prune corresponding sets C_2, S_2. The elements of the bounding matrices U^{C_2}, U^{S_2} can be represented as follows:

$$\begin{aligned}
U_{\alpha\beta}^{C_2} &= \max_z(\xi_z^{\alpha}\Gamma_{\alpha\beta}) + \max_z(\xi_z^{\beta}\Gamma_{\beta\alpha}) \\
&= \max\{\Gamma_{\alpha\beta}\max_z(\xi_z^{\alpha}), \Gamma_{\alpha\beta}\min_z(\xi_z^{\alpha})\} + \max\{\Gamma_{\beta\alpha}\max_z(\xi_z^{\beta}), \Gamma_{\beta\alpha}\min_z(\xi_z^{\beta})\}
\end{aligned} \tag{4.5}$$

$$U_{\alpha z}^{S_2} = \max_\beta(\xi_z^\alpha \Gamma_{\alpha\beta}) + \max_\beta(\xi_z^\beta \Gamma_{\beta\alpha}) = \max\{\xi_z^\alpha \max_\beta(\Gamma_{\alpha\beta}), \xi_z^\alpha \min_\beta(\Gamma_{\alpha\beta})\}$$
$$+ \max\{\max_\beta(\Gamma_{\beta\alpha}) \max_\beta(\xi_z^\beta), \min_\beta(\Gamma_{\beta\alpha}) \min_\beta(\xi_z^\beta)\} \quad (4.6)$$

In equations 4.5 and 4.6 all components of both $U_{\alpha\beta}^{C_2}$ and $U_{\alpha z}^{S_2}$ can be calculated before computing U^{C_2} and U^{S_2} in just $O(|C_2| + |S_2|)$ operations, which gives a complexity of computing matrices U^{C_2} and U^{S_2} to be $O(N_L)^2$ if the LODFs matrix is given. It should be noticed that the matrices defined in 4.5 and 4.6 are not exactly the matrices in 4.4, they are rather pessimistic bounds on them.

4.3 Iterative algorithm

The algorithm being constructed in this section works towards obtaining the set C_2^{final} via pruning the sets C_2 and S_2. Each k's iteration of the algorithm starts with sets C_2^k and S_2^k containing pairs that are still subject to verification. Based on k's sets, the algorithm computes the new bounding matrices $U_{\alpha\beta}^{C_k}$, $U_{\alpha z}^{S_k}$ and uses them to prune C_2^k, S_2^k producing C_2^{k+1}, S_2^{k+1}. The pruning happens according to the equations 4.4 rewritten for sets C_2^k and S_2^k:

$$\begin{aligned} C_2^{k+1} &= \{(\alpha,\beta) \in C_2^k : U_{\alpha\beta}^{C_2^k} > 1\} \\ S_2^{k+1} &= \{(\alpha,z) \in S_2^k : U_{\alpha z}^{S_2^k} > 1\} \end{aligned} \quad (4.7)$$

From 4.7 the following relationships for the structure of the pruned sets are true:

$$\begin{aligned} C_2^{final} \subseteq C_2^{filtered} \subseteq C_2^{k+1} \subseteq C_2^k \subseteq C_2^1 \subseteq C_2 \\ S_2^{final} \subseteq S_2^{filtered} \subseteq S_2^{k+1} \subseteq S_2^k \subseteq S_2^1 \subseteq S_2 \end{aligned} \quad (4.8)$$

The reason behind iterative nature of the algorithm is the fact that pruned sets C_2^{k+1}, S_2^{k+1} could produce tighter bounds than preceding sets C_2^{k+1}, S_2^{k+1}. Therefore the following relationship for all elements in bounding matrices is held:

$$\begin{aligned} U_{\alpha\beta}^{C_2^{final}} \le U_{\alpha\beta}^{C_2^{filtered}} \le U_{\alpha\beta}^{C_2^{k+1}} \le U_{\alpha\beta}^{C_2^k} \le U_{\alpha\beta}^{C_2^1} \le U_{\alpha\beta}^{C_2} \\ U_{\alpha z}^{S_2^{final}} \le U_{\alpha z}^{S_2^{filtered}} \le U_{\alpha z}^{S_2^{k+1}} \le U_{\alpha z}^{S_2^k} \le U_{\alpha z}^{S_2^1} \le U_{\alpha z}^{S_2} \end{aligned} \quad (4.9)$$

The algorithm starts with sets C_2^1, S_2^1 and iteratively prunes them until the sets stop changing. In K iterations the algorithm outputs the sets $C_2^{filtered}$ and $S_2^{filtered}$ that then have to be enumerated over to produce the final set of dangerous contingencies C_2^{final}. It is worth to be noticed that S_2^{final} does not have to be searched for, since the goal is to find only the set of dangerous contingencies. Thus the complexity of the algorithm can be estimated as $O(KN_L^2 + N_L|C_2^{filtered}|)$ since each iteration requires $O(N_L^2)$ computations and the total number of iterations is K (if the LODF matrix is given).

As was discussed in 3.1 and 3.2 all outages that lead to an immediate islanding are considered to be dangerous. In Chapter 3 these contingencies were separated into the sets C_1^{isl} and C_2^{isl} that can be easily found before running the algorithm by computing corresponding values q_α defined in 3.3 and determinants of matrices $D_{\alpha\beta}$ defined in 3.10. After finding the sets C_1^{isl} and C_2^{isl} the algorithm is initiated with the sets C_2^1 and S_2^1 defined as follows:

$$\begin{aligned} C_2^1 &= C_2 \setminus [C_2^{isl} \cup (C_1^{isl} \times C_1)] \\ S_2^1 &= S_2 \setminus [C_1^{isl} \times S_1] \end{aligned} \tag{4.10}$$

Excluded set of double contingencies that result in islandings has to be added to the final contingencies set giving the full set of dangerous contingencies:

$$C_2^{dangerous} = C_2^{final} \cup [C_2^{isl} \cup (C_1^{isl} \times C_1)] \tag{4.11}$$

The formal representation of the algorithm discussed above is given in the **Algorithm 1**. The loop defined at lines 11-16 is the main iteration loop that represents the essence of the approach.

