Analysis of worst case Power system contingencies using vulnerability frontier

by

Vikas Dawar

A thesis submitted in partial fulfillment of the

requirements for the degree of

Master of Science

(Electrical Engineering)

at the

UNIVERSITY OF WISCONSIN-MADISON

2010

Abstract

One of the most important aspect in a power system network is to make a network secure against unanticipated events. In general, a power system network is always made N-1 secure: any single outage will not lead to instability in a network. The electrical transmission system is vulnerable to extreme events which can cascade resulting in blackouts. In this thesis, we identify critical corridors, subsets of those transmission lines that are most vulnerable to extreme events. It is computationally difficult to take into account all possible multiple outages of lines. In this thesis, we consider the worst case scenario in which the outages of a particular subset of transmission lines result in maximum power imbalance. Subsets of transmission lines that appear with higher frequencies are candidates for detailed study as they may lead to extreme events. We present a vulnerability frontier (graph between number of lines cut and maximum power imbalance corresponding to those lines cut) for 30, 118, 179, 225 and 1553 bus system. Tables listing the frequencies at which a particular cut-set of transmission lines (for 30,118 and 179 bus system) occur have also been presented.

Acknowledgements

I want to thank my advisor Professor Bernie Lesieutre, for giving me the opportunity to pursue my research under his able guidance. I thank him for continuously supervising me throughout the period of my Masters.

I want to thank Professor Ian Dobson to help me through my research work. I also want to thank Professors Christopher DeMarco, B. R. Barmish for teaching me and providing me the tools to tackle my research problem. I want to thank my friends Rahul Srinivasan, Rishi Srivastava, and all members of the power system group for supporting me throughout.

Last but not least, I want to thank my family, my Mom (Kanta Dawar), my Father (Harish Dawar) ,my sister (Saru Dawar) and Deepika Ganju for encouraging me to pursue higher studies and for always being with me.

I would also like to thank the California Institute for Energy and Environment for supporting the research.

Table of Contents

List of Figures and tables

Chapter 1 Introduction

The electric power industry works hard to avoid blackouts, and there are many practical methods to maintain reliability. It is possible to make power system $N-1$ secure through existing technologies. But, still there is no method available to make power system secure against multiple outages. As a recent example, the August 14, 2003 blackout in the northeast of the U.S. resulted in a loss of estimated 61.8 GW of electric load and affected 50 million people [1]. The cost associated with this blackout was about $6 billion as estimated by the U.S. Department of Energy (DOE) [2]. While many factors contributed the prevailing conditions on that afternoon, three transmission lines that underwent faults and subsequent outages in relatively short succession initiated the blackout process. Although these types of extreme events are infrequent, historical statistics have shown that they do occur. The identification of lines, removal of which will lead to power system infeasibility is crucial in terms of power system security.

To make power system $N-k$ secure means that no combination of k facility outages will cause a widespread blackout. The brute force method to test k components involves choosing all combinations of k outages and testing whether a blackout might occur. This is computationally impractical. The computation grows combinatorially as the size of the system increases. For example if in a system we want to test all combinations of 4 lines out of 10,000, we have to carry out power flow for all possible combinations choosing 4 out of 10,000, that is approximately $4 * 10^{18}$ cases. This exceeds the capability of even super computers.

The goal of this thesis is to explore worst case cut-sets of transmission lines whose outage will cause maximum power imbalance in a system given the power injected by the generators and the power absorbed by the loads. This maximum power imbalance represents an estimate of the severity of an event relating to the outage. In this thesis we view the power system network as a graph, with buses as nodes and lines as edges. We pose a dual objective optimization problem to find minimum number of lines that will be out of service and to maximize the corresponding power imbalance. We explore the trade-off between the objectives by constructing a "vulnerability frontiers" that bounds the maximum power imbalance as a function of line outages. We show that optimization problem can be formulated as the well known maximum flow minimum cut problem in graph theory. In our work we introduce vulnerability frontier to take into account details of occurrence of events and their severity. Cut-sets of transmission lines corresponding to points in a frontier appear in notable patterns, and tables representing the frequency at which they occur form an integral part for future work. Results for 30 bus system, 118 bus system, 179 bus system, 225 bus system and 1553 bus system has been shown.

1.1 Literature review

Previous work done by Vaibhav Donde, Vanessa López, Bernard Lesieutre, propose a two-stage screening approach to identify severe multiple contingencies [3]. Their paper identifies severe multiple contingencies that may result in very severe disturbances and blackout. Rather than exhaustive brute force technique of removing lines and then solving power flow to find whether solution exists or not, their paper discusses a more practical approach to find multiple contingencies. It advocates a two-stage process to identify multiple contingencies, first lowering down the search to small number (user defined) of lines that cause maximum severity, and then perform detailed analysis on these small numbers of lines. Given the severity of disturbance it poses an optimization problem to find fewest lines cut and subsequent subsets of these lines cut are analyzed to figure out all possible $N-k$ contingencies which are then ranked according to their severity. That work is based on changing the line characteristics such that the power flow solution just becomes infeasible. The line characteristic corresponds to outage of a line or a line reaching its steady state limit. Both these characteristics affect the power flow solution and may result in infeasibility of a power system. For initial screening partial outages of lines are allowed. An optimization problem is introduced. The optimization problem gives the fewest lines cut given the severity function. For more detail analyses refer to "Identification of severe multiple contingencies in electric Power Systems" [3]. Their approach drastically reduces the number of lines to be combinatorially searched for a power flow solution. As an example for 1000 lines system, initial screening will only give 100 lines for given severity and then for say $N-3$ contingency we only have to choose 3 lines out of 100 lines rather than 1000 lines thereby allowing brute force to find all subsets of these 3 lines.

Another previous works to find multiple contingencies, by Qiming Chen, James D. McCalley [4] analyzes $N-k$ contingency according to the probabilities of their consequences. The $N-k$ contingency in the substation can be due to the failure of the protection system or breakers tripping more than one line when clearing the faults. In their paper a substation is divided into functional groups (these are the groups of all components in a substation separated by the circuit breaker or by open switches). The utility presents the data representing the list of all components and probability of the failure of these components in a functional group. The data from the utility also presents the list of all circuit breakers, switches and respective probabilities of their failure to operate. Probability rules are applied to find net probability of a particular contingency. Contingencies can be because of the failure of the components in a particular functional group which results in tripping of all the components of those functional groups, only if the circuit breakers are able to isolate that functional group. Contingencies can also be caused due to the failure of the circuit breaker to operate which results in tripping of all the components of functional groups corresponding to that circuit breaker. Their paper gives the algorithm for FG Decomposition (Functional Group Decomposition), which divides the substation into functional groups and probabilities of contingencies is then evaluated. The approach is tested and results are presented in [4]. Only those $N - k$ contingencies are important to be realized that have more probability to occur. So for an N-4 contingency, rather than searching for all possible 4 line removal choices out of N lines we only consider those 4 lines removal important that have high probability. This paper applies a graph theory approach to represent the substation as a graph network with bus sections as nodes and all the components as edges. Probabilities of subsequent events is evaluated (where subsequent events refer to an initializing fault followed by a series of events such as failure to open a circuit breaker), and components loss due to any contingency can be seen as the measure of severity.

In the paper by J.Z. Zhu, Prof. G.Y. Xu, [5] they propose an algorithm to rank the contingencies severity according to some performance index based upon the reactive power flow. The contingency such as a line outage will result in variations in bus voltages and hence variations in reactive power flow. If we tend to sustain the reactive power requirements of all loads even after outage, the voltages at some buses and reactive power flow at some branches will violate their constraints. Hence the performance index to rank the severity of a particular contingency can be given as:

$$PI = \frac{(Qd^{\circ} - Qdk)}{Qd^{\circ}}$$

where PI is the performance index, Qd^0 is the total reactive load in the pre-contingency state and Qdk in the reactive power flow in load branch k after a line outage. They propose a reactive optimization problem, to minimize the voltage drop across the transmission branch subjected to constraints, to limit the reactive power flow across generator branch, transmission branch and load branch and conservation of flows at each node. A particular line outage can be imitated in the constraint such as to have zero reactive power flow across that line. Hence, the solution can be used to calculate performance index for each contingency. Their paper does not find multiple contingencies but proposes a method to measure the severity of a particular contingency (if identified) and rank all possible contingencies. Since there can be large number of contingencies which can be critical, this paper proposes to find only those contingencies that are severe enough with severity defined by user. This is implemented by setting the lower and upper bounds of the reactive power flow appropriately that rules out all contingencies that are not severe.

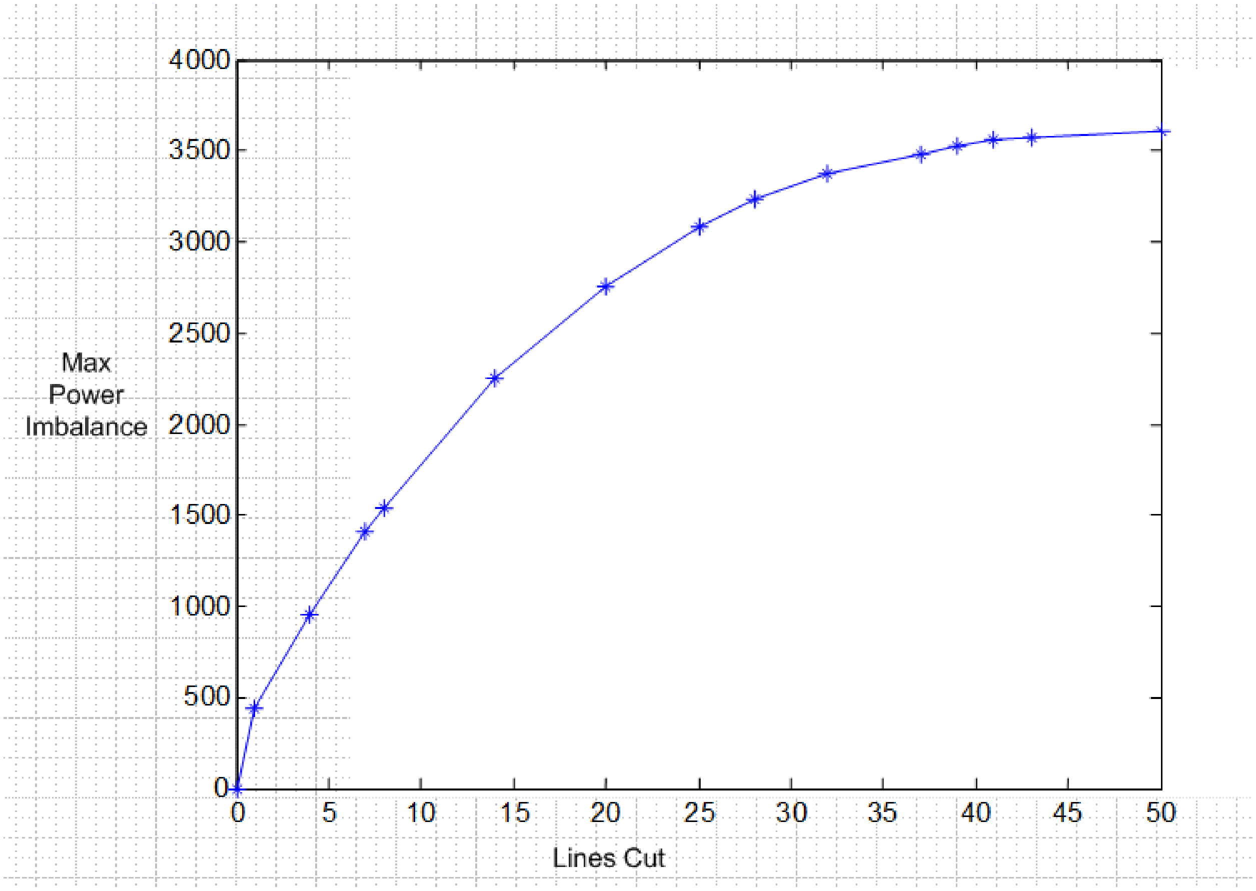
The paper by Ahmed M.A. Haidar, Azah Mohamed and Aini Hussain [6] proposes artificial neural network technique to assess how vulnerable is the power system. There are multiple contingencies such as generator outage, load outage or line outage to which the power system is vulnerable. The power system vulnerability is measured by the Vulnerability Index. The vulnerability index which is based on power system loss considers total system loss, generation loss due to generation outage, power line loss due to line outage, increase in total load and amount of load disconnected. The rational for considering PSL is due to the fact that losses in a power transmission system are a function of not only the system load but also of the generation. In addition, each contingency has an effect not only on the system performance but also on power losses in the system. The outage of transmission line, transformer or generator may result in line overload and cause increased active power loss in lines and reactive power loss in transformers. A similar effect may result if a contingency such as loss of load is said to occur. Therefore, it is important to consider total power system loss as a quantitative measure for assessing vulnerability of power systems. The vulnerability index is the measure of the power load loss. To maintain the integrity of the power system in case of the cascading failure it is required to shed some amount of load in order to sustain reliability of some important parts in the system. ANN (Artificial Neural Network) is used for calculating the vulnerability index in the proposed paper by Ahmed M.A. Haidar, Azah Mohamed and Aini Hussain [6]. The methods of ANN radial basis function and back propagation is used to assess the vulnerability of system. The first step before applying ANN is to collect as much data as possible from the power system, in which the data is assumed to be of physical interest for vulnerability assessment. The data can be obtained from vulnerability analysis simulations carried out on the test system. The procedures involved in power system vulnerability assessment begin by first analyzing the system behavior at the base case condition. The next step is to analyze the system behavior when subjected to credible system contingencies by considering several test cases such as line outage, generator outage, and increase in total load and amount of load disconnected. The vulnerability indices PSL and PLL are then calculated for each test case. So, the input features for ANN are reactive and active power flows in the power system, reactive and active power generations in the power system. The outputs of the ANN are the vulnerability indices. With training data set chosen as mentioned before, the authors in [6] train ANN. Then this trained ANN can be used on more complex power system. A new technique of feature reduction is also mentioned to cut down the input features of ANN, so as to reduce the complexity burden. For detail of analysis of the features reduction algorithm refer to [6].

The paper by Javier Salmeron, Kevin Wood, and Ross Baldick [7] proposes a bi-level optimization technique to minimize the generation cost and the cost of load shedding, and maximize the disruption from point of view of the terrorists with given amount of resources. The min max problem proposed in the paper deals with the above aspects. The algorithm given in the paper identifies the critical components (generators, transmission lines) in a power system by creating the maximum disruptive plans by the terrorists. In brief, an interdiction plan is represented by the binary vector, whose i^{th} entry is 1 if component of the system is attacked and is 0 otherwise. An initial DC-OPF (DC Optimal Power Flow) problem is solved for a particular set of the binary vectors depicting the disruptive plan by the terrorist. The solution of the DC-OPF will yield power outputs of generators and power flow along transmission lines such as to solve for the min problem minimizing the cost of generation and cost for load shedding. This vector set of power flows will be used to measure the attractiveness or estimated values of components of power system. These estimated values are then used to solve for a new interdiction plan to maximize the disruption caused by a terrorist. With this new interdiction plan a DC-OPF is solved again to yield new vector set of power flows. This iterative process converges to a feasible and an optimum value of an interdiction plan and cost of generation and cost of load shedding. For detail descriptions of the algorithm and results refer to [7].

The paper by Bernard et.al [8] proposes propose a computationally efficient method based on nonlinear optimization to identify critical lines, failure of which can cause severe blackouts. These critical lines are further analyzed for multiple contingencies. To measure the criticality of lines a concept of modified admittance is introduced. Admittance of each line is multiplied by a factor γ which can take value between 0 and 1. When γ is 0, admittance of line is zero and line is out of service. When γ is 1, admittance of line is 1 and line is operating at its full capacity. An optimization problem is introduced to minimize γ for all lines subjected to solve power flow. All the lines having γ close to 1 will be the most critical lines since they will be operating at their full capacity subjected to solution of power flow. Brute force technique to solve “$N - k$” contingencies for given power system is computationally exhaustive. Solution to an optimization problem reduces the number of lines and only combinations of these critical lines are analyzed for multiple contingencies. A Branch algorithm is introduced in the paper to find all possible multiple contingencies that are severe enough to cause blackout. Results of 30 bus system and 118 bus system are shown to be consistent [8].

The paper by Ali Pinar, Juan Meza, Vaibhav Donde, and Bernard Lesieture [9] proposes a multilevel non-linear formulation for vulnerability analysis of electric grid, and then with help of spectral graph theory it is able to reduce the complexity to solve MINLP to a simple combinatorial problem. The basic strategy exploits the structure of Jacobian matrix with the Laplacian matrix in spectral graph theory. Formulation in MINLP involves converting min max problem of maximizing the severity of outage and minimizing the outages. This bi-level optimization problem is transformed into MINLP by replacing the maximizing problem of severity of outages by its KKT conditions, hence solving MINLP problem of minimizing the outages with specified severity of outages. This can also be viewed as a problem of maximizing the severity of outages given the specified number of lines. The structure of Jacobian matrix resembles the Laplacian matrix. The Laplacian matrix in graph theory helps to find the connected components of the graph. Hence, the Jacobian can be used to find the connectivity of the power system. When the operating point is on the feasibility boundary the multiplicity of zero eigenvalue of Jacobian matrix gives the measure of sub-graphs of power system. These sub-graphs can be viewed as decomposition of a system into a load rich region and generating rich region, and the paper proves that set of lines connecting these sub-graphs is always saturated. This is the maximum flow minimum cut problem in graph theory which minimizes the number of lines that are saturated and maximizes the mismatch between generation and load. Our thesis work is primarily based on this work and details of modeling this idea will be reviewed later in thesis.