The computational complexity of pruning the sets C_2^k and S_2^k and to update bounding matrices decreases with each iteration since it depends on the size of the sets. In the worst case scenario each pruning 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 $C_2^{dangerous}$ largely depends on three factors. The

Algorithm 1 `N-2 contingency selection`

for $\alpha \in C_1$ **do** $\qquad \triangleright$ Complexity $\sim O(N_L^{2.2})$
    Calculate q_α using 3.3
end for
$C_1^{isl} \leftarrow \{\alpha \in C_1 : 1/q_\alpha = 0\}$ $\qquad \triangleright$ Form the set of single islanding outages

for $(\alpha, \beta) \in C_2$ **do** $\qquad \triangleright$ Complexity $O(N_L^2)$
    Calculate LODF matrix using 3.10
end for
$C_2^{isl} \leftarrow \{(\alpha, \beta) \in C_2 : \det(D_{\alpha\beta}) = 0\}$ $\qquad \triangleright$ Form the set of double islanding outages

$C_2^1 \leftarrow C_2 \setminus [C_2^{isl} \cup (C_1^{isl} \times C_1)]$ $\qquad \triangleright$ Initialize C_2^1
$S_2^1 \leftarrow S_2 \setminus [C_1^{isl} \times S_1]$ $\qquad \triangleright$ Initialize S_2^1

repeat
    Calculate $U_{\alpha\beta}^{C_2^k}$ $\qquad \triangleright$ Update bounding matrices
    Calculate $U_{\alpha\beta}^{S_2^k}$ $\qquad \triangleright$ Complexity $\sim O(N_L^2)$
    $C_2^{k+1} \leftarrow \{(\alpha, \beta) \in C_2^k : U_{\alpha\beta}^{C_2^k} > 1\}$ $\qquad \triangleright$ Prune C_2^k, complexity $O(|C_2^k|) \leq O(N_L^2)$
    $S_2^{k+1} \leftarrow \{(\alpha, z) \in S_2^k : U_{\alpha z}^{S_2^k} > 1\}$ $\qquad \triangleright$ Prune S_2^k, complexity $O(|S_2^k|) \leq O(N_L^2)$
until $C_2^k = C_2^{k+1}$ and $S_2^k = S_2^{k+1}$ $\qquad \triangleright$ Until sets stop changing

$C_2^{filtered} \leftarrow C_2^K$, $S_2^{filtered} \leftarrow S_2^K$
Brute force search: $\qquad \triangleright$ Complexity $O(N_L|C_2^{filtered}|)$
$C_2^{final} \leftarrow \{(\alpha, \beta) \in C_2^{filtered} :$ Constraints 3.14 are satisfied$\}$
$C_2^{dangerous} \leftarrow C_2^{final} \cup [C_2^{isl} \cup (C_1^{isl} \times C_1)]$

return $C_2^{dangerous}$

first factor is the number of elements in the set C_1 which is equal to the number of lines N_L in our case. The second one is the number of iterations K necessary for the main loop to complete. As will be shown in the next chapter, K was of order $5-7$ in the performed simulations. Finally, the complexity depends on the final size of the filtered set $C_2^{filtered}$, which usually was of the same order as N_L in the simulations. Taking into account all the factors with their empiric estimations, the total complexity of the pruning algorithm presented in this section can be estimated as $O(N_L^2)$ (assuming that the LODF matrix is given).

The pruning algorithm poses a trade-off between the size of the set $C_2^{filtered}$ and the number of operations necessary to filter the set 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 C_2^1 can be divided into subsets W_1 and W_2. After such division the problem separates into three subproblems:

- Contingency selection when both outages occur in the set W_1
- Contingency selection when both outages occur 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 selection of the dividing technique, the size of the output set may theoretically be decreased. However, the number of the computations 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, when an 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 dangerous contingencies more often than others into a subset. Such lines would typically inflate the bounding matrices and their separation and analysis, hence, can substantially increase efficiency of the contingency selection in the remainder part of the grid. The study of different dividing methods is beyond the scope of this work and has to be performed to establish positive results.

Another degree of freedom that was left behind the presented algorithm is related to the formal definition of the components in 3.14. The expression $\xi^\alpha \Gamma_{\alpha\beta}$ is invariant under the diagonal matrix transformations $\xi^\alpha \leftarrow r_\alpha \xi^\alpha$, $\Gamma_{\alpha\beta} \leftarrow r_\alpha^{-1} \Gamma_{\alpha\beta}$ for any non-singular matrix $R = diag(r_1, \dots, r_{N_L})$. This transformation affects bounding matrices for sets C_2^k, 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 cases. 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 next chapter will present experimental results of an implementation of the presented algorithm.

Chapter 5

Simulation results

This chapter will present the simulation results for the fast contingency selection algorithm described in the Chapter 4. Various MATPOWER grid cases under different operating conditions will be examined. The performance of the algorithm will be analysed and compared to the brute force search.

5.1 Initial data

Initial data for this study was obtained using MATPOWER package for MATLAB [30]. First, the steady state DC Optimal Power Flow procedure from the package was used to calculate the vector of power injections p for a given grid case. Then all grid cases were studied in the same manner in order to avoid a bias in results. The following are the steps performed before running $N-2$ contingency selection procedure:

1. **Grid normalization.** Some cases from MATPOWER package contain buses with multiple generators at them. For each such case multiple generators at a bus were aggregated. Besides, in order to simplify equations and calculations all information about phase shifters and transformers was deleted from the studied case. Moreover, all parallel lines $l^1_{v_i,v_j}, \ldots, l^k_{v_i,v_j}$ between nodes v_i, v_j were replaced by a single line with corresponding susceptance given by the

following relationship:

$$\frac{1}{y} = \sum_{l=l^1_{v_i,v_j},\dots,l^k_{v_i,v_j}} \frac{1}{y_l} \tag{5.1}$$

2. **$N-1$ contingency selection and LODF computation.** Brute force $N-1$ contingency selection was performed. During the $N-1$ contingency selection the set of single islanding contingencies C_1^{isl} and the set of dangerous non-radial single outages $C_1^{dangerous}$ were identified. A matrix of line outage distribution factors corresponding to the studied case was calculated using the equations 3.5.

3. **$N-1$ contingency protection.** A studied case was protected from all dangerous non-radial $N-1$ contingencies by increasing flow limits for all violated lines in the following manner. For each violated line l the limit was increased such that in the worst case of a single line outage the flow $f_l^{worst\ case}$ on the line l would be as relatively close to the flow limit f_l^{max} as the minimum relative margin for all non-violated lines. In other words:

$$\begin{aligned}
C_1^{violated} &= \{l \in C_1 : \exists \alpha \in C_1^{dangerous}, |f_l^\alpha| > f_l^{max}\} \\
m_l &= \max_{\alpha \in C_1 \setminus l} \frac{f_l^\alpha}{f_l^{max}} \\
M &= \max_{l \in C_1 \setminus C_1^{violated}} m_l \\
f_l^{max} &\leftarrow f_l^{max} \frac{m_l}{M}, \quad \forall l \in C_1^{violated}
\end{aligned} \tag{5.2}$$

4. **Input sets C_2^1 and S_2^1** for the pruning loop of the algorithm were found using 4.10.

5.2 Polish power grid

5.2.1 Summer and Winter off-peak cases

The first simulations were 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, 399 generators and 3506 lines, which suggests the size of the set C_2 to be $(N_L-1)N_L/2 \sim 6*10^6$. After the preparation steps presented in the previous section, the grid was analysed using the fast contingency selection algorithm developed in the Chapter 4. The algorithm has managed to reduce the size of the non-islanding double outage contingencies set from $|C_2^1| = 3.5 * 10^6$ to $|C_2^3| = 5346$ in just 2 iterations. The Table 5.1 shows the evolution of the candidate sets C_2^k and S_2^k as the algorithm proceeds. As can be seen from this table the algorithm converged to the set $C_2^{filtered}$ in 6 iterations, but only the first two of them led to the strong reduction of the candidate sets. The size of the final set of dangerous double outage non-islanding contingencies C_2^{final} given by the algorithm is equal to $|C_2^{final}| = 463$. Thereby the set of the potentially dangerous contingencies $C_2^{filtered}$ obtained during the iterative loop of the algorithm (Lines 11-16) was just ~ 10 times larger then the sough-for set C_2^{final}.

The simulations performed on the Winter off-peak case have shown very similar convergence results (Table 5.2). 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 dangerous non-islanding $N-2$ contingencies $|C_2^{final}| = 928$ and slightly less effective performance of the pruning loop ($|C_2^{filtered}|/|C_2^{final}| \sim 30$).

k	Size of the set C_2^k	Size of the set S_2^k
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 5.1: Summer off-peak Polish power grid. Sets C_2^k and S_2^k evolution with progression of the algorithm. $C_2^{final} = 463$

k	Size of the set C_2^k	Size of the set S_2^k
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 5.2: Winter off-peak Polish power grid. Sets C_2^k and S_2^k evolution with progression of the algorithm. $C_2^{final} = 928$

k	Size of the set C_2^k	Size of the set S_2^k
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

Table 5.3: Winter peak Polish power grid. Sets C_2^k and S_2^k evolution with progression of the algorithm. $C_2^{final} = 15882$

Thus, the simulations for both off-peak Polish grid cases demonstrate impressive efficiency of the algorithm. 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. The factors allowing such effective pruning will be discussed in the next chapter. Meanwhile, in order to further analyse the performance of the algorithm a significantly more loaded "peak" Polish case was investigated.

5.2.2 Winter peak case

The simulation results for the Winter peak Polish power grid case (*case2383wp* in MATPOWER package) are presented in the Table 5.3. This peak case is significantly more loaded than the off-peak cases studied above. As a consequence the algorithm gives a very high number of the dangerous non-islanding double outage contingencies $|C_2^{final}| = 15882$. The pruning loop of the algorithm was able to yield the output set of the size $|C_2^{filtered}| = 267681$, filtering out 89% of candidates in the first two iterations. On the one hand this result is not as impressive as for the off-peak cases. However, on the other hand, the efficiency of the algorithm is given not only by the size of the $C_2^{filtered}$, but rather by the combination of its size and the ratio $|C_2^{filtered}|/|C_2^{final}|$. This ratio is equal to ~ 17 for the Winter peak case, which is even better than the same ratio for the Winter off-peak case. It suggests that the efficiency of the pruning loop should remain approximately the same as the grid becomes more stressed. This assumption will be studied further in the next section.

The simulation results for the three Polish grid cases studied above are summarized in the Table 5.4. As can be seen from the table the completion time of the pruning loop remained very low compared to the brute force enumeration running time for all three cases. This can be explained by the N_L quadratic complexity of the pruning loop. Meanwhile, the full search (Lines 11-20) completion time depends not only on the complexity of the pruning loop, but also on the complexity of the brute force enumeration over the $C_2^{filtered}$. As the grid becomes more loaded the size of the

$C_2^{filtered}$ grows and hence the full search time grows as well as can be noticed from the table. Thus the pruning loop itself shows very promising performance, filtering out 89% of contingency candidates in 0.3% of the brute force search completion time for the most loaded case.

5.3 IEEE 300-bus test case under different stress conditions

The *IEEE 300-bus* test case was used to examine how the algorithm performs as the load on a grid increases. Without an additional load the algorithm finds zero dangerous non-islanding $N-2$ contingencies for IEEE 300 case ($|C_2^{final}| = 0$). To perform stress analysis, first, the maximum possible load coefficient s^{max} was calculated. That is both maximum generation and consumption were increased (multiplied by s) until the OPF solver didn't converge. s^{max} found using this method is equal to 14.8. Then the grid was incrementally loaded starting from the unloaded state with $s = 1$ to the maximum stress condition defined by $s = s^{max}$. The $N-2$ analysis was performed at each step, computing the corresponding sizes of the sets $C_2^{filtered}$, C_2^{final} and running time of the algorithm (Lines 11-20). The results are presented in Fig. 5-1.