We propose a vulnerability frontier to bind the maximum power imbalance as a function of lines cut for various test cases. As an example, vulnerability frontier for a 30 bus system is shown below.



The rest of the thesis is organized as follows. Chapter 2 gives a brief introduction to the graph theory along with detailed description of the max-flow/min-cut problem. Chapter 3 introduces our dual objective optimization problem of minimizing the number of lines which results in maximum power imbalance. Chapter 3 outlines the proof of equivalence between our proposed optimization problem and max-flow/min-cut problem. Chapter 3 also outlines a trade-off between minimizing the number of lines and maximizing the subsequent power imbalance. Vulnerability frontier is introduced to explore this trade-off. Chapter 4 presents vulnerability frontiers for 30 bus system, 118 bus system, 225 bus system, 179 bus system and 1553 bus system. Number of cut sets and their corresponding frequencies of appearance are tabulated. Chapter 5 concludes the thesis.

Chapter 2 Introduction to Graph Theory

This chapter gives brief introduction to graph theory. The purpose of this chapter is to introduce some important terminology that will be used in later chapters. Also this chapter will introduce and underline the max-flow/min-cut algorithm that will be used as primary tool for our thesis.

2.1 The node-arc Incidence and Laplacian Matrix

In mathematics and computer science, "graph theory is the study of graphs: mathematical structures used to model pair-wise relations between objects from a certain collection. A "graph" in this context refers to a collection of vertices or 'nodes' and a collection of edges that connect pairs of vertices. A graph may be undirected, meaning that there is no distinction between the two vertices associated with each edge, or its edges may be directed from one vertex to another" [10]. In a directed graph, bus incidence matrix is defined as a matrix in which each entry is either -1 , 1 or 0. Rows of the bus incidence matrix corresponds to number of edges in the graph, number of columns are equivalent to number of nodes in the graph. Let G = (V, E) be a graph with n vertices and m edges. We use (v_i, v_j) to denote an edge that goes from vertex v_i to vertex v_j. The node-arc incidence matrix, A, of this graph is an m×n matrix, where the j^{th} column of A represents the j^{th} vertex, v_j, and the i^{th} row represents the i^{th} edge, v_i, in G. Each row has two non- zero elements that represent the end vertices of the respective edge. The entry is -1 or 1, depending on whether the respective edge is directed from or to the corresponding vertex, respectively. Formally, we use a_{ij} to denote the matrix entry at the i^{th} row and the j^{th} column of A, which is defined as follows.

$$a_{ij} = \begin{Bmatrix} -1 & if\ e_i = (v_j, u) \in E \\ 1 & if\ e_i = (u, v_j) \in E \\ 0 & otherwise \end{Bmatrix}$$

In a spectral graph theory, the Laplacian matrix is used to measure the connectedness of graph. A graph is connected in the sense of a topological space such that there is a path from any point to any other point in the graph. A graph that is not connected is said to be disconnected. The Laplacian matrix of a graph $G = (V, E)$ is an n × n matrix, where each row and column represents a vertex in the graph. In case where weights are not associated with edges or weight of each edge is 1, the diagonal entry is equal to the degree of the associated vertex. In simple terms degree of a vertex is defined as the number of vertices connected to it. An off-diagonal entry is 1, if the associated vertices of the row and column are connected in the graph, and 0 otherwise [11]. Formally, let d_i denote the degree of a vertex v_i, and let l_{ij} denote the entry of the Laplacian matrix at the i^{th} row and the j^{th} column, which we define as follows.

$$l_{ij} = \begin{cases} d_i, & i = j \\ -1, & (v_i, v_j) \in E\ or\ (v_j, v_i) \in E \\ 0, & otherwise \end{cases}$$

We can also define the Laplacian matrix as $A^T * A$. It is possible that weights are associated with each edge. We account for the weights associated with each edge in the Laplacian matrix by replacing the diagonal entry with the sum of the weights associated with all edges connected to a referred vertex, and negative edge weight replaces -1 in an off diagonal entry. In our thesis all the weights associated with edges are 1 and so Laplacian Matrix is defined as $A^T * A$.

2.2 Max-Flow/Min-Cut Algorithm

In optimization theory, the max-flow/min-cut theorem states that, "In a flow network, the maximum amount of flow passing from the source to the sink is equal to the minimum capacity that needs to be removed from the network so that no flow can pass from the source to the sink" [12]. A flow network $G = (V, E)$ is defined by a set of vertices V, a set of edges E, a source s and a terminal t, where each edge (u, v) has a non-negative capacity $c(u, v)$. A flow in G is a function defined as f. We use $f(u, v)$ to refer to a flow on the edge from vertex u to vertex v. A single source and a single terminal vertex provide a standard form for the maximum flow problem. Even if there are multiple vertices with production, a single source vertex,s, is used that is connected to all other vertices with production and the capacity of the connecting edge is equal to the production on that node. Similarly, only a single terminal vertex,t, is used that is connected to all other vertices with consumption, and the capacity of the connecting edge is equal to the consumption on that node. We say a flow is feasible if it obeys conservation of flow and the capacity constraints on edges. Conservation of flow requires that the total flow into a node is equal to total flow out of that node, except for the source and terminal vertices. The value of a flow is defined by the total flow leaving the source, and the maximum flow problem aims to find a feasible flow with maximum value. As shown in the Figure 2.1, a graph with 6 nodes and 7 edges, maximum flow is equivalent to 10. Each edge below has flows and capacities represented on it as f/c, where f represents the current flow through that edge and c represents the capacity of that edge. The graph shown below also shows the duality between min-cut/max-flow. The edges that are saturated, that are the edges with $f = c$, form the min cut. And, max-flow/min-cut theorem states that the sum of capacities of these edges forming the min cut is equivalent to maximum flow that is possible from source to sink.

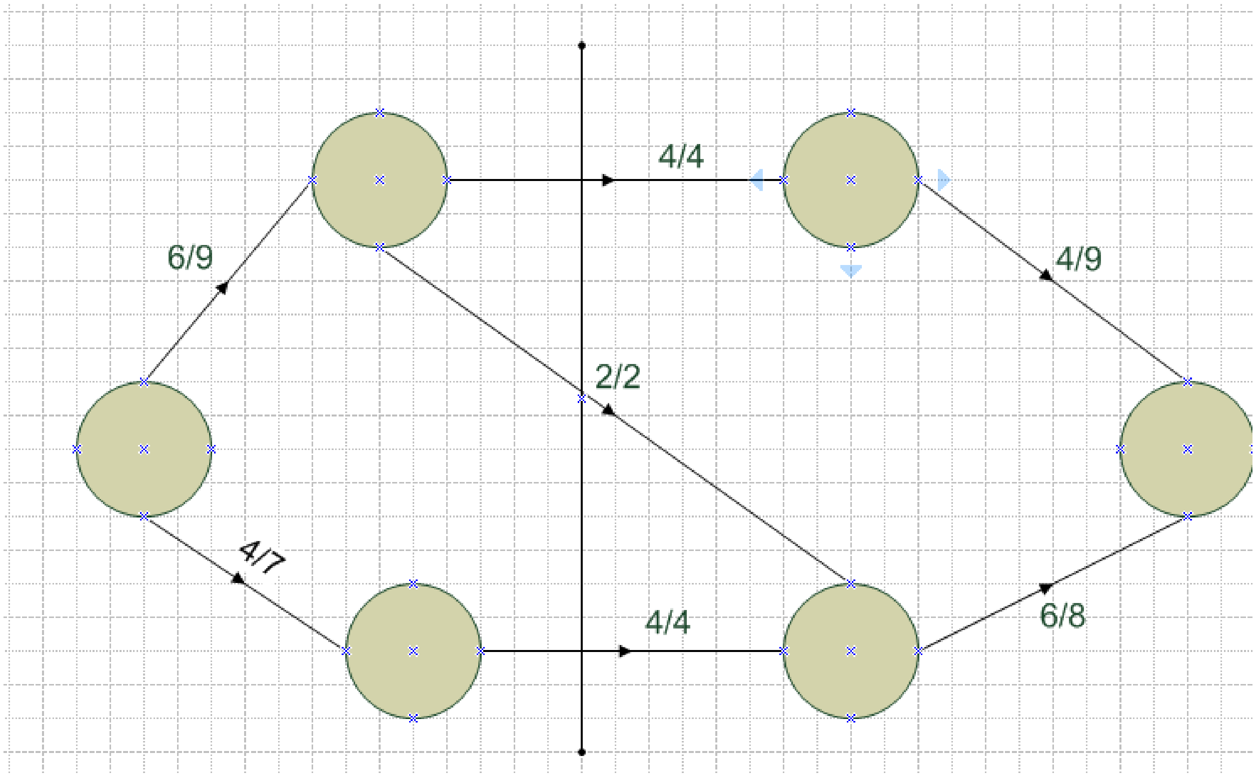


Figure 2.1 Graphical Example

The min cut as shown in the graph above is comprised of three edges, which tend to separate source and sink into two different groups. In more general sense, "a cut in a flow graph is defined by bi-partitioning vertices V into S and $T\ into\ V/S$, where $s \in S$ and $t \in T$" [13]. We say an edge is on the cut if one of its end vertices lies in S and the other lies in T. The capacity of a cut is defined as the sum of capacities of the edges on the cut, and a minimum cut is one with minimum capacity among all the cuts. Cut sets are the set of edges that if removed disconnects the graph. There are many max-flow/min-cut algorithms available. In our thesis work we use preflow push algorithm for max flow. We can consider our max flow problem as maximizing the flow out of our source node, alternatively maximizing the flow into sink. The preflow push algorithm [14] is basically based on maintaining a capacity feasible flow vector. We assume that the lower arc flow bounds for all edges are 0 and the upper bound c_{ij} is some positive number for$arc(i,j)$.

$$0 \leq x_{ij} \leq c_{ij} \quad \forall (i,j) \in A$$

Where x_{ij} is the flow across$arc(i,j)$. Preflow push algorithm use prices to guide the flow change and maintain a valid flow-price vector. To introduce preflow push algorithm [14] we will introduce certain terminology. Given a capacity feasible flow vector, x, the set of eligible arcs of i is $A(i,x) = \{(i,j) | x_{ij} < c_{ij}\ \} \cup \{(j,i) | 0 < x_{ij}\}$ and the corresponding set of eligible neighbors of i is $N(i,x) = \{j | (i,j) \in A(i,x)\ or\ (j,i\ ) \in A(i,x)\}$.

The candidate list of a node i is defined to be the set of its eligible incident $arc(i,j)$ or $arc(j,i)$ such that the price of node i=price of node $j + 1$, i.e. $p(i) = p(j) + 1$. A capacity feasible flow vector x together with a price vector $p = \{p(i)|\mathrm{i} \in \mathrm{N}\}$, where N is the number of nodes, are said to be a feasible or a valid pair if $p(i) \leq p(j) + 1 \quad \forall\, j\ that\ are\ eligible\ neighbors\ of\ i.$

Finally we introduce the term surplus of node i,

$$g(i) = \sum_{\{j|(j,i)\in A\}} x_{ji} - \sum_{\{j|(i,j)\in A\}} x_{ij}$$

In the above equation $g(i)$ is surplus of i. The preflow push algorithm starts with and maintains a valid flow price pair (x, p) such that x is feasible flow that is maintaining its constraints. Preflow push algorithm maintains the valid flow price pair at each step of iteration and the following relations hold true

$$g1 \leq 0, \qquad g(i) \geq 0 \forall\, i \neq 1, \qquad p(1) = N, \qquad p(N) = 0, \qquad 0 \leq p(i) \leq N, \qquad \forall i \neq 1$$

Initial condition for flow and price can be defined as below, where $\mathrm{c}_{1\mathrm{j}}$ is the capacity of $arc(1, j)$, we define node 1 to be our source node. Similarly we initialize the price of node 1 to be equal to N (where N are the number of nodes), and price of all other nodes to be zero.

$$p(i) = \begin{Bmatrix} N & if\ i = 1 \\ 0 & if\ i \neq 1 \end{Bmatrix}$$

$$x_{ij} = \begin{Bmatrix} c_{1j} & if\ i = 1 \\ 0 & if\ i \neq 1 \end{Bmatrix}$$

There are two types of operations in the pre-flow push algorithm [14]:

1) A flow change, which modifies the flow of some arc belonging to the candidate list of some node. The flow change is always in the direction from the node of higher price to the node of lower price.

2) A price rise increases the price of some node whose candidate list is empty. The increment of price increase is the largest value that satisfies the constraints mentioned above. With the price increment, the candidate list of the node becomes non empty.

The idea of the algorithm is to have a direct flow from nodes of higher price to the nodes of lower price. By setting the price of the source node to N and the price of sink node to 0, the algorithm moves the flow in the desired general direction from source to sink. At the start of each iteration of the preflow push algorithm, a node $i \neq N$, with $p(i) < N$ and $g(i) > 0$, is selected. If no such node can be found, the algorithm terminates. The typical iteration consists of the following steps [14]:

Step1: We select an $arc(i,j)$ of the candidate list of i and go to Step 2, or an $arc(j,i)$ of the candidate list of i and go to Step 3, and if the candidate list is empty, go to Step 4.

Step2: Push flow forward along $arc(i,j)$). Increase x_{ij} by $\delta = \min(g(i), c_{ij} - x_{ij})$. If now $g_i = 0$ and $x_{ij} < c_{ij}$ stop; else go to Step 1.

Step 3: Push flow backward along $arc(j,i)$. Decrease x_{ij} by $\delta = \min(g(i), x_{ji})$. If now $g_i = 0$ and $x_{ji} > 0$ stop; else go to Step 1.

Step 4: Increase price of node i. Rise $p(i)$ to the level shown below and goto Step 1.

$$p(i) = \min\{p(j) + 1 \big| (i,j) \in A \; and \; x_{ij} < c_{ij}, or \; (j,i) \in A \; and \; 0 < x_{ji}\}.$$

The algorithm terminates with a flow vector under which a minimum cut separating the source from sink is saturated. This flow vector is not necessarily maximum or even feasible, because some nodes other than the source and the sink may have a non zero surplus. But in our thesis work we are interested in minimum cut in order to get the indicator vector and solve the optimization problem, as will be defined in Chapter 3. We are basically interested in finding indicator vector x which defines the partitioning of graph in two parts. For demonstration of deducing the maximum flow from this flow vector refer to [14]. The preflow push algorithm terminates, and upon termination the flow vector X is such that there is a saturated cut $[N+, N-]$ with

$1 \in N+,\ N \in N-$

$g(i) > 0\ \ \forall i \neq 1\ with\ i \in N+$

$g(i) = 0, \forall i \neq N\ with\ i \in N-$

Chapter 3 Optimization Problem

In this chapter we address the challenge of finding all possible $N - k$ contingencies and will solve an optimization problem with help of graphical tools. We build upon the optimization problem as the chapter progresses. The main concern is that we cannot solve for "$N - k$" contingencies, because the problem tends to grow combinatorially as the complexity of the system grows if we do an exhaustive search to find all the possible N-k contingencies. So, rather than brute force approach to find all $N - k$ contingencies we will only find those contingencies that will have maximum severity. We represent the severity of an event by maximum power imbalance it causes. If an event refers to an outage of a line, then for a given amount of severity we need to find minimum number of lines cut that will cause the specified amount of severity. We can also view this problem as, for a specified number of lines cut we want to find the maximum amount of the load that can be shed to restore feasibility which is equivalent to finding the maximum power imbalance because of specified amount of lines cut. So basically we can pose an optimization problem of finding the minimum amount of lines such that it causes maximum amount of power imbalance. A trade-off variable c is associated with maximum power imbalance and minimum number of lines, this trade-off variable can be associated with explanation provided above in the paragraph. An algorithm for choosing value of c will be introduced later in the chapter. We can relate power system network to a graph in which buses represent nodes and lines represent the edges of a graph. In terms of the graph theory we can formulate an optimization problem as finding a min cut that will cause maximum power imbalance in a graph. From previous discussion we note that when an operating point is on the feasibility boundary the power system network is divided into two regions, generator rich region and load rich region. So in terms of graph theory the min cut will divide the network into the generating rich region and the load rich region. Our problem of finding the $N - k$ contingencies is to find the min cut such that there is maximum power imbalance between two regions. To start formulating the optimization problem as a graph partitioning problem we will use the graph tools. All the terminology required has been explained in Chapter 2.

3.1 Graphical framing of our optimization problem

A graph shown in Figure 3.1 consists of 4 nodes and 5 edges.