As can be seen from Fig. 5-1 the size of both the filtered set and the set of dangerous non-islanding contingencies increase exponentially as the load coefficient grows. Completion time of the algorithm also shows approximately exponential growth. It should be noticed, that the noise in the running time dependence might have been associated with variations in background tasks on the machine during simulations. Running time of the direct brute force enumeration over the set C_2^1 was almost constant for all stress coefficients and equal to $\sim$ 30 seconds. This shows that the developed algorithm is significantly more efficient than the brute force approach even for high stress conditions.

Case	$\lvert C_2^1 \rvert$	$\lvert C_2^{filtered} \rvert$	$\lvert C_2^{final} \rvert$	Lines 11-16 completion time, sec	Lines 11-20 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	$1.9 \cdot 10^4$

Table 5.4: 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

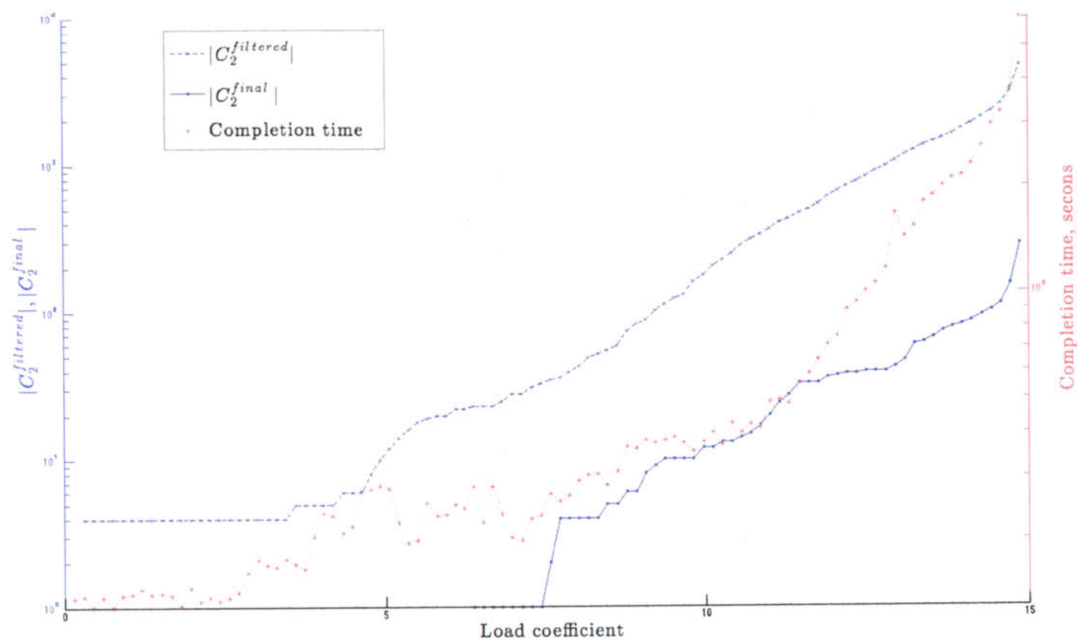

Figure 5-1: $|C_2^{filtered}|$, $|C_2^{final}|$ and completion times of the algorithm. 100 simulations with different values of the load coefficient $s \in [1; s^{max}]$

The dependence of the ratio $C_2^{filtered}/C_2^{final}$ on the stress was studied to further analyse the efficiency of the algorithm. Fig. 5-2 illustrates that the pruning loop performance ratio stays in the range between 10 and 35, which is consistent with the results obtained for Polish grid cases. The dependence of the pruning loop performance ratio has descents and ascents along the stress coefficient range. This can be explained by the structure of the bounding matrices used in the algorithm. The size of the $C_2^{filtered}$ increases when some elements of bounding matrices exceed 1, whereas the size of the C_2^{final} increases only when the some of the contingencies actually becames dangerous. It is worth to be noticed that in the limit $s \to \infty$ the ration is expected to approach 1 since both the size of the set $C_2^{filtered}$ and C_2^{final} will approach the size of C_2^1.

Fig. 5-2 also shows completion time of the full search (Lines 11-20 in the Algorithm 1) and just the pruning loop (Lines 11-16 in the Algorithm 1). The fact that running time of the pruning loop is approximately constant for any stress coefficient suggests that the pruning loop itself is efficient for any load condition. This supports the reasoning in the previous section and establishes some empirical guarantee for the performance of the iterative pruning loop.

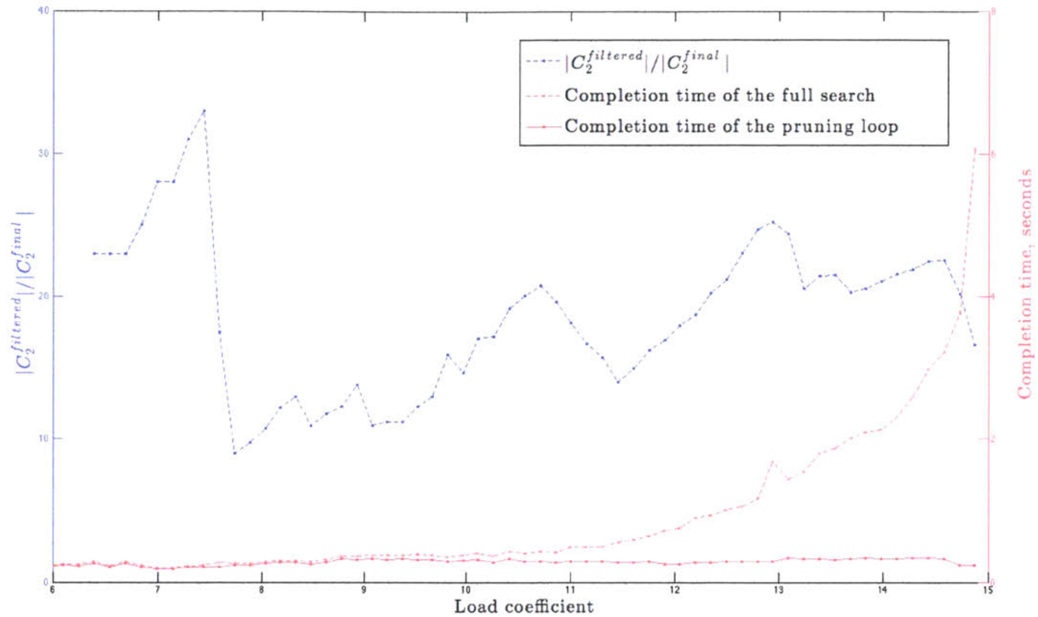

Figure 5-2: The efficiency of the pruning loop $|C_2^{filtered}|/|C_2^{final}|$ and completion times of the pruning loop (Lines 11-16 in the Algorithm) and the full search (Lines 11-20 in the Algorithm). Simulations with different values of the load coefficient $s \in [1; s^{max}]$

The next chapter will discuss the reasons behind the effectiveness of the algorithm and present some physical intuition.