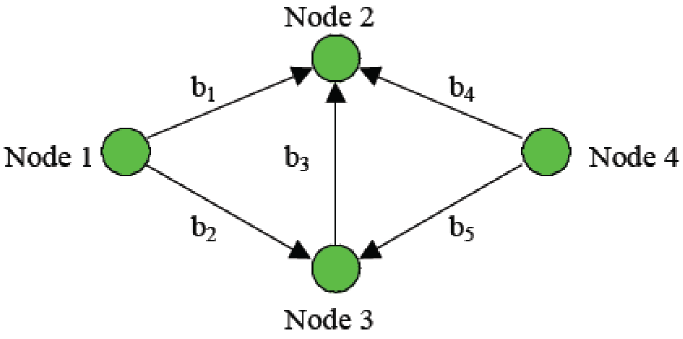


Figure 3.1 Graphical model for optimization problem

The corresponding node arc incidence matrix of the graph shown above is given as A in equation 3.1.

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \tag{3.1}$$

The first problem is to find the number of branches that will partition the above graph into two groups. Our original problem, as explained before, is to minimize these branches. In order to find these branches we will introduce an indicator vector x. The indicator vector x can have entries 1 or -1. The length of vector x is equal to the number of nodes in graph and vector x takes value +1 for all the nodes that lie in one particular group, and -1 for all the nodes that lie in another group. For example, as shown in the graph above, if we want nodes 1 and 2 in one group and nodes 3 and 4 in another group, then the indicator vector correspondingly will be given as:

$$x = [-1 \quad -1 \quad 1 \quad 1]^T \tag{3.2}$$

If we multiply vector x by the incidence matrix we can identify the branches that partition the graph in the desired groups. Let this vector be given as y:

$$y = Ax = \begin{bmatrix} 0 \\ -2 \\ 2 \\ 2 \\ 0 \end{bmatrix} \tag{3.3}$$

Entries in y correspond to the respective edges of the graph. The value of an entry in vector y is 0 if the branch is internal to a group, and is ± 2 if the branch connects the two groups. The sign depends on the direction of the arrow. y is the measure of the identification of branches that separate the graph into 2 groups, hence

$$y^T y = x^T (A^T A) x = 4 \text{ times the number of separating branches} \tag{3.4}$$

So, we have the measure of number of branches that will separate the graph into two groups. We denote the Laplacian matrix as $L = A^T A.$ First part of optimization is to minimize the number of branches which can be posed as an optimization problem as shown in equation 3.5.

$$x^* = \arg\ \min_{x_i \in \{-1,1\}} x^T L x \tag{3.5}$$

Returning to our optimization problem, in our thesis work we are finding those $N - k$ contingencies that will cause maximum power imbalance. That is only those outages that will result in maximum severity. We have already posed first part of our optimization problem to find minimum number of branches that will separate the power system network into two groups. But for our original problem we want to find the minimum number of events whose outage will result in maximum power imbalance between two groups. To obtain an expression for the power flows across a specific cut set of lines, we might employ a power flow program to calculate the line flows given the power injections. By assuming a lossless network, as an approximation, we can avoid this added complexity. Let the vector p denote the bus power injections and let x be an indicator vector with entries equal to ±1 then,

$$p^T x = 2 \text{ (power flow between groups)} \tag{3.6}$$

So the above expression gives the power flow between two groups, or it gives the power flow through the branches that partition the graph in two groups. Now, we are in the position to pose our optimization problem to find minimum number of outages that will cause maximum power imbalance.

$$min_{x \in \{-1,1\}} \frac{x^T A^T A x}{4} - \frac{c P^T x}{2} + constant \tag{3.7}$$

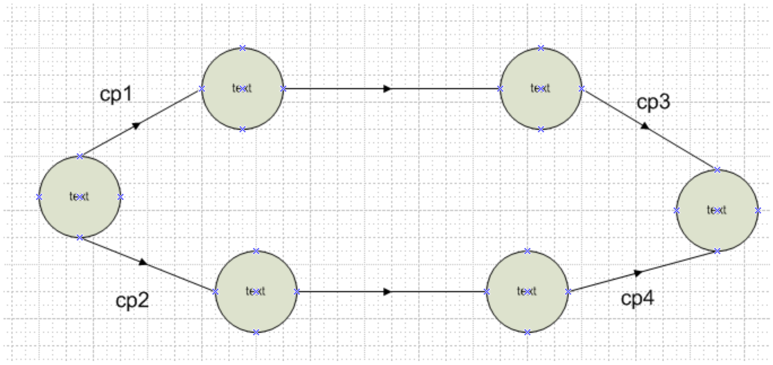
The first term is equal to the number of cuts which we want to minimize, the second term is proportional to the power flow disrupted by these cuts which we want to maximize, and the positive scalar c explicitly represents the trade-off between these objectives. Note the negative sign in front of power flow disrupted is because we want to maximize it. The trade-off value c which we described at the start of the chapter has been made explicit here.

3.2 Modeling

In the previous section we have posed an optimization problem to minimize the number of lines cut such that it maximizes the power imbalance or results in maximum severity. In this section we model our optimization problem using graph theory tools such that it is easy to simulate. For simulation we use Matlab [15]. We can visualize our optimization problem as a solution to max-flow/min-cut problem with an artificial source connected to all nodes corresponding to generators in a power system, and an artificial sink connected to all nodes corresponding to loads. The capacities of all edges connected from an artificial source to generators are equal to $c * power\ generated\ by\ generators$, and capacities of all edges connected from loads to artificial sink are equal to $c * power\ consumed\ by\ loads$. All the other edges in the graph have their capacities as 1. Mathematical proof of how the above mentioned problem of graph theory corresponds to our optimization problem will be given later in the chapter.

To Prove: Our optimization problem corresponds to a max-flow/min-cut problem.

Proof: Consider a candidate graph with above mentioned properties.



Where power generated by generators is given as $p1$ and $p2$, power consumed by loads as $p3$ and $p4$, and c is the trade-off value in our optimization problem. In general, Augemented Laplacian Matrix for graph with n nodes, bus incidence matrix A, a system with a power flow vector P, power generation vector Pg and power consumption vector Pl is given as:

$$\begin{bmatrix} \mathbf{0} & \mathbf{0}^T & \mathbf{0} \\ \mathbf{0} & \boldsymbol{A^T A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}^T & \mathbf{0} \end{bmatrix} + \boldsymbol{c} \begin{bmatrix} \mathbf{1}^T \boldsymbol{Pg} & -\boldsymbol{Pg}^T & \mathbf{0} \\ -\boldsymbol{Pg} & \boldsymbol{diag}(\boldsymbol{P}) & \boldsymbol{Pl} \\ \mathbf{0} & \boldsymbol{Pl}^T & -\mathbf{1}^T \boldsymbol{Pl} \end{bmatrix} \quad (\mathbf{3.8})$$

For a graph with above mentioned properties, indicator vector, y, is given as:

$$y = \begin{bmatrix} 1 \\ x \\ -1 \end{bmatrix} \quad (3.9)$$

Here x is the indicator vector of a graph without artificial source and sink.

From equation 3.4, 4 $*$ number of lines cut is given as y^T*Laplacian Matrix*y:

$$4*\text{Number of lines cut}=[1 \quad x^T \quad -1]L \begin{bmatrix} 1 \\ x \\ -1 \end{bmatrix}$$

$$[1 \quad x^T \quad -1]L\begin{bmatrix}1\\x\\-1\end{bmatrix} = x^TA^TAx - c2P^Tx + constant \tag{3.10}$$

Solution to max-flow/min-cut problem for given graph is to find the maximum amount flow from source to sink which is in equivalence with finding minimum number of lines cut that will disrupt the maximum flow. Max-flow/min-cut problem for our candidate graph can be written as:

$$min_{x\epsilon\{-1,1\}} No\ of\ Lines\ Cut \tag{3.11}$$

$$min_{x\in\{-1,1\}}\frac{x^TA^TAx}{4} - \frac{cP^Tx}{2} + constant \tag{3.12}$$

Comparing equation 3.12 and 3.7, it can be seen that our optimization problem is equivalent to max-flow/min-cut problem for the graph with desired properties.

We will complete our proof by giving detail analysis for a small 4 bus system. Let us take a candidate power system model of 4 buses with 2 generators and 2 loads. Let the power generated by generators be $p1$ and $p3$ and power absorbed by loads be $p2$ and $p4$. Power system model is represented in Figure 3.2:

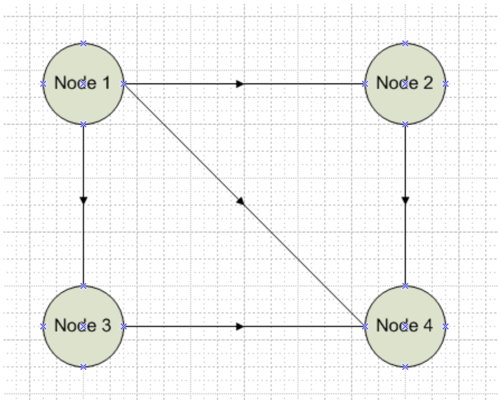


Figure 3.2 Graphical representation of 4 bus system

The node arc incidence matrix of a 4 bus system network is given by $A1$. Rows and columns of $A1$ correspond to edges and nodes in the same order as shown in above graph, respectively. A1 is given as:

$$A1 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \qquad (3.13)$$

Let us say the graph above is partitioned into two groups with Node 1 and Node 2 in one group and Node 3 and Node 4 in another group. Then indicator vector x will be given as:

$$\text{x}=\begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix} \qquad (3.14)$$

The branches that partition the graph into these two groups are edges 2, 3 and 4.

$$A1 * x = \begin{pmatrix} 0 \\ -2 \\ -2 \\ -2 \\ 0 \end{pmatrix} \qquad (3.15)$$

$$(x^T A1^T) * (A1 * x) = 12 \qquad (3.16)$$

Hence from above equation we can see that $(x^T A1^T) * (A1 * x) = 4$ *number of lines cut. Here, Lalplacian matrix is define as $A1^T * A1$. Our original optimization problem is to minimize the number of lines cut or outages such as it results in maximum severity. We can model this optimization problem by introducing an artificial source and an artificial sink connected to generators and loads respectively. Capacities of the edges directed from artificial source to generators and those directed from loads to artificial sink are formulated as c* power generated by generators and c*power absorbed by loads, respectively. Capacities of all internal edges are 1. The model for our optimization problem is shown in Figure 3.3:

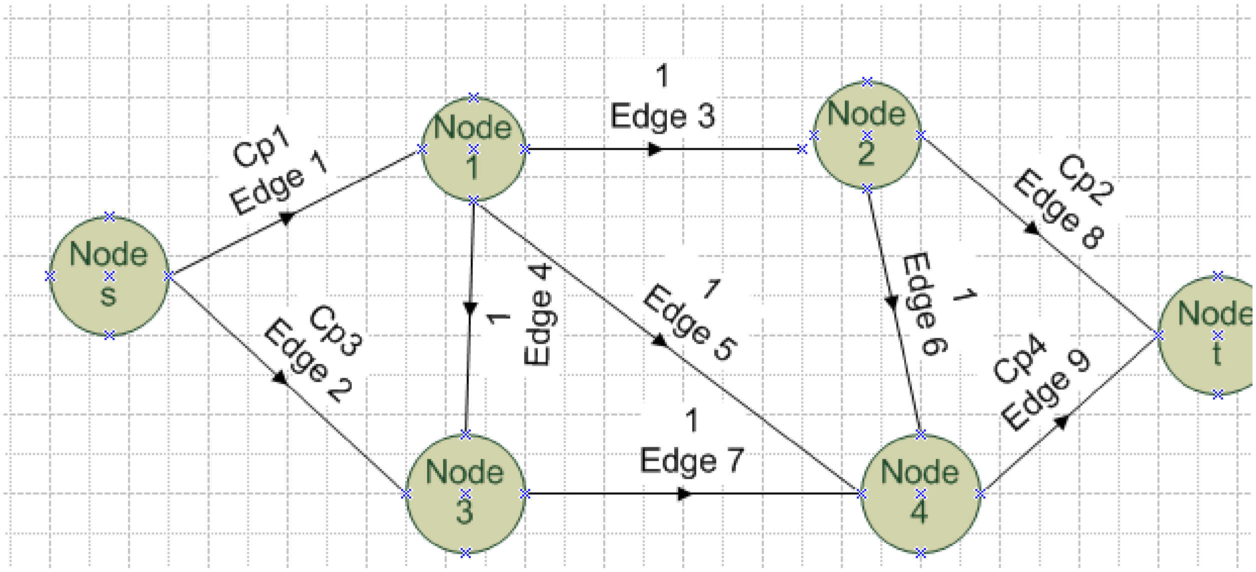


Figure 3.3 Graphical model for 4 bus system

Graph shown in Figure 3.3 has 6 nodes and 8 edges. Node 1 is an artificial source and Node 6 is an artificial sink. Node 2 and Node 3 correspond to generators and Node 4 and Node 5 corresponds to loads. $p1$ and $p3$ corresponds to the power generated by generators, and $p2$ and $p4$ corresponds to the power absorbed by loads. c is the trade-off variable. The columns and rows of node arc incidence matrix $A1$ (of 4 bus system network) are appended to formulate A (node arc incidence matrix of our modeled graph). Rows and columns of A correspond to edges and nodes in the same order as represented in above graph.

$$A = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.17}$$

The above graph can be partitioned such that source (Node s) and sink (Node t) lie in 2 different groups. Correspondingly indicator vector will be given as:

$$\text{y=}\begin{pmatrix} s \\ x1 \\ x2 \\ x3 \\ x3 \\ t \end{pmatrix}\text{, where s= } 1,\ x(i) \in \{-1,1\} \text{ and } t = -1 \tag{3.18}$$

Here, i corresponds to nodes 1, 2, 3 and 4. Numbers of cuts are given as $\frac{y^T * L * y}{4}$.

The minimum number of branches that will partition the graph in 2 groups such that it causes maximum power imbalance can be evaluated by max-flow/min-cut algorithm. We will now prove how the max-flow/min-cut algorithm corresponds to our original optimization problem. The Laplacian matrix, L, is given as $A^T * w * A$. Here w is a $n \times n$ diagonal matrix, n are the number of edges, and each diagonal entry corresponds to the weights or the capacities of respective edge. For above modeled graph w is given as:

$$w = \begin{pmatrix} cp1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & cp3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & cp2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & cp4 \end{pmatrix} \tag{3.19}$$

We can calculate $L = A^T wA$,

$$L = \begin{pmatrix} cp1 + cp3 & -cp1 & 0 & -cp3 & 0 & 0 \\ -cp1 & cp1 & 0 & 0 & 0 & 0 \\ 0 & 0 & cp2 & 0 & 0 & -cp2 \\ -cp3 & 0 & 0 & cp3 & 0 & 0 \\ 0 & 0 & 0 & 0 & cp4 & -cp4 \\ 0 & 0 & -cp2 & 0 & -cp4 & cp2 + cp4 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & -1 & -1 & -1 & 0 \\ 0 & -1 & 2 & 0 & -1 & 0 \\ 0 & -1 & 0 & 2 & -1 & 0 \\ 0 & -1 & -1 & -1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.20}$$

Now we can calculate the number of branches to partition the graph in 2 groups.

$$\text{Number of cuts=}\frac{y^T*L*y}{4} \tag{3.21}$$

Here, $y^T * L * y$ can be written as follows

$$(s \quad x1 \quad x2 \quad x3 \quad x4 \quad t) * \begin{pmatrix} c(p1 + p3) & -cp1 & 0 & -cp3 & 0 & 0 \\ -cp1 & cp1 + 3 & -1 & -1 & -1 & 0 \\ 0 & -1 & 2 + cp2 & 0 & -1 & -cp2 \\ -cp3 & -1 & 0 & cp3 + 2 & -1 & 0 \\ 0 & -1 & -1 & -1 & cp4 + 3 & -cp4 \\ 0 & 0 & -cp2 & 0 & -cp4 & cp4 + cp2 \end{pmatrix} \begin{pmatrix} s \\ x1 \\ x2 \\ x3 \\ x4 \\ t \end{pmatrix}$$

(3.22)

$$Let, L_{11} = c(p1 + p3), \qquad L_{12} = (-cp1 \quad 0 \quad -cp3 \quad 0), \qquad L_{13} = 0,$$

$$L_{21} = \begin{pmatrix} -cp1 \\ 0 \\ -cp3 \\ 0 \end{pmatrix}, \qquad L_{22} = \begin{pmatrix} cp1+3 & -1 & -1 & -1 \\ -1 & 2+cp2 & 0 & -1 \\ -1 & 0 & cp3+2 & -1 \\ -1 & -1 & -1 & cp4+3 \end{pmatrix}, \quad L_{23} = \begin{pmatrix} 0 \\ -cp2 \\ 0 \\ -cp4 \end{pmatrix},$$

$$L_{31} = 0, \qquad L_{32} = (0 \quad -cp2 \quad 0 \quad -cp4), \qquad L_{33} = cp4 + cp2$$

(3.23)

Using the set of equations 3.23, equation 3.22 can be written as:

$$(s \quad x1 \quad x2 \quad x3 \quad x4 \quad t) * \begin{pmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{pmatrix} * \begin{pmatrix} s \\ x1 \\ x2 \\ x3 \\ x4 \\ t \end{pmatrix} \tag{3.24}$$

Equation 3.21 implies addition of equations from 3.25 to 3.33. Equation 3.21 is equal to 4*number of branches that will partition the graph in 2 groups.

$$s * L_{11} * s = cp1 + cp3 \quad (Since\ s = 1\ as\ we\ have\ discussed\ above) \tag{3.25}$$

$$s * L_{12} * \begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \end{pmatrix} = -cp1 * x1 - cp3 * x3 \tag{3.26}$$

$$s * L_{13} * t = 0 \tag{3.27}$$

$$(x1 \quad x2 \quad x3 \quad x4) * L_{21} * s = -cp1 * x1 - cp3 * x3 \tag{3.28}$$

$$(x1 \quad x2 \quad x3 \quad x4) * L_{22} * \begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \end{pmatrix} = (x1 \quad x2 \quad x3 \quad x4) * \begin{pmatrix} cp1 & 0 & 0 & 0 \\ 0 & cp2 & 0 & 0 \\ 0 & 0 & cp3 & 0 \\ 0 & 0 & 0 & cp4 \end{pmatrix} *$$

$$\begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \end{pmatrix} + (x1 \quad x2 \quad x3 \quad x4) * \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{pmatrix} * \begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \end{pmatrix} = cp1 * x1^2 + cp2 * x2^2 +$$

$$cp3 * x3^2 + cp4 * x4^2 + x^T * (A1^T A1) * x \ = cp1 + cp2 + cp3 + cp4 + x^T * (A1^T A1) *$$

$$x \quad (Since\ x1, x2, x3\ and\ x4\ can\ take\ values\ only\ 1\ or - 1) \tag{3.29}$$

Here x corresponds to column vector (x1, x2, x3, x4) and $A1$ corresponds to node arc incidence matrix of original 4 bus model given in equation 3.13.

$$(x1 \quad x2 \quad x3 \quad x4) * L_{23} * t = cp2 * x2 + cp4 * x4 \quad (Since\ t = -1) \tag{3.30}$$

$$t * L_{33} * s = 0 \tag{3.31}$$

$$t * L_{32} * \begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \end{pmatrix} = cp2 * x2 + cp4 * x4 \quad (Since\ t = -1) \tag{3.32}$$

$$t * L_{33} * t = cp4 + cp2 \quad (Since\ t = -1) \tag{3.33}$$

Now with the help of above equations we can calculate the number of branches that will partition the graph in 2 groups. Sum of all equations from 3.25 to 3.33 will be equal to 4*number of cuts.

$$y^T * L * y = cp1 + cp3 \ \ + -cp1 * x1 - cp3 * x3 - cp1 * x1 - cp3 * x3 + cp1 + cp2 + cp3 + cp4 + x^T * (A1^T A1) * x + cp2 * x2 + cp4 * x4 + cp2 * x2 + cp4 * x4 + cp4 + cp2$$

(3.34)

$$y^T * L * y = x^T * (A1^T A1) * x + 2 * c(-p1 \quad p2 \quad -p3 \quad p4) * \begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \end{pmatrix} + 2 * (cp1 + cp2 + cp3 + cp4)$$

(3.35)

So Number of cuts that will partition the graph in 2 groups is given as

$$\frac{y^T * L * y}{4} = \frac{x^T * (A1^T A1) * x}{4} - \frac{c}{2} * (p1 \quad -p2 \quad p3 \quad -p4) * \begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \end{pmatrix} + constant$$

(3.36)

So. we have found the number of cuts, and we can return to our original optimization problem to minimize the number of branches such that outage of these branches will result in maximum severity. Max-flow/min-cut problem for graph model, we present, implies minimizing equation (3.36). Hence, we deduce that our original optimization problem is equivalent to max-flow/min-cut for our graph model.

Now, we can state the above proof. Given any power system network with n buses and vector p (entries of p corresponds to the power generated by generators (+ve values) and power absorbed by loads (–ve values)), to find all worst case $N - k$ contingencies, we can model our system to a graph with an artificial source node connected to all the generators through edges with capacities equal to c*power generated by generators and an artificial sink node to which all the loads are connected through edges with capacities equal to c*power absorbed by loads. All the internal edges have capacities equal to 1 and all the internal edges are bidirectional. Minimum s-t cut of our modeled graph is equivalent to finding $N - k$ contingencies for our original power system model. We will present now the algorithm for finding the trade-off value c that has been explicitly shown in equation 3.36.

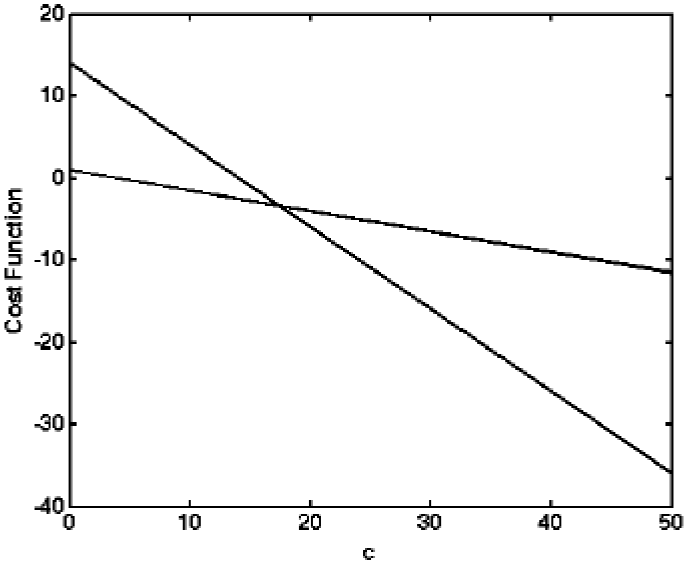
3.3 Algorithm for finding trade-off value c

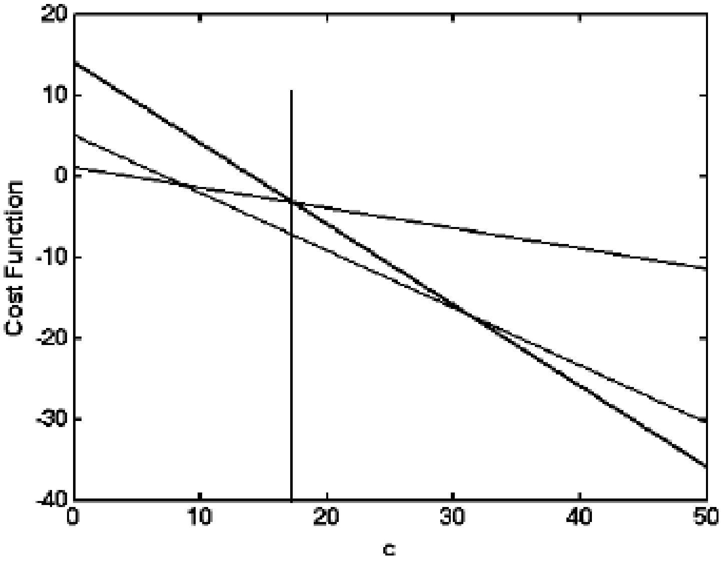
For a user defined trade-off value c, the approach we presented in Chapter 3 can be used to find minimum number of cuts that will result in maximum power disruption. This approach is to vary the value of c over a range, but this can potentially skip important values or inefficiently find repeated solutions

Rather than varying the value c we present an algorithm to find needed value of c's [16]. When c is very small, our optimization problem simplifies to simple min cut algorithm, and when c is very large, our problem completely separates generator and loads, so as to yield maximum power imbalance. As the value of parameter c varies, we can calculate other points that lie on and bound the vulnerability frontier.

For a specific grouping or for a specific value of x, we can view our optimization cost function (3.36) as a linear function of c. We can start with two easy solutions: a very large value of c (say

$c = 100$) and very small value of c (say $c = 0$). Max-flow/min-cut problem will yield a particular grouping x ($x0$ for c =100 and $x1$ for $c = 0$) such that 3.36 can be viewed as a linear function of c. To start with we have two set of solutions as shown in figure below. To determine if there are any other solutions possible we determine the value of $c*$ at which the two given solutions (linear functions of c) intersect. We solve max-flow/min-cut problem for the value of $c*$ and obtain the indicator vector $x*$. If $x*$ is equal to $x0$ or $x1$, we terminate, else we plot new line equivalent to equation 3.36 as a function of c (for grouping $x*$). We repeat above process calculating the intersection of new line with neighboring lines and calculating all possible new solutions. This process is bounded by the number of lines in the system.





Different values of c will give different groupings (solution of max-flow/min-cut problem), and for different groupings we will have different indicator vectors. For each indicator vector we calculate the lines cut (given by equation 3.4) and corresponding power imbalance (given by equation 3.6). Hence, we can plot the vulnerability frontier. In chapter 4 we will present vulnerability frontier for 30 bus, 118 bus, 179 bus, 225 bus and 1553 bus system. A special note to mention here before presenting the results is that we are not taking into account line cuts that will disconnect radial generators. We model this in our equivalent graph model by making capacities of lines connecting all radial generators to be very large.

Chapter 4 Results

In this chapter we will present results for 30 bus system, 118 bus system, 179 bus system, 225 bus system and 1553 bus system. For each system, we will present its vulnerability frontier, tables corresponding to frequency of cut-sets as they appear by themselves, and tables corresponding to frequency of cut-sets as they may appear as part of some other cut-sets.

4.1 30 bus system

For each system we draw a plot between the number of lines cut and associated max power imbalance. We are excluding the analysis of radical connection to generators.

Total points in the plot: 7

Max number of Lines Cut: 15 (completely separating generation and load)

Each solution shows a separation of generation-rich regions from load-rich regions. The sub-regions of load and generators need not be connected. So that the main cut set determined by our method may consist of sub cut sets, each of which would separate the entire graph into two parts. We rank the two-part cut sets – potentially critical lines – by the frequency in which they appear in the solutions – either alone as a separator of load/generation regions, or in any solution that separates load/generator regions.

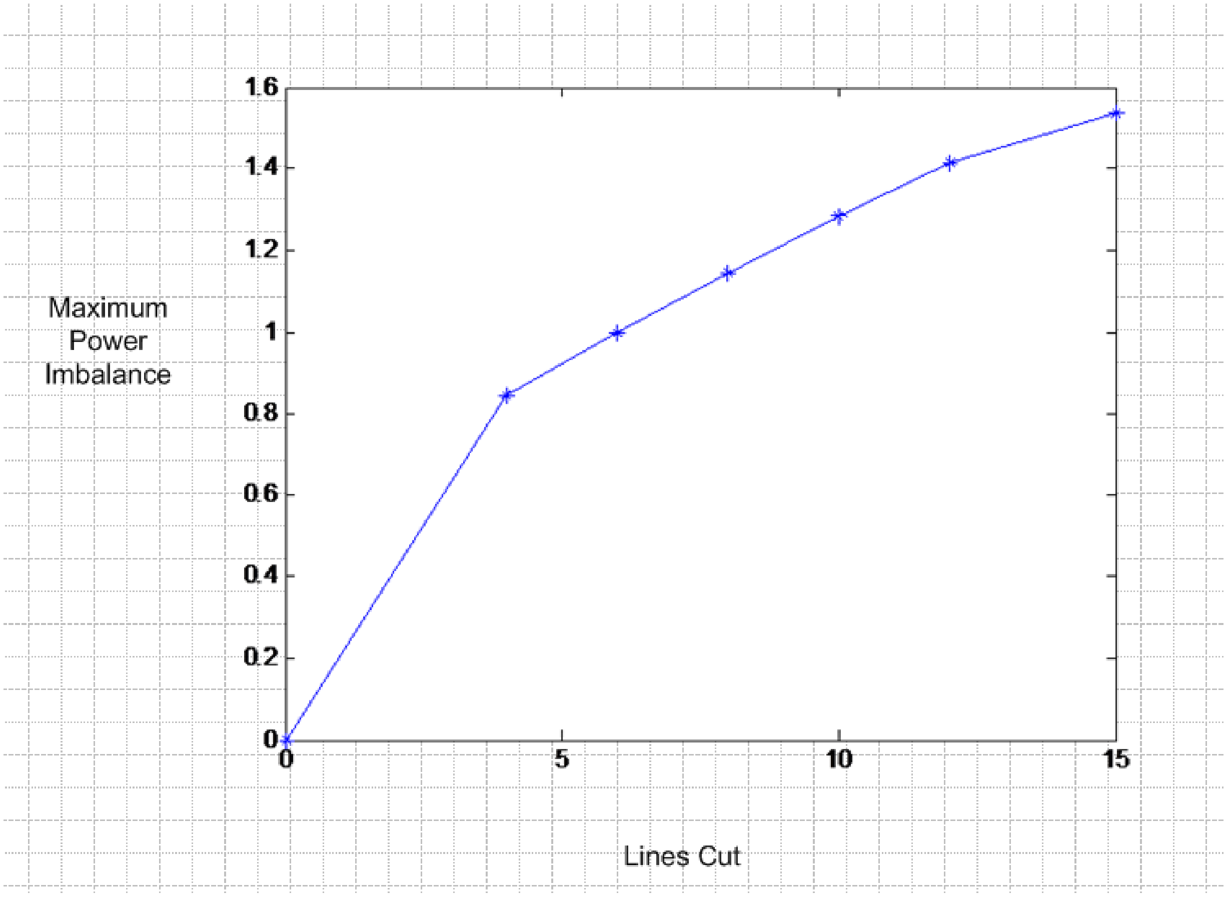


Figure 4.1 Vulnerability Frontier for 30 bus system

Table 1 and Table 2, shown below, contains the following information- a particular cut set, the number of times this cut set appears as a part of any cut set (alone or as a subset), and the number of times this cut set appears by itself. For the plot shown in Figure 4.1, we have total of 7 points that is we have 7 different cut sets forming the boundary of the frontier. Each cut set will divide the network into two groups. In Table 1, we arrange these cut sets in a descending order according to the number of times they appear as a part of any of the 7 solutions. In Table 2, we arrange these cut sets in a descending order according to the number of times they appear by themselves as a separator for a region.

Table number	Number of cut sets
1	6
2	6

Table 1

Lines (cut set)	Number of times cut set appears as part of any cut set	Number of times cut set appears by itself
28 29	6	1
1 2	4	4
15 17 18 19	3	3
28 29 30 36	3	3
15 19 22 28 29 36	2	2
37 38	2	2

Table 2

Lines (cut set)	Number of times cut set appears as part of any cut set	Number of times cut set appears by itself
1 2	4	4
15 17 18 19	3	3
28 29 30 36	3	3
15 19 22 28 29 36	2	2
37 38	2	2
28 29	6	1

Observations:

1) Cut set composed of Lines 28 , 29 appears maximum number of times when considered to appear as part of any solution (6 times out of total 7 points).

2) Cut set composed of Lines 1, 2 appears the most number of times by itself (4 times out of total 7 points).

3) There are in total 6 unique cut sets in total 7 points of the plot.

4.2 118 bus system

For each system we draw a plot between the number of lines cut and associated max power imbalance. We are excluding the analysis of generators with single radial connection.

Total points in plot: 15

Max number of Lines Cut: 50 (completely separating generation and load)

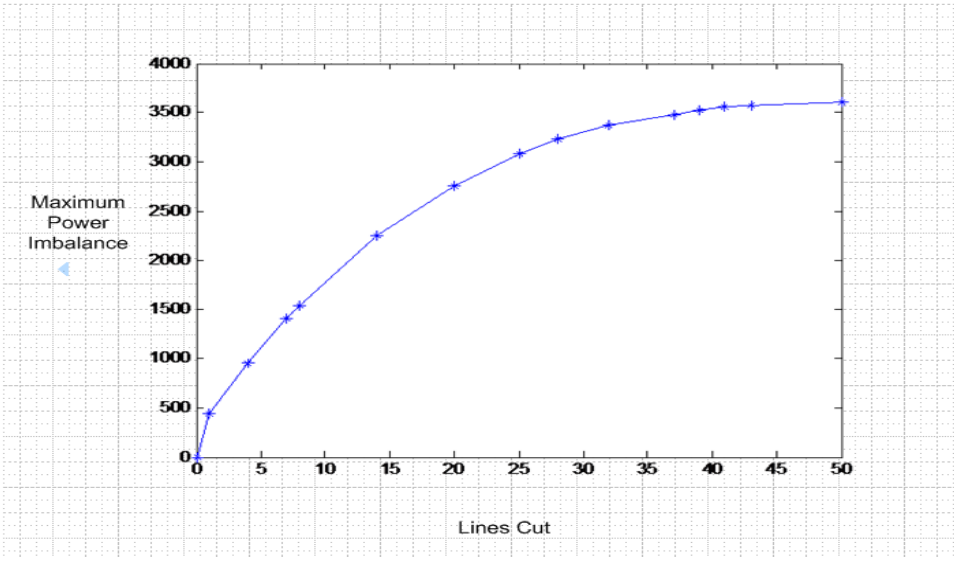


Figure 4.2 Vulnerability Frontier for 118 bus system.