Chapter 6

Analysis of the results

As the efficiency of the developed in Chapter 4 algorithm has been shown in the previous chapter, the reasons behind its performance are to be investigated further. This chapter discusses the simulation results and aims to build some physical intuition around the connection of the performance of the pruning loop to the actual structure of the power grid.

Discussion in this chapter will evolve around Polish Summer off-peak case. However, any case can be analysed in the same manner.

6.1 LODFs distribution

Line Outage Distribution Factors are the crucial part of almost any relationship derived in this work. Chapter 3 showed that LODFs give the relative influence of the tripping of one particular line α on the remaining lines in the grid. As was discussed in Chapter 1 a number of contingency selection approaches have been developed using just information about LODFs to predict the severity of contingencies. Despite the fact that approaches based on solely the LODF data do not account for information about the power flows closeness to the lines constraints, quite a number of them have shown successful results and are currently used in the real life application. All these show the importance of studying the LODFs further as they are in the essence of the contingency analysis.

Particularly the distribution of d^α_β directly affects the performance of the algorithm developed in this study. The intuition behind this is rather straightforward: the smaller the components of the line outage vector d^α, the more contingencies containing line α may potentially be filtered out during the pruning stage of the algorithm. The Figure 6-1 presents the distribution of the absolute values of LODFs in the Polish off-peak Summer case. The distribution is given for the pairs from the set of non-islanding candidates C^1_2 as considering islanding double outages would give the distribution an additional peak at 1.

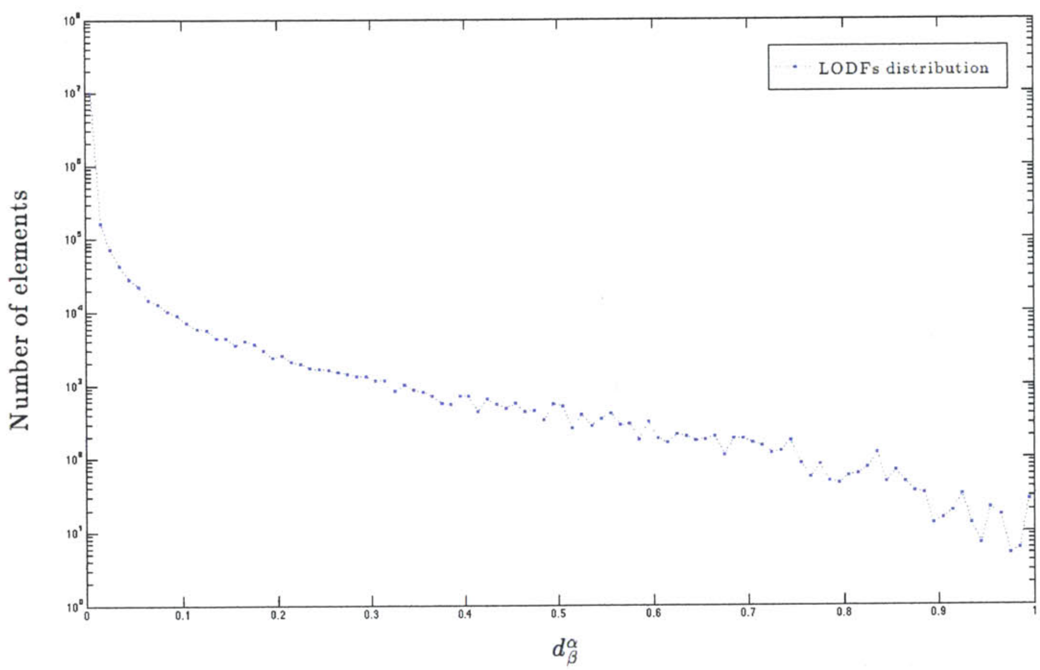

Figure 6-1: The distribution of Line Outage Distribution Factors $|d^\alpha_\beta|$ for the C^1_2 set for Polish off-peak Summer case. The horizontal axis shows the values of the elements. The vertical axis represent the number of the elements.

As can be seen from the figure the significant number of the line outage distribution factors do not even exceed 0.1 mark. In fact there is just 0.4% of pairs (α, β) for which $d^\alpha_\beta > 0.1$. This supports a fundamental assumption necessary for the algorithm to work efficiently. That is the outage of a single line affects only a very small percentage of the remainder lines in the system. This assumption is important for both influence ξ^α and interference $\Gamma_{\alpha\beta}$ components in the equations 3.14 to be efficiently bounded. While it is unproven for an arbitrary case the assumption holds for the power system models encountered during this study. Moreover, some theoretical insights can be obtained by considering local neighbourhood of the tripped line to estimate its influence [26].

6.2 ξ^α and $\Gamma_{\alpha\beta}$ distributions

As was discussed in the Chapter 3 the $\Gamma_{\alpha\beta}$ can be interpreted as the interference of the line β on the line α when both lines are tripped. The assumption that most of the lines do not affect each other directly leads to the intuition behind the form of the distribution of the elements $\Gamma_{\alpha\beta}$ (Figure 6-2). Indeed, for the case of small interference between lines α and β the $\Gamma_{\alpha\beta}$ and $\Gamma_{\beta\alpha}$ should be close to 1.