Tables 3, 4 and 5, contains the following information- a particular cut set, the number of times this cut set appears as a part of any cut set (alone or as a subset), and the number of times this cut set appears by itself. For the plot shown in Figure 4.2, we have total of 15 points that is we have 15 different cut sets forming the boundary of the frontier. Each cut set will divide the network into two groups. In Table 3 we arrange these cut sets in a descending order according to the number of times they appear as a part of any of the 15 solutions. In Table 4 we arrange these cut sets in a descending order according to the number of times they appear by themselves as a separator for a region, In Table 5 we only consider first 10% of the total points (here first 6 points), and arrange the cut sets appearing in first 10% of points in descending order.

Table Number	Number of cut sets
3	27
4	27
5	12

Table 3

Lines (cut set)	Number of times cut set appears as part of any cut set	Number of times cut set appears by itself
31 33 38	8	8
136 137 139 142	7	2
136 137 138 139 141 142	5	5
154 155 160	4	2
70 71 76 89 90 92 94 100 103	3	1

138 139 189 197 221 256	3	3
163 164 167	3	3
65 66 67 68 69 96 105 108 116 119 123 124 125 148 151	2	2
70 71 75 76 89 90 92 94 100 103	2	2
89 90 92 94 96 99 100 103 105 106 108 116 119 124 125 148 151 152 153	2	1
1 158 166	1	1
5 8 195	1	1
24 25 218	1	1
29 40 44	1	1
36 37 39	1	1
50 51 53 60 256	1	1
65 67 68 69 96 105 108 116 119 124 125 148 151 154 155 160	1	1
70 72	1	1
89 90 92 94 96 98 99 100 103 105 106 108 116 119 123 124 125 148 151 152 153	1	1
89 90 92 94 96 98 99 100 103 105 106 108 116 119 123 124 125 148 151 154 155 160	1	1
96 97 98 99 100 103 104	1	1
96 110 113 199 230	1	1
105 106 107 108 116 119	1	1
120 121	1	1
133 135	1	1
138 140	1	1
164 165 166 167 174	1	1

Table 4

Lines (cut set)	**Number of times cut set appears as part of any cut set**	**Number of times cut set appears by itself**
31 33 38	8	8
136 137 138 139 141 142	5	5
138 139 189 197 221 256	3	3
163 164 167	3	3
65 66 67 68 69 96 105 108 116 119 123 124 125 148 151	2	2
70 71 75 76 89 90 92 94 100 103	2	2
136 137 139 142	7	2
154 155 160	4	2
1 158 166	1	1
5 8 195	1	1
24 25 218	1	1
29 40 44	1	1
36 37 39	1	1
50 51 53 60 256	1	1
65 67 68 69 96 105 108 116 119 124 125 148 151 154 155 160	1	1
70 71 76 89 90 92 94 100 103	3	1
70 72	1	1
89 90 92 94 96 98 99 100 103 105 106 108 116 119 123 124 125 148 151 152 153	1	1
89 90 92 94 96 98 99 100 103 105 106 108 116 119 123 124 125 148 151 154 155 160	1	1

89 90 92 94 96 99 100 103 105 106 108 116 119 124 125 148 151 152 153	2	1
96 97 98 99 100 103 104	1	1
96 110 113 199 230	1	1
105 106 107 108 116 119	1	1
120 121	1	1
133 135	1	1
138 140	1	1
164 165 166 167 174	1	1

Table 5

Lines (cut set)	**Number of times cut set appears as part of any cut set**	**Number of times cut set appears by itself**
138 139 189 197 221 256	3	3
1 158 166	1	1
5 8 195	1	1
24 25 218	1	1
29 40 44	1	1
36 37 39	1	1
50 51 53 60 256	1	1
70 72	1	1
96 110 113 199 230	1	1
120 121	1	1
133 135	1	1
138 140	1	1

4.3 179 bus system

For each system we draw a plot between the number of lines cut and associated max power imbalance. We are excluding the analysis of generators with single radial connection.

Total points in plot: 6

Max number of Lines Cut: 60 (completely separating generation and load)

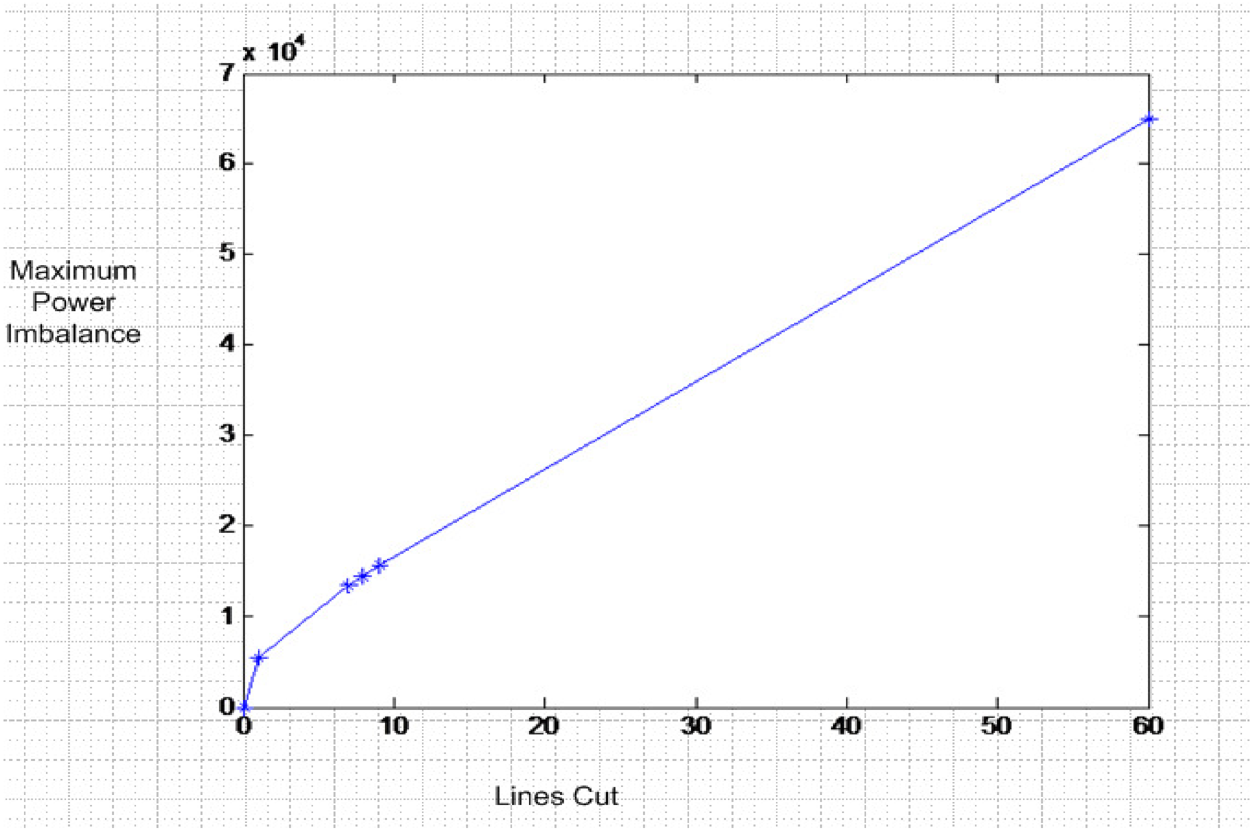


Figure 4.3 Vulnerability Frontier for 179 bus system.

Each solution shows a separation of generation-rich regions from load-rich regions. The sub-regions of load and generators need not be connected. So that the main cut set determined by our method may consist of sub cut sets, each of which would separate the entire graph into two parts. We rank the two-part cut sets – potentially critical lines – by the frequency in which they appear in the solutions – either alone as a separator of load/generation regions, or in any solution that separates load/generator regions. Table 6 shows the cut sets arranged in descending order as they appear together in different solutions (column 2) and as they appear separate by itself as a solution (column3).

Table 6

Lines (cut set)	**Number of times cut set appears as part of any cut set**	**Number of times cut set appears by itself**
138 139 189 197 221 256	3	3
1 158 166	1	1
5 8 195	1	1
24 25 218	1	1
29 40 44	1	1
36 37 39	1	1
50 51 53 60 256	1	1
70 72	1	1
96 110 113 199 230	1	1
120 121	1	1
133 135	1	1
138 140	1	1

4.4 225 bus system

For 225 bus system we will present only the vulnerability frontier. Because of the NDA agreement, we will not present the cut-sets or potentially important lines for 225 bus system. Rather, we will present the map of 225 bus system and the sequential variation in map as we move up the points on vulnerability frontier. For different points on vulnerability frontier we have different groupings of generator rich region and load rich region.

Total points in plot: 30

Max number of Lines Cut: 78 (completely separating generation and load)

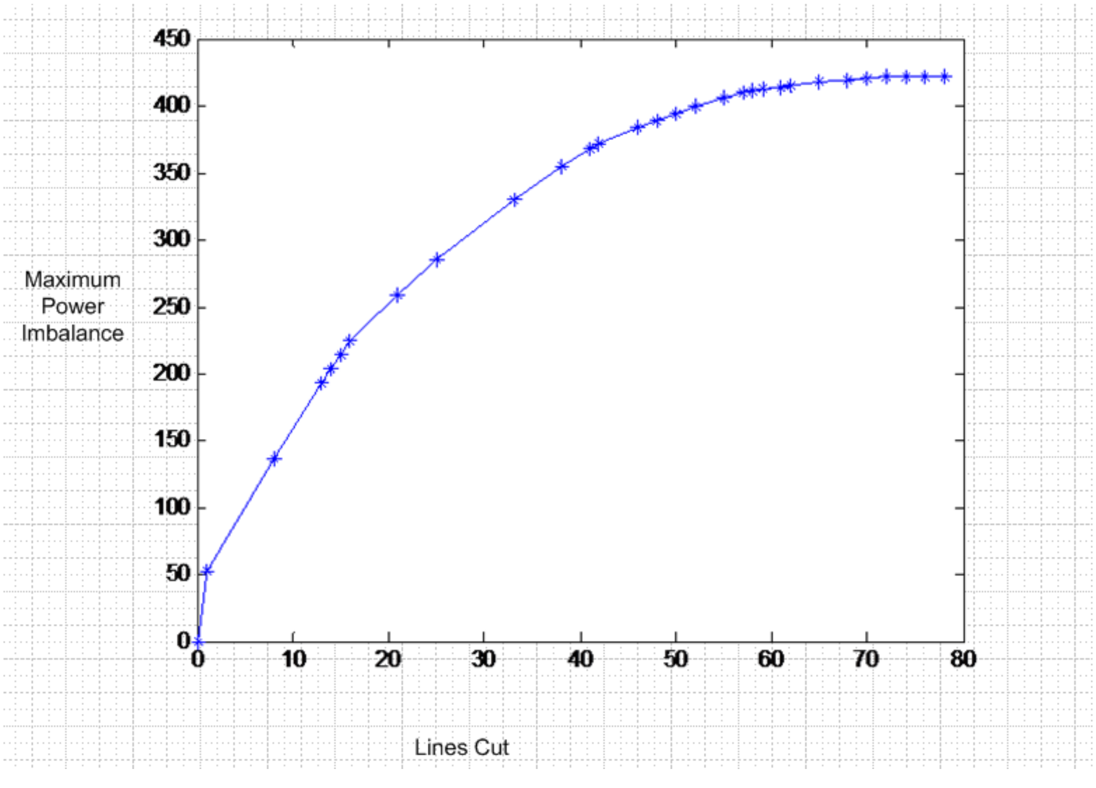


Figure 4.4 Vulnerability Frontier for 225 bus system.

Figures 4.5, 4.6, and 4.7 shows the sequence of maps of 225 bus system Blue lines lie in one group, green lines lie in a different group and red lines are those which partition the graph in blue and red groups. This work was done in Mathematica [17]. These figures correspond to first 10% points (3points) on vulnerability frontier. For sequential maps for each point on vulnerability frontier refer to Appendix.

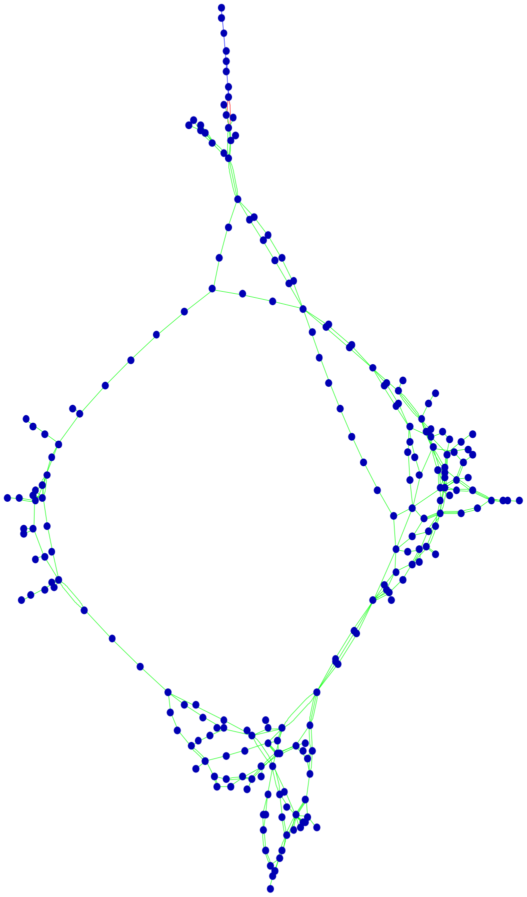


Figure 4.5 Partition of 225 bus map with a single line cut

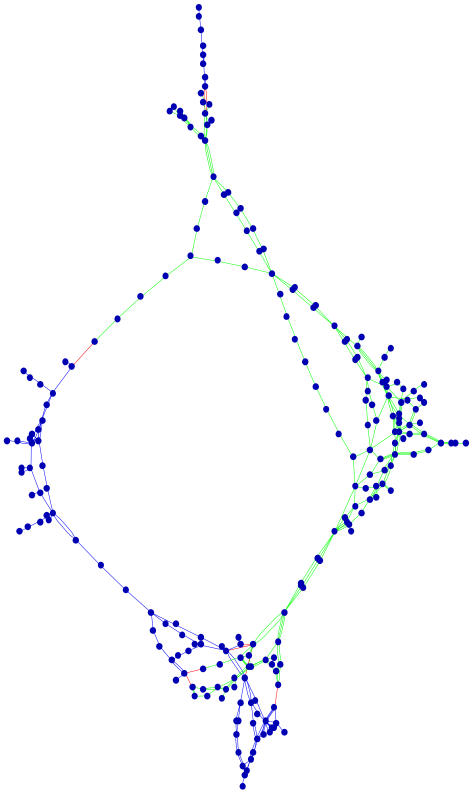


Figure 4.6 Partition of 225 bus map with 8 lines cut

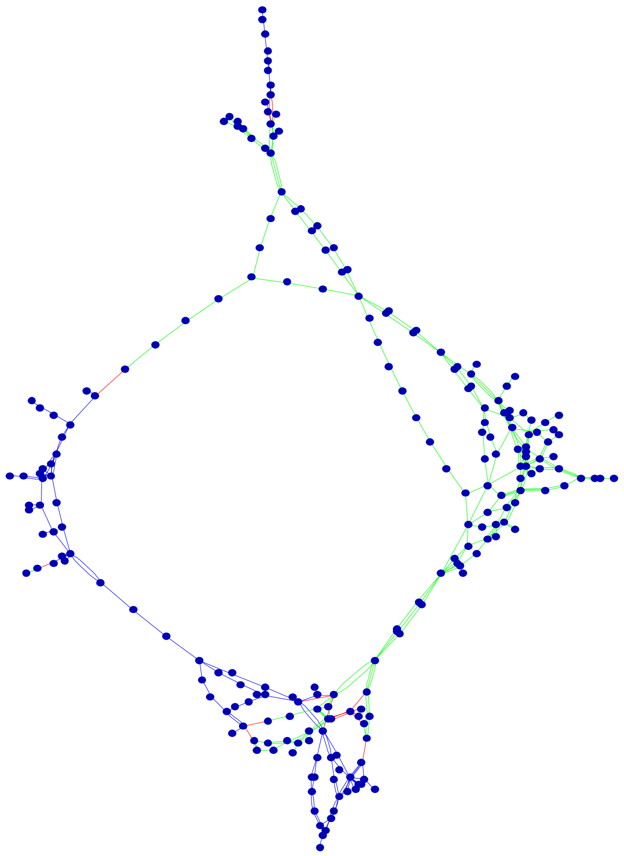


Figure 4.7 Partition of 225 bus map with 15 Lines cut

4.5 1553 bus system

In this section we will only present the vulnerability frontier for 1553 bus system. For this system we draw a plot, as shown in Figure 4.8, between the number of lines cut and associated max power imbalance.

Total points in plot: 189

Max number of Lines Cut : 536 (completely separating generation and load)

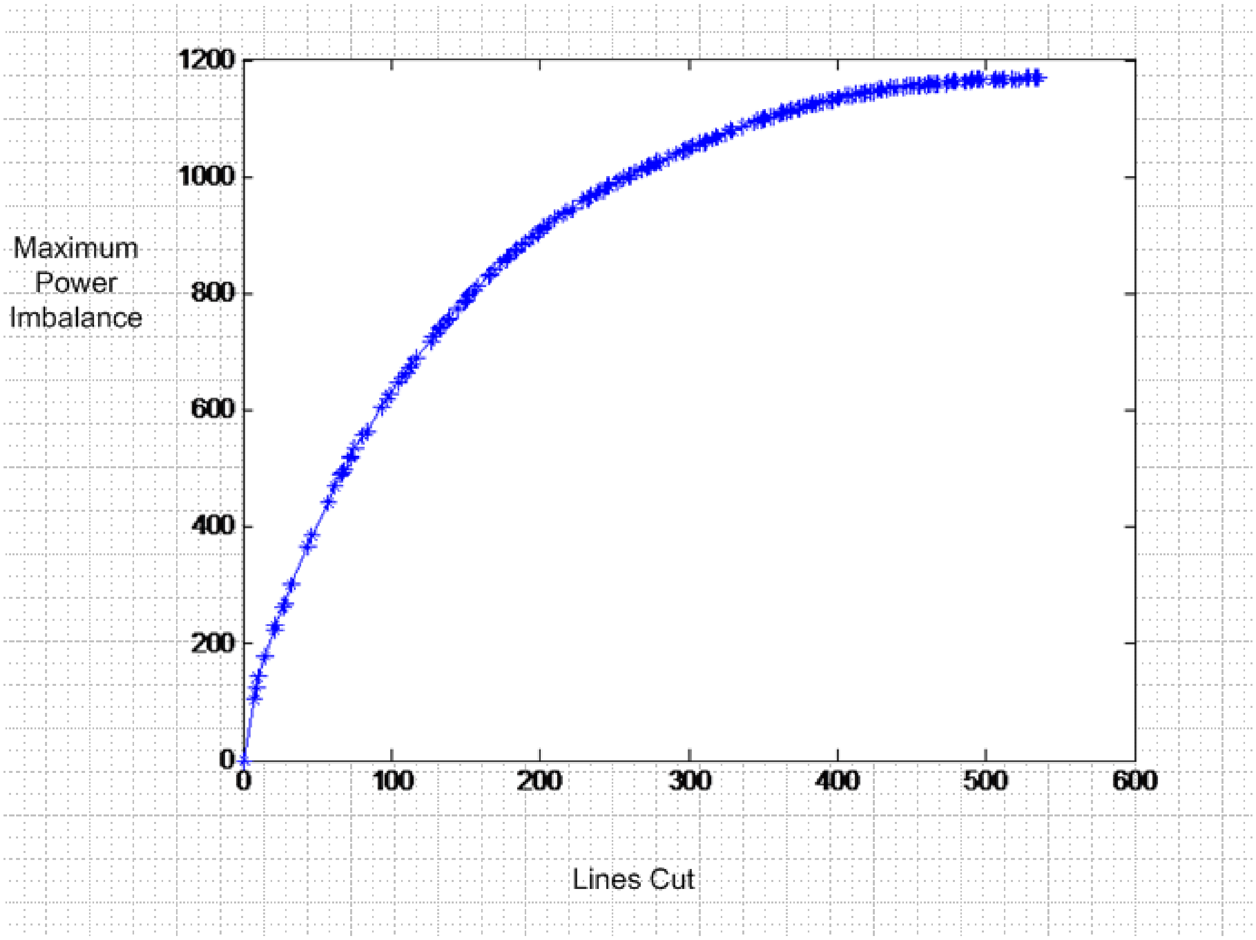


Figure 4.8 Vulnerability Frontier of 1553 bus system.

Chapter 5 Conclusions

Since it is computationally very difficult to find all possible $N-k$ contingencies, we presented a worst case approach to find severe contingencies. We only consider worst case scenarios that result in maximum power disruption.

We approach the problem of finding all possible contingencies as an optimization problem of minimizing the outages that will cause maximum power disruption. For the worst case scenario, we need to find the minimum number of events that will cause specified amount of severity or given the number of events (contingencies) we need to figure out the maximum power disruption that the contingencies results in. We introduce a trade-off variable c between events and power imbalance that they result in. We can solve this optimization problem using Multi-integer formulation, or various other techniques. But in our thesis work we show that this optimization problem is equivalent to max-flow/min-cut problem for our proposed graphical model. Any power system network with n number of buses and e number of edges can be reframed into a graphical model with an artificial source connected to all generators and all loads connected to artificial sink. All internal edges are equivalent to lines in original power system model. Max-flow/min-cut problem on proposed graphical model is equivalent to our original optimization problem. For user specified value of c, we can calculate the lines cut and the power imbalance from our proposed theory. Rather than using the user defined value for c, we propose an algorithm for finding the needed values of c to construct the vulnerability frontier.

The vulnerability frontier is the plot between lines cut on x-axis and corresponding power imbalance that those lines cut results in on y axis. So, in our thesis, we present analysis of all possible contingencies using vulnerability frontier. Different points on vulnerability frontier depicts potentially important lines whose outages may result in black out. Frequency of lines appearing independently as solution or as a part of other solution is tabulated revealing some important considerations in power security. These potentially important lines serve as critical corridors for future research. Results for 30 bus system, 118 bus system, 179 bus system, 225 bus system and 1553 bus system are presented in detail.

References

[1] U.S Secretary of Energy, Minister of Natural Resources Canada. (2004). Final report on the august 13th blackout

[2] Electricity Consumers Resource Council. (2004). The economic imapcts of august 13th blackout.

[3] Donde, V., Lopez, V., Lesieutre, B., Pinar, A., Yang, C., & Meza, J. (2006). Identification of severe multiple contingencies in electric PowerSystems. *IEEE Transactions on Power Systems, 23*(2).

[4] Chen, Q., & McCalley, J. (2005). Identifying high risk nk contingencies for online security assessment. *IEEE Transactions on Power Systems, 20*(2), 823-834.

[5] Zhu, J., & Xu, G. (1991). Approach to automatic contingency selection by reactive typeperformance index. *IEE Proceedings C Generation, Transmission and Distribution [See also IEE Proceedings-Generation, Transmission and Distribution], 138*(1), 65-68.

[6] Haidar, A. M. A., Mohamed, A., & Hussain, A. (2008). Vulnerability assessment of power system using radial basis function neural network and a new feature extraction method. *American Journal of Applied Sciences, 5*(6), 705-713.

[7] Salmeron, J., Wood, K., & Baldick, R. (2004). Analysis of electric grid security under terrorist threat. *IEEE Transactions on Power Systems, 19*(2), 905-912.

[8] Pinar, A., Reichert, A., & Lesieutre, B. (2007). Computing criticality of lines in power systems. *IEEE International Symposium on Circuits and Systems, 2007. ISCAS 2007,* 65-68.

[9] Pinar, A., Meza, J., Donde, V., & Lesieutre, B. Optimization strategies for the vulnerability analysis of the electric power grid. *OPTIMIZATION, 20*(4), 1786-1810.

[10] Wikipedia. *Graph theory.*

[11] Anderson, W. N., & Morley, T. D. (1985). Eigenvalues of the laplacian of a graph. *Linear and Multilinear Algebra, 18*(2), 141-145.

[12] *Max-flow/min-cut theorem.*http://en.wikipedia.org/wiki/Max-flow_min-cut_theorem

[13] Wikipedia. http://en.wikipedia.org/wiki/Cut_(graph_theory)

[14] Bertsekas, D. P. (1998). *Network optimization: Continuous and discrete models* Citeseer.

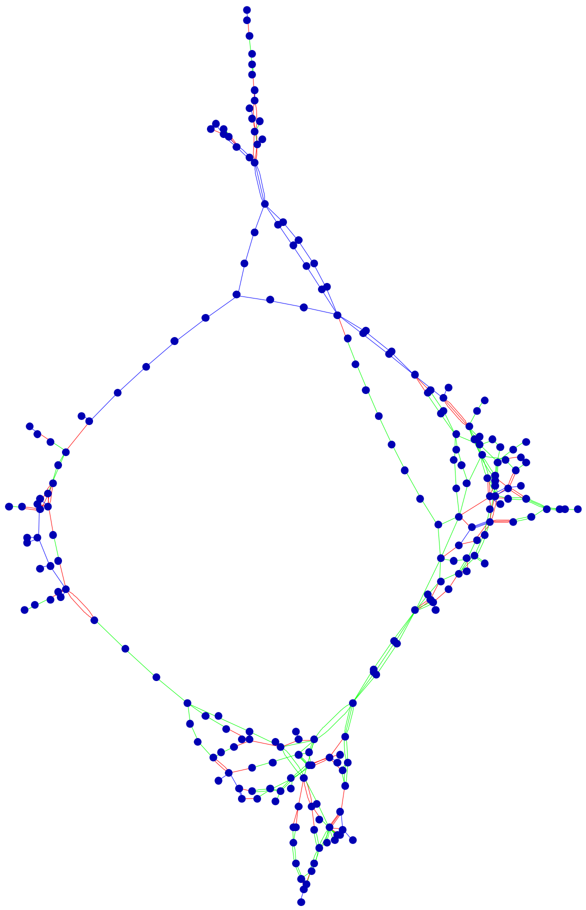
[15] Matlab, U. (1760). The MathWorks. *Inc., Natick, MA, 1992*

[16] Lesieutre, B. C., Pinar, A., & Roy, S. (2008). Power system extreme event detection: The vulnerability frontier. *Hicss,* 184.

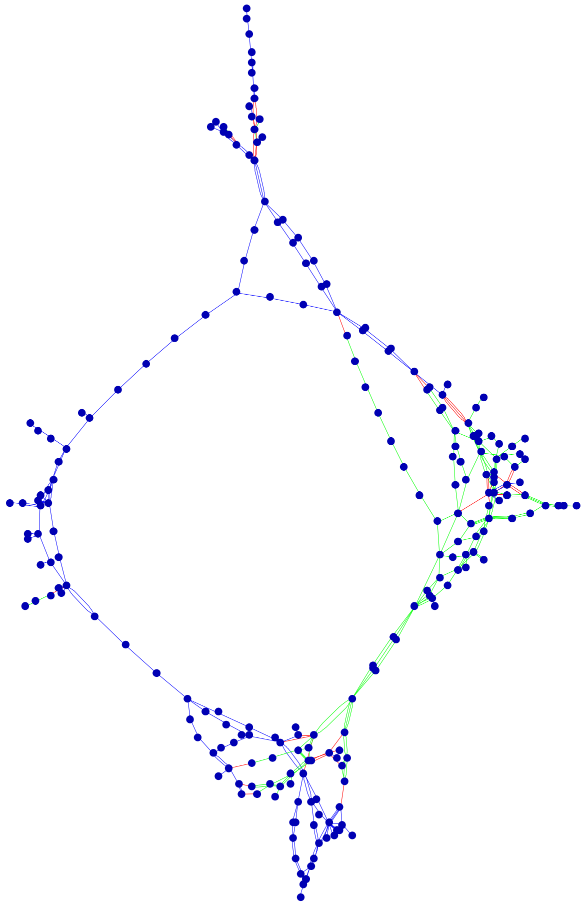
[17] Wolfram, S. (1999). *The mathematica book* Cambridge university press.