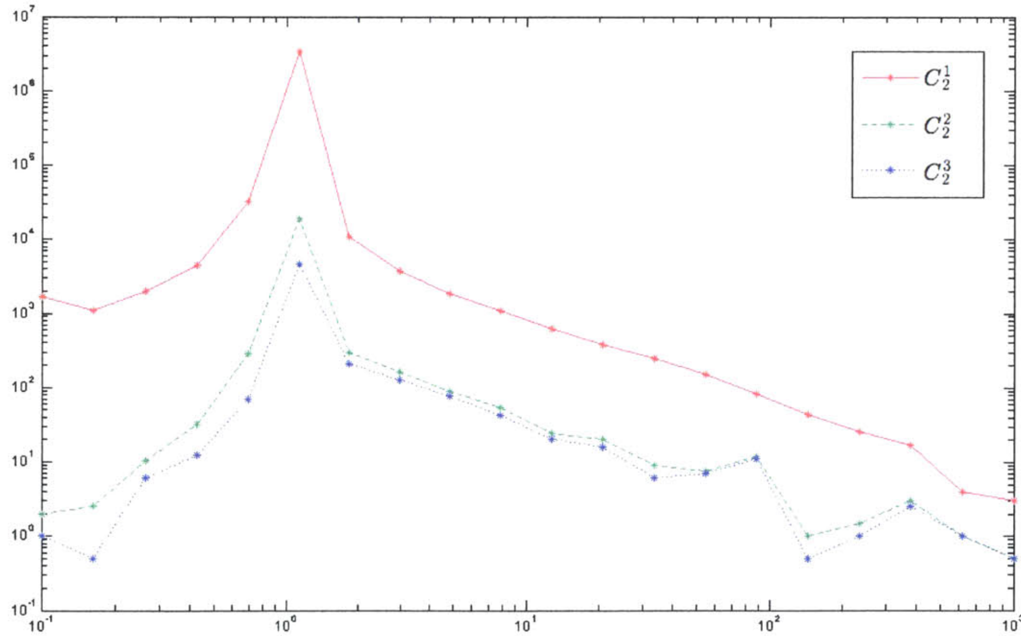

Figure 6-2: The distribution of the elements $\Gamma_{\alpha\beta}$ for the sets C_2^1, C_2^2 , C_2^3 for Polish off-peak Summer case. The horizontal axis shows the values of the elements. The vertical axis represents the number of the elements.

The Figure 6-2 also shows the evolution of the distribution as the algorithm proceeds. It can be seen that the pruning operations have more significant effect on the left part of the distribution. This is an expected property as the corresponding candidates are likely to have small bounds.

The vector ξ^{α} was interpreted in the Chapter 3 as the influence vector. Its absolute value distribution is presented in the Figure 6-3. Since elements ξ^{α}_{β} are proportional to the LODFs most of them are concentrated near zero. The distribution has a flat profile in the log-log scale which indicates the power law dependence. This property might be linked to the power law distribution of large blackout sizes [31, 32] since the elements ξ^{α}_{β} contain information about both the relative influence (LODF) and closeness to the limits $(\text{diag}(f^{max} - f)^{-1}; \text{diag}(-f^{max} - f)^{-1})$.

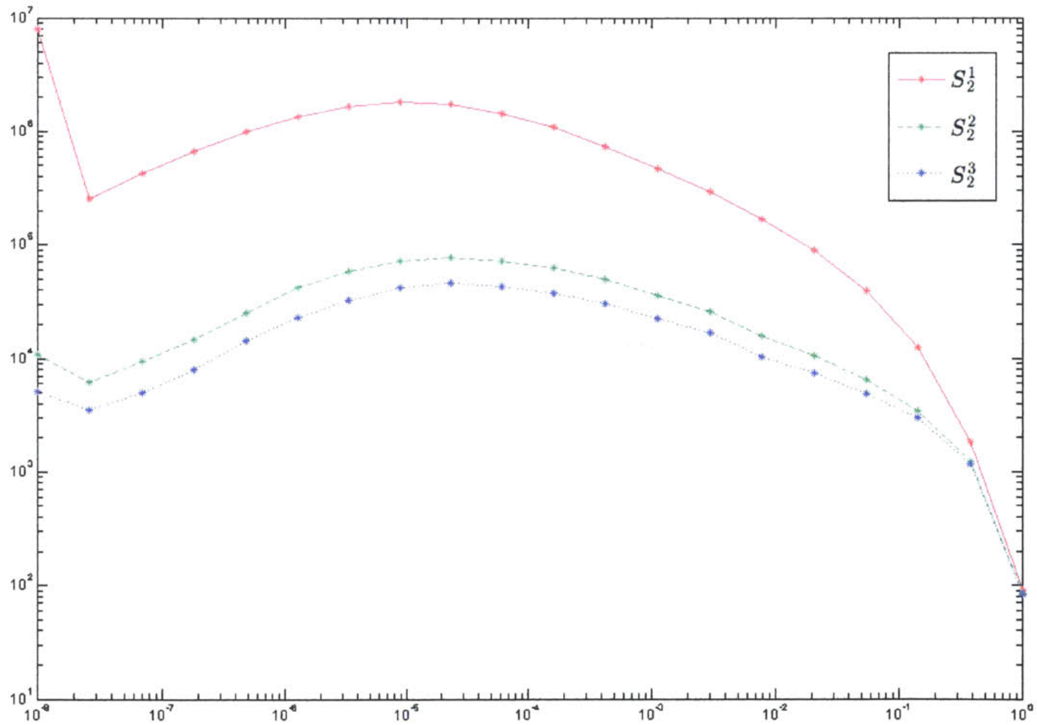

Figure 6-3: The distribution of the elements ξ^{α}_{β} for the sets C_2^1, C_2^2 , C_2^3 for Polish off-peak Summer case. The horizontal axis shows the values of the elements. The vertical axis represents the number of the elements.

The form of the distributions of ξ^{α}_{β} and $\Gamma_{\alpha\beta}$ suggests the reasons for the exceptional efficiency of the developed algorithm. First, outages of the most lines do not interfere with each other (most of the elements of $\Gamma_{\alpha\beta}$ are concentrated around 1). Secondly, the effect of an arbitrary single line outage usually is small compared to the flow limit margin on the affected line. This allows to produce rather tight bounding matrices.