Appendix

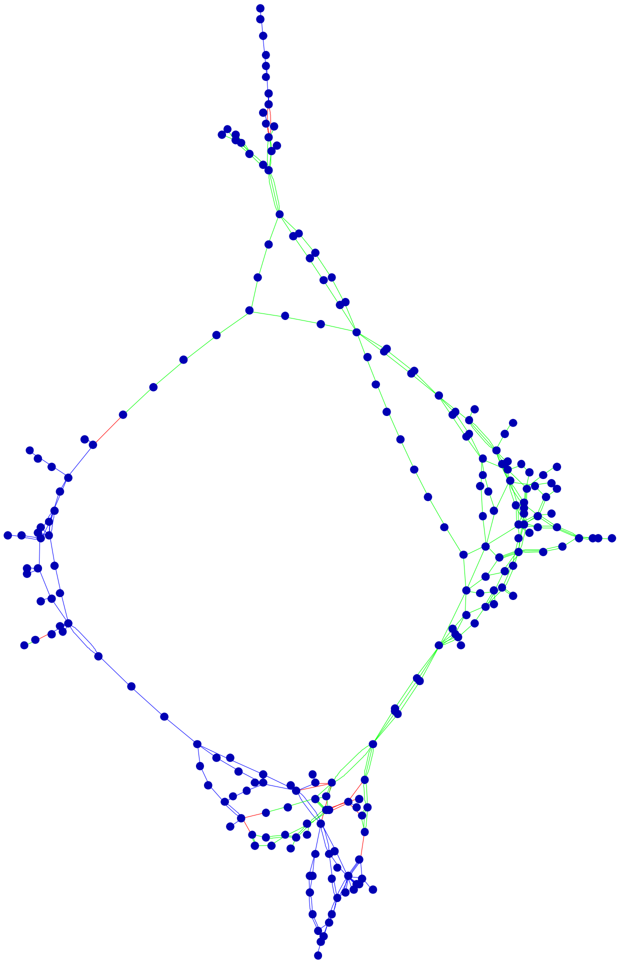
Lines being cut: 78



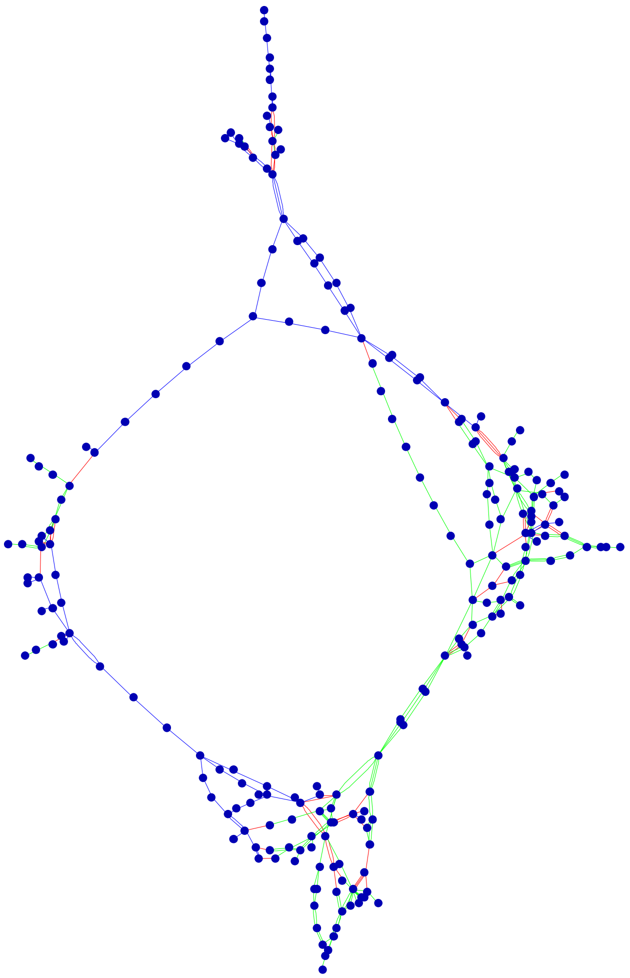
Lines being cut: 33



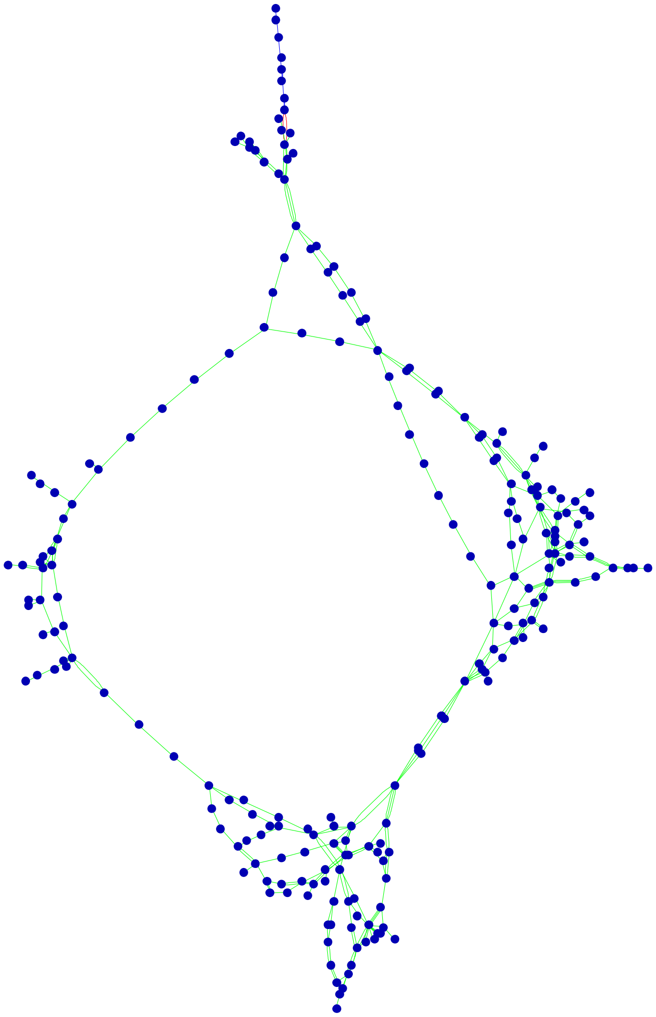
Lines being cut: 15



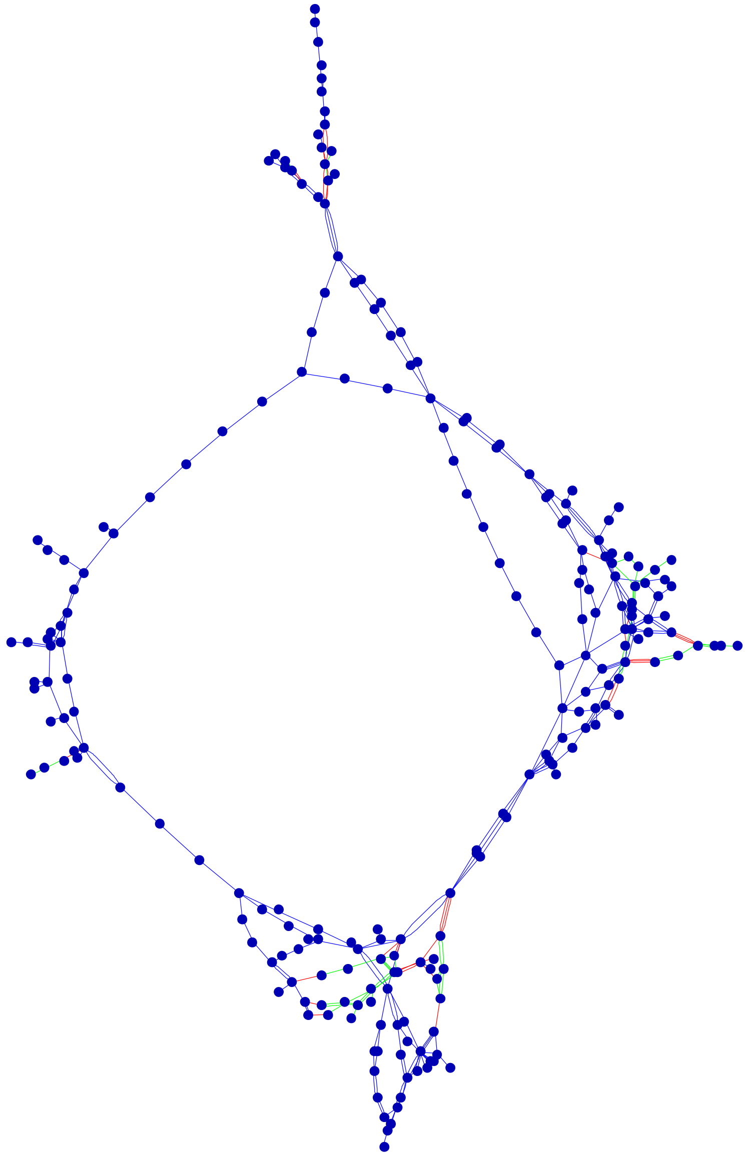
Lines being cut: 52



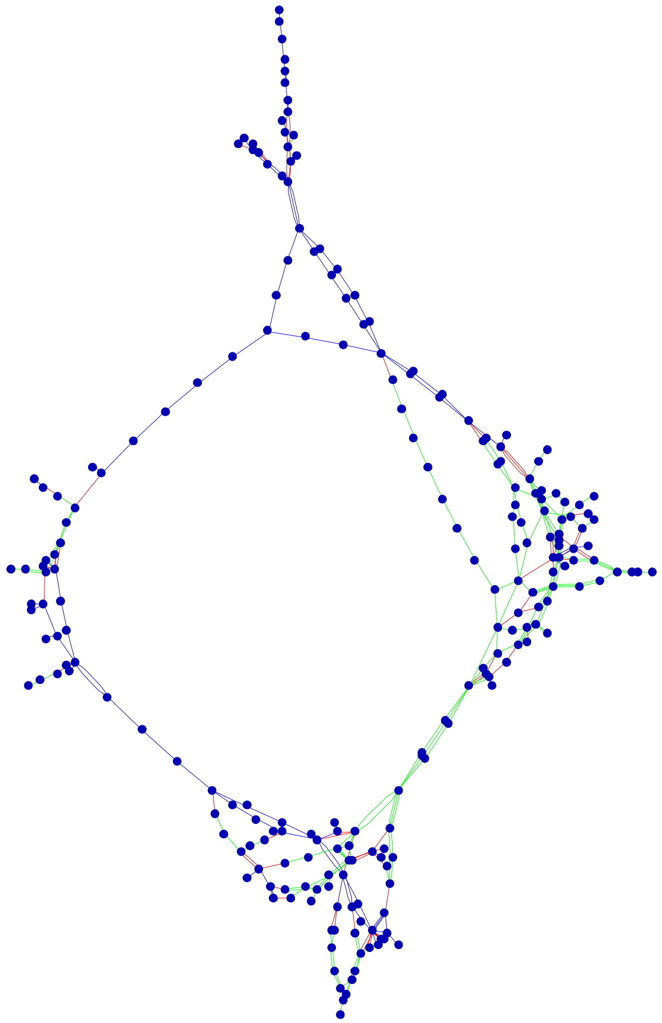
Lines being cut:1



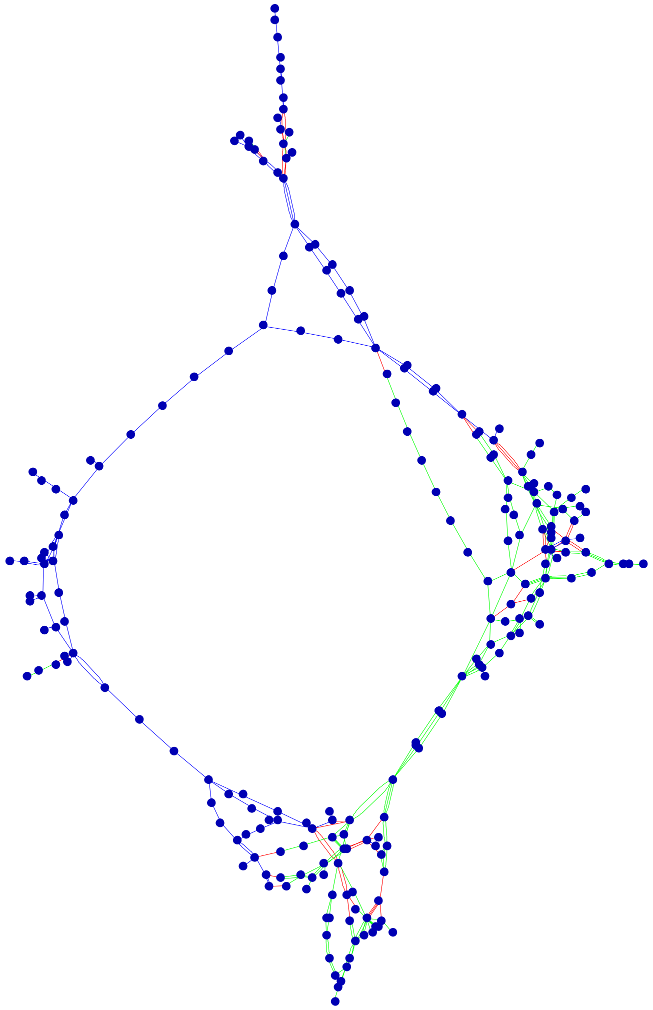
Lines being cut: 25



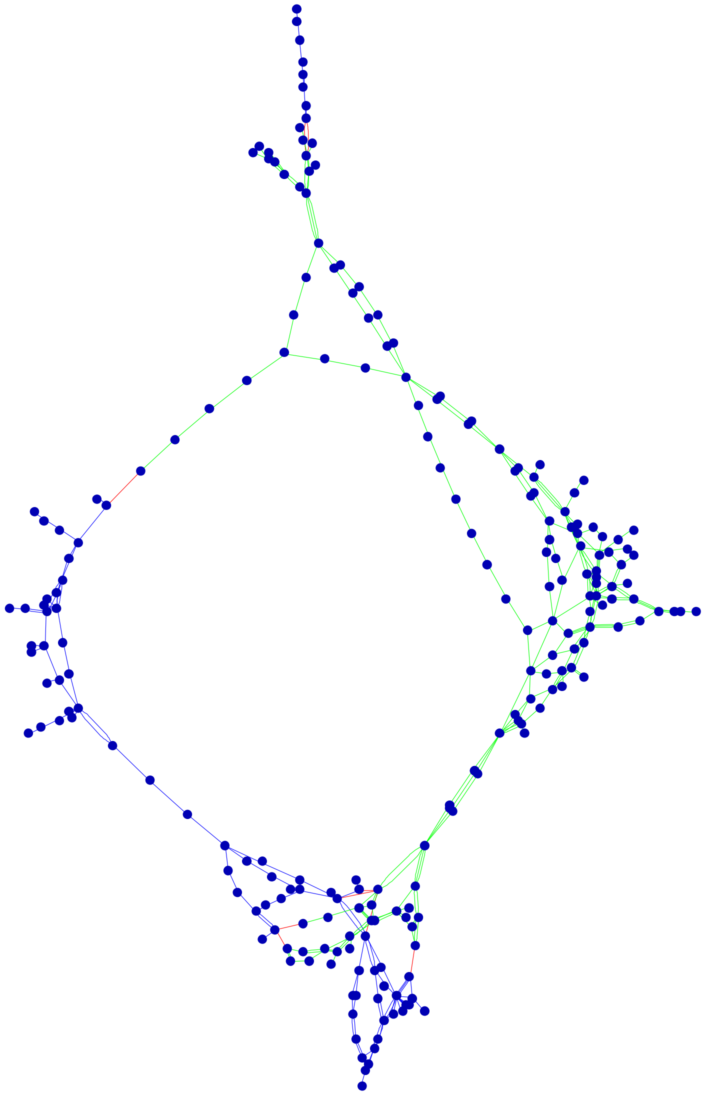
Lines being cut: 62



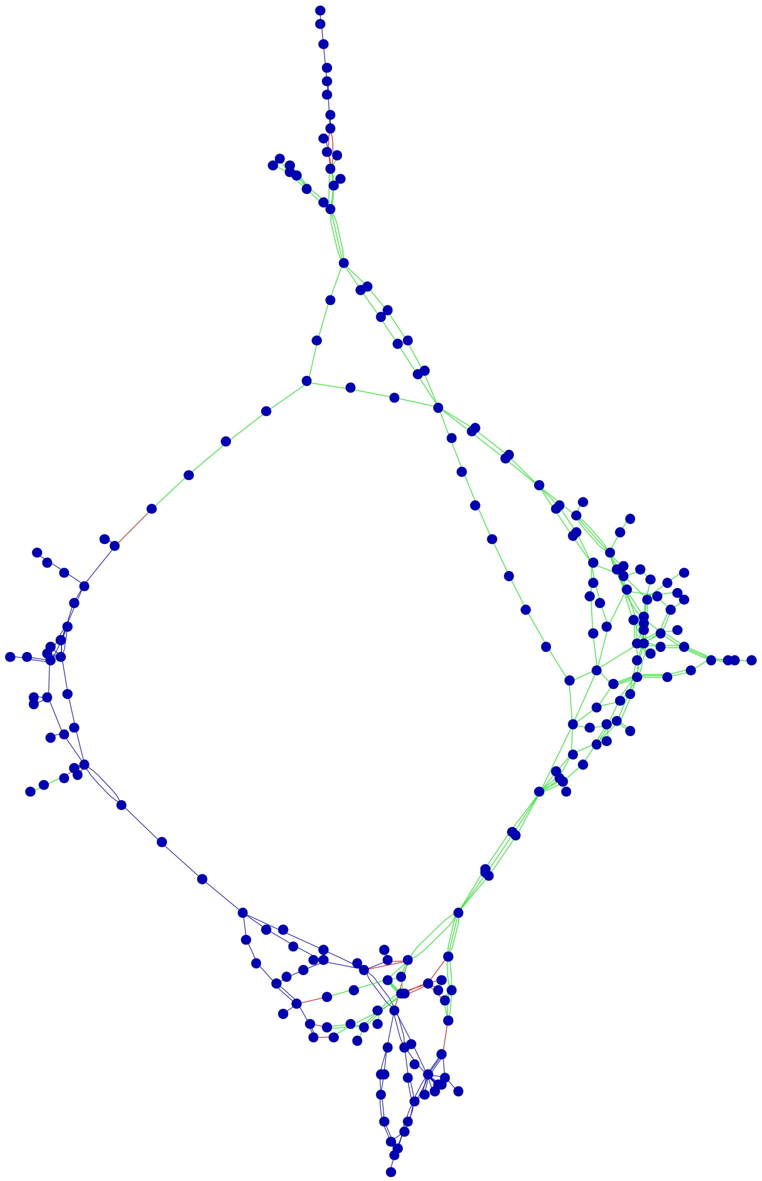
Lines being cut: 41



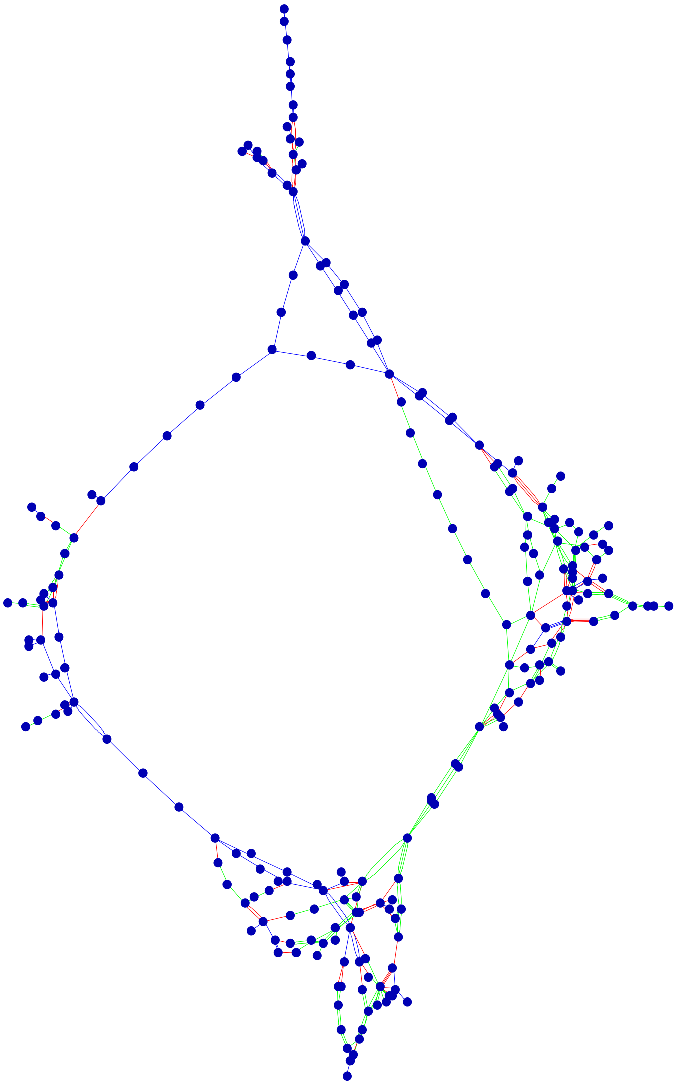