6.3 Distribution of the elements in the bounding matrices

The efficiency of the algorithm can also be studied using the distribution of the elements in the bounding matrices $U_{\alpha\beta}^{C_2^k}$ and $U_{\alpha z}^{S_2^k}$ (Figure 6-4 and Figure 6-5). These distributions explain how the bounding matrices improve as the algorithm proceeds. As can been seen from the both figures the algorithm filters out small elements in the bounding matrices converging to the distribution on the right side of the dotted line (which depicts the pruning boundary).

On each iteration of the pruning loop (Lines 11-16) all elements on the left side of the dotted line (< 1) are filtered out. However, the new pruned sets produce tighter bounds and populate new elements that are smaller then 1. The number of elements on the left side of the dotted line decreases as the algorithm goes on creating the "cliff" near the pruning boundary. The bounding matrix for the set S_2^1 has significantly more elements that are > 1 than the corresponding matrix for the set C_2^1. This gives the intuition about substantial difference in the pruning efficiency of these sets.

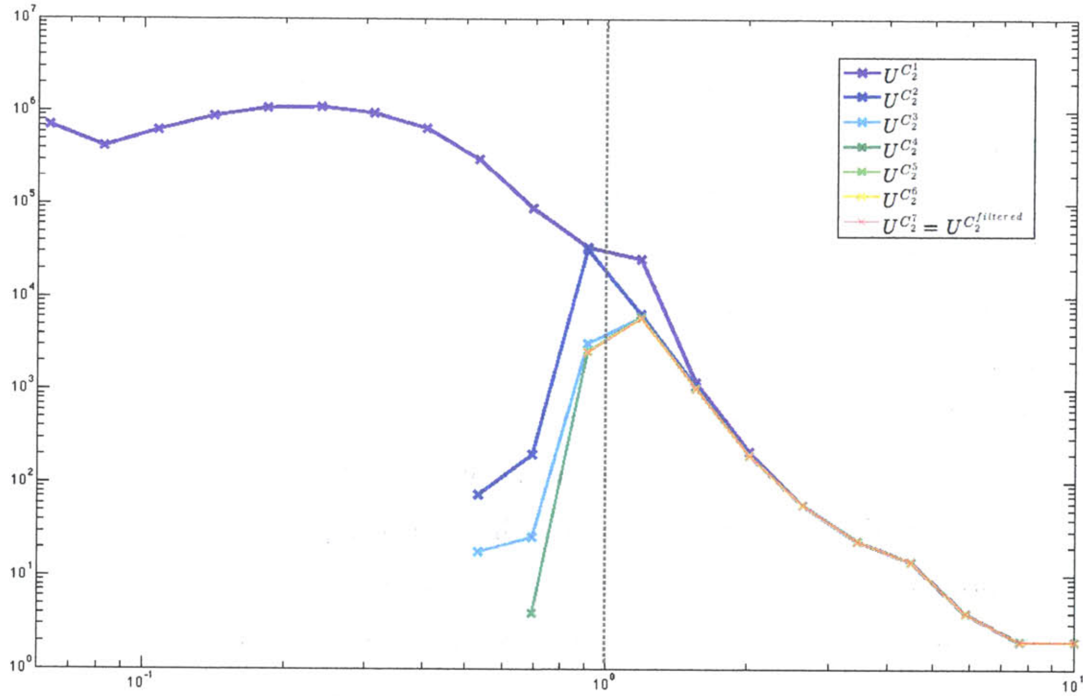

Figure 6-4: The distribution of the elements in the bounding matrix U^{C_2} for the sets C_2^k for Polish off-peak Summer case. The horizontal axis shows the values of the elements. The vertical axis represents the number of the elements. The dotted line is drawn at 1 and depicts the pruning boundary

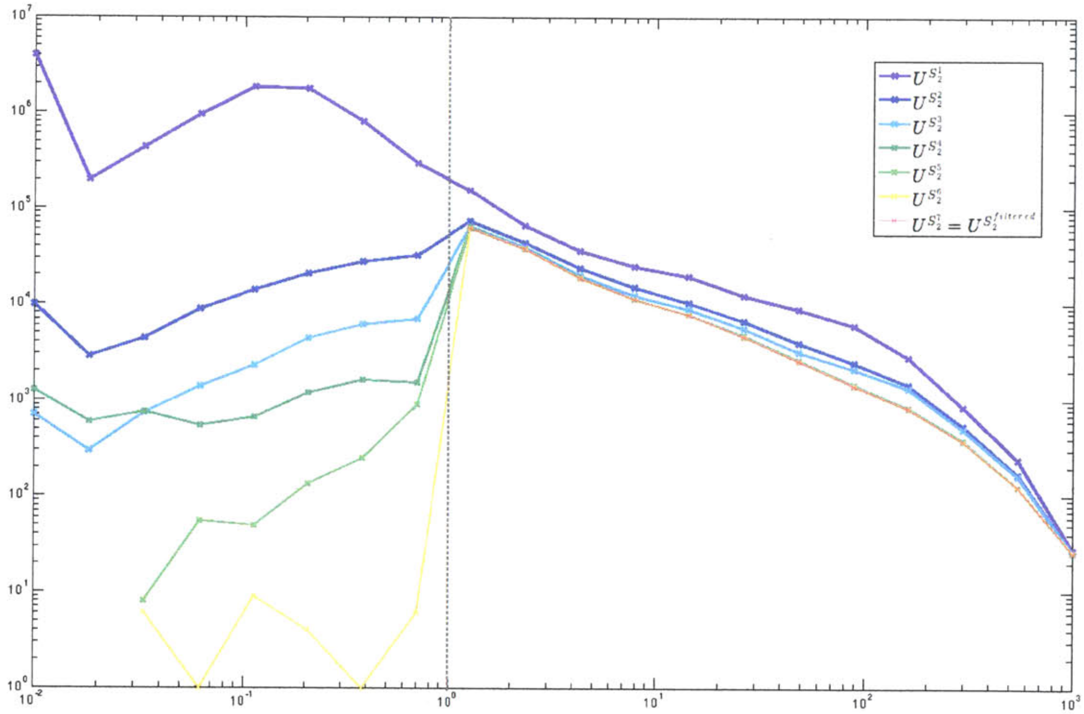

Figure 6-5: The distribution of the elements in the bounding matrix U^{S_2} for the sets S_2^k for Polish off-peak Summer case. The horizontal axis shows the values of the elements. The vertical axis represents the number of the elements. The dotted line is drawn at 1 and depicts the pruning boundary

Chapter 7

Conclusion

The novel approach for $N-2$ contingency selection has been presented. The developed algorithm is based on the idea of iterative pruning of the candidate sets. Only potentially dangerous contingencies remain on each iteration as the algorithm proceeds. Unlike many other approaches, the developed algorithm is not heuristic as it guarantees to find all dangerous double outages.

The algorithm filters out the set of potentially dangerous contingencies in just $O(N_L^2)$ operations if the matrix of Line Outage Distribution Factors is given. Thus the complexity of the pruning loop of the algorithm is substantially smaller than the complexity of brute force enumeration over the full set of the double outages. The latter would require $O(N_L^3)$ operations.

The algorithm has been validated and tested on the three Polish power grid cases. The pruning loop of the algorithm in just 2 iterations has managed to reduce the size of potentially dangerous non-islanding double outages by 99.9%, 99.1% and 89% for Summer off-peak, Winter off-peak and Winter peak Polish cases correspondingly.

It has been shown that the complexity of the full search depends on the size of the output set after K pruning iterations and is equal to $O(KN_2^2 + N_L|C_2^{filtered}|)$. While the size of the $C_2^{filtered}$ is expected to be small for the most real cases, it might increase substantially for extremely stressed systems. However, the ratio of the size of the iterative loop output set to the size of the final set of dangerous non-islanding $N-2$ contingencies has empirically established to remain of the order of $10-30$.

This gives empirical guarantee for the performance of the algorithm for even highly loaded cases.

In cases when the full search is infeasible for some reasons the pruning loop of the developed algorithm can be used as a pre-filter for any consequent contingency selection approach based on DC approximation as it decreases the size of the candidates set without any significant computational overhead to $N-1$ contingency selection.

It has been shown that the primary reason for an impressive performance of the algorithm is twofold. The first part of it is the fact that the power flows on the most lines are far from their limits compared to an average single line outage influence. The second part is that outages of the most lines almost do not interfere with each other. While this does not hold for a general system, it is expected to be the case for any real grid and has been shown to be true for all cases studied in this work.

Although the effectiveness of the approach is quite impressive, there is a number of ways one can proceed to improve and generalize it.

1. **Optimization**. The optimization directions have to be studied further. Among them are technical implementation of the algorithm, divide and conquer method to decrease the size of the output set as well as the degree of freedom in the representation of the contingency equations, that can improve bounding matrices.

2. $N-k$ **generalization**. It is possible to generalize the presented approach to the case of $N-k$ contingencies, where $k \geq 2$. This would require reconsideration of the bounding process as the governing equations for $N-k$ contingencies have very similar form. It is expected that whenever the number of dangerous contingencies is small one will be able to produce sufficiently tight bounds to prune the candidate sets.

3. **AC generalization**. It should be feasible to extend the approach to the full AC model. Since the presented algorithm is based on bounding one might

find an efficient way to bound different terms in the full AC equations without solving them in closed form. Moreover, one might successfully utilize the fact that power systems are usually operated in a weakly nonlinear regime.

Bibliography

[1] Western Electricicty Coordinating Council, Operating committee handbook, III-119, Revised, 2005.

[2] USCA, "Final Report on the August 14, 2003 Blackout in the United States and Canada," Tech. Rep. US-Canada Power System Outage Task Force, 2004.

[3] UTCE, "Final Report System Disturbance on 4 November 2006," Tech. Rep. Union for the Co-ordination of Transmission of Electricity, 2007.

[4] India Ministry of Power, "Report of the Enquiry Committee on Grid Disturbance in Northern Region on 30th July 2012 and in Northern, Eastern & North-Eastern Region on 31st July 2012," Tech. Rep., 2012.

[5] EATON, "Power Outage Annual Report," Tech. Rep. Blackout Tracker, 2012.

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

[7] F. F. Alvarado, "Computational complexity in power systems," *Power Apparatus and Systems, IEEE*, vol. 95, no. 4, pp. 1028–1037, Jul. 1976.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] M. Halappanavar, Y. Chen, R. Adolf, D. Haglin, Z. Huang, and M. Rice, "Towards Efficient N-x Contingency Selection Using Group betweenness Centrality," *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*, pp. 273–282, Nov. 2012.

[17] J. Yan, S. Member, Y. Zhu, H. He, S. Member, and Y. L. Sun, "Multi-Contingency Cascading Analysis of Smart Grid Based on Self-Organizing Map," vol. 8, no. 4, pp. 646–656, 2013.

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

[19] 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.

[20] H. Mori and Y. Goto, "A tabu search based approach to (n-k) static contingency selection in power systems," in *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, vol. 3, 2001, pp. 1954–1959 vol.3.

[21] 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 *PowerTech, 2009 IEEE Bucharest*, 2009, pp. 1–8.

[22] J. Thorp, "An importance sampling application: 179 bus WSCC system under voltage based hidden failures and relay misoperations," *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, vol. 3, pp. 39–46, 1998.

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

[24] M. J. M. Eppstein and P. D. H. Hines, "A Random Chemistry Algorithm for Identifying Multiple Contingencies that Initiate Cascading Failure," *IEEE Transactions on Power Systems*, no. 99, 2011.

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

[26] F. Galiana, "Bound estimates of the severity of line outages in power system contingency analysis and ranking," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-103, no. 9, pp. 2612–2624, 1984.

[27] V. Brandwajn, "Efficient bounding method for linear contingency analysis," *Power Systems, IEEE Transactions on*, vol. 3, no. 1, pp. 38–43, 1988.

[28] B. Stott, J. Jardim, and O. Alsac, "DC Power Flow Revisited," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1290–1300, Aug. 2009.

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

[30] 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," pp. 12–19, 2011.

[31] P. Hines, K. Balasubramaniam, and E. Sanchez, "Cascading failures in power grids," *Potentials, IEEE*, vol. 28, no. 5, pp. 24–30, 2009.

[32] P. Hines, J. Apt, and S. Talukdar, "Trends in the history of large blackouts in the united states," in *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, 2008, pp. 1–8.