Lines being cut: 8



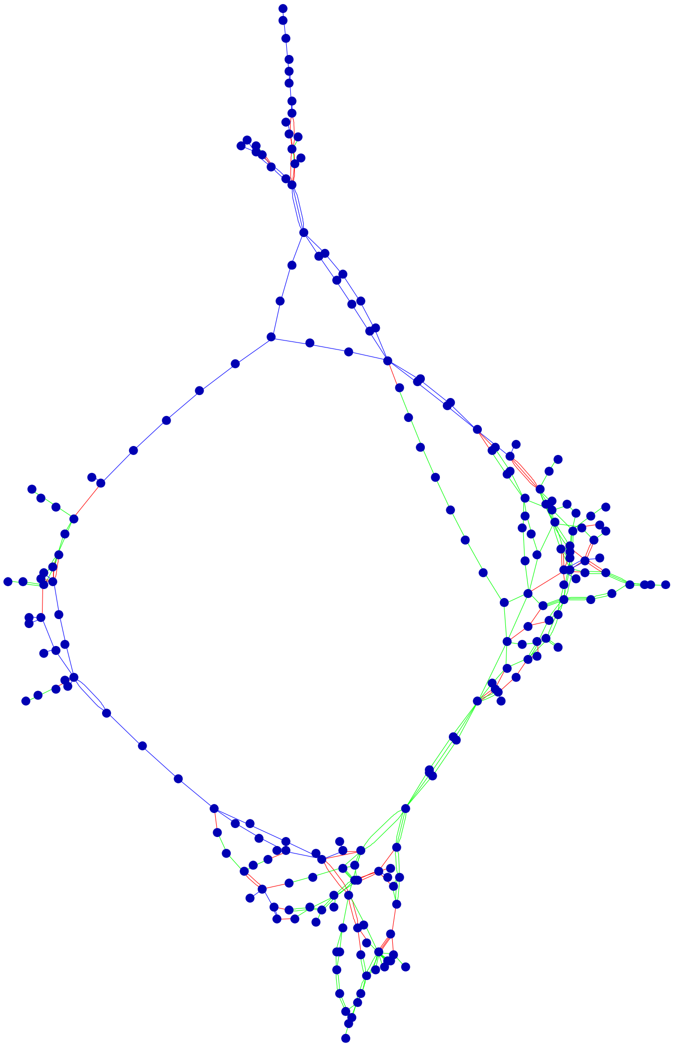
Lines being cut: 16



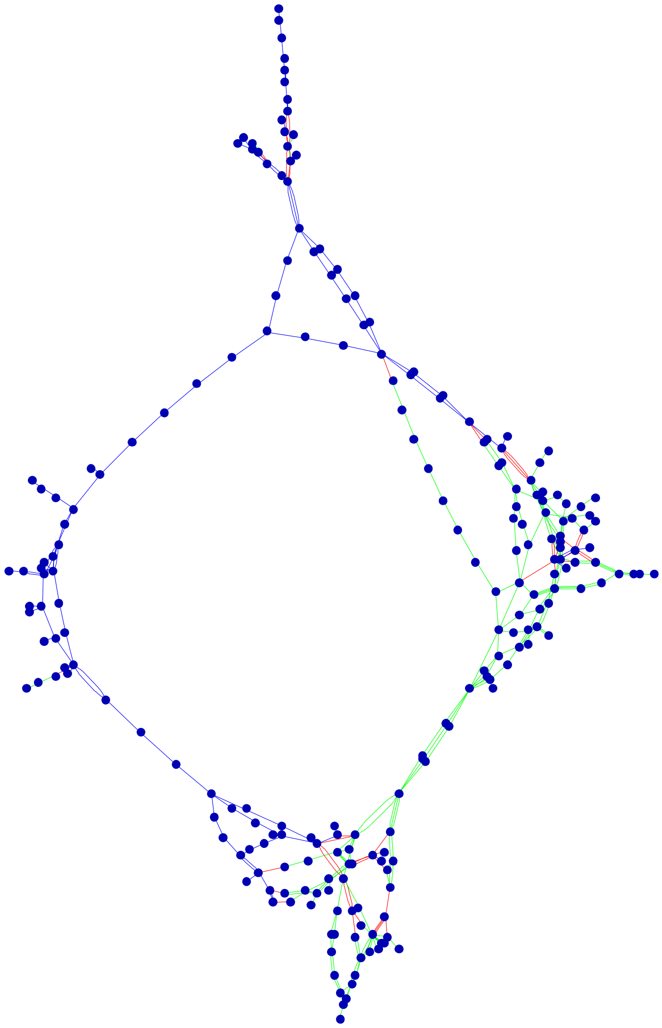
Lines being cut:70



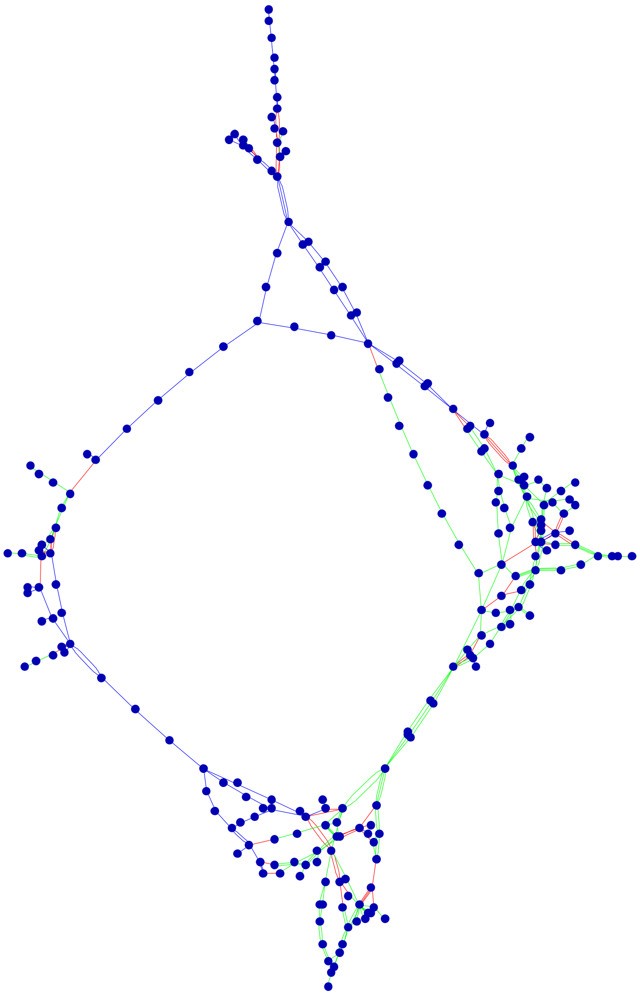
Lines being cut: 57



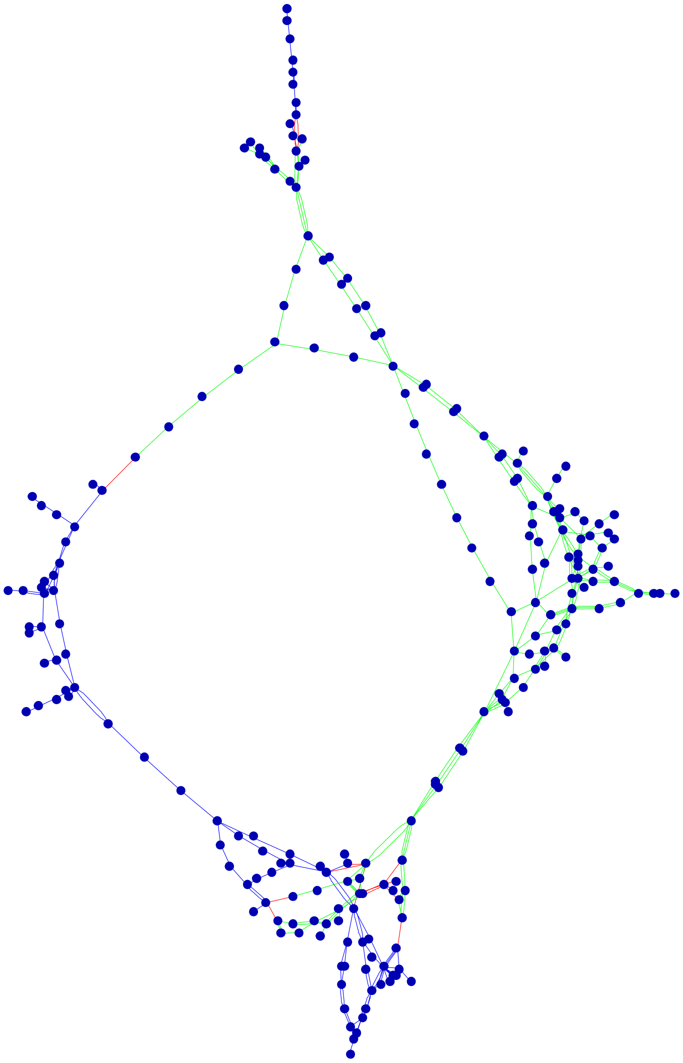
Lines being cut: 38



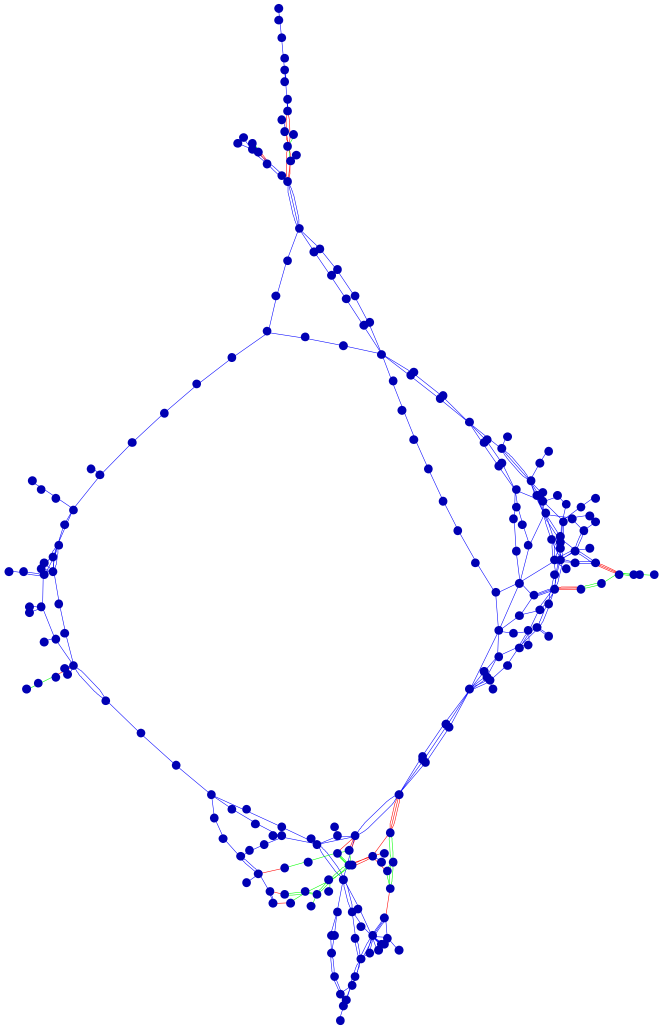
Lines being cut: 48



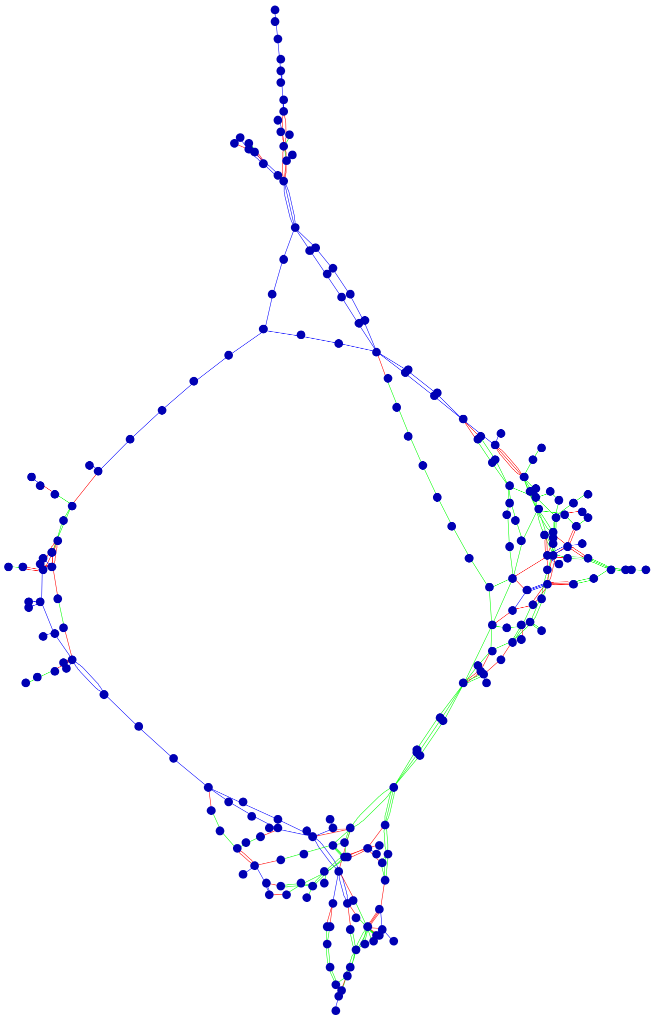
Lines being cut: 14



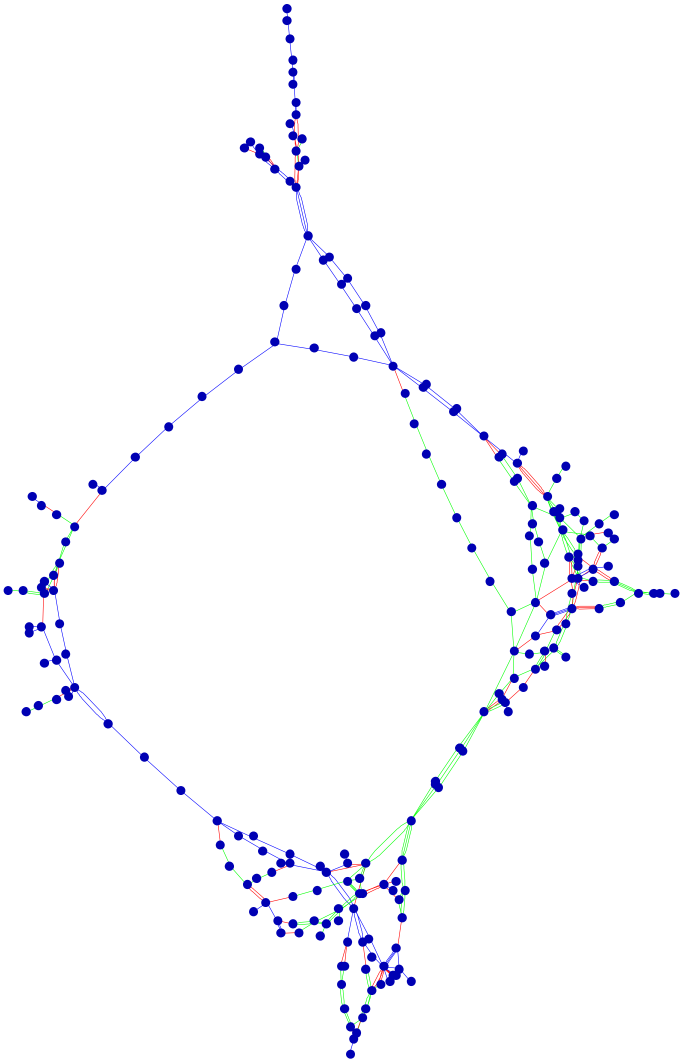
Lines being cut: 21



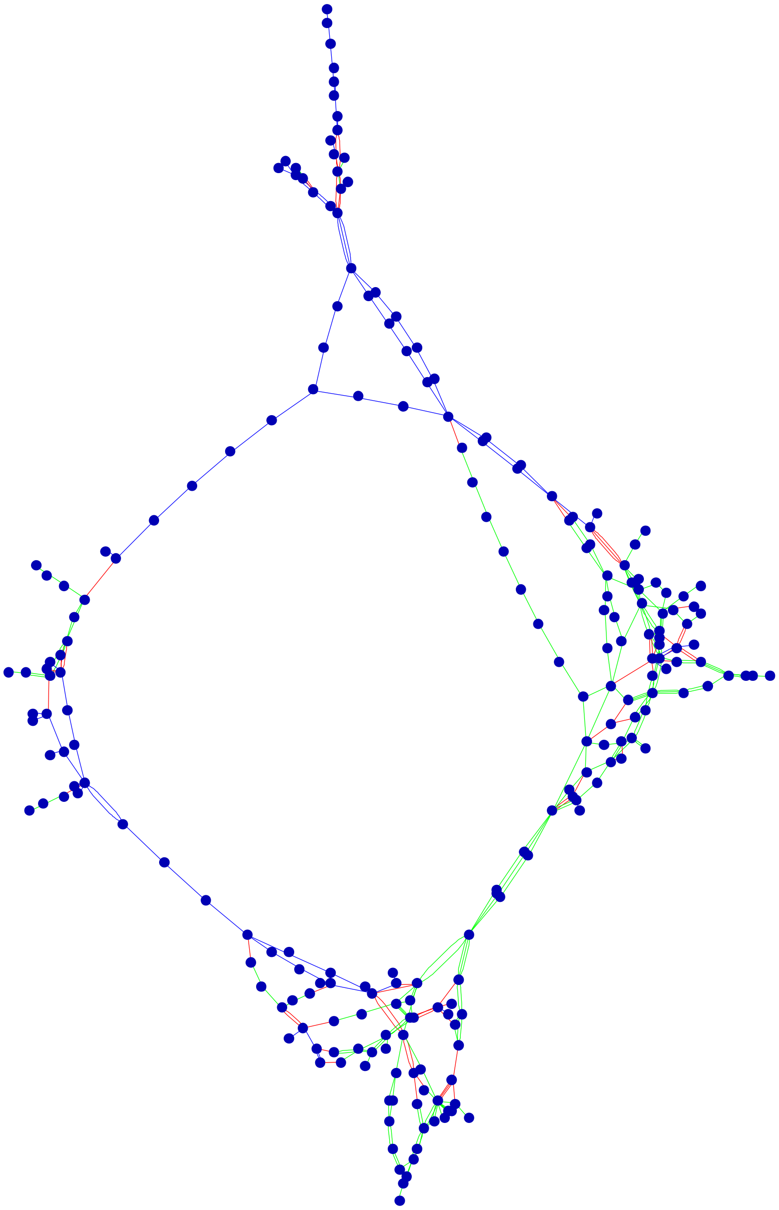
Lines being cut: 74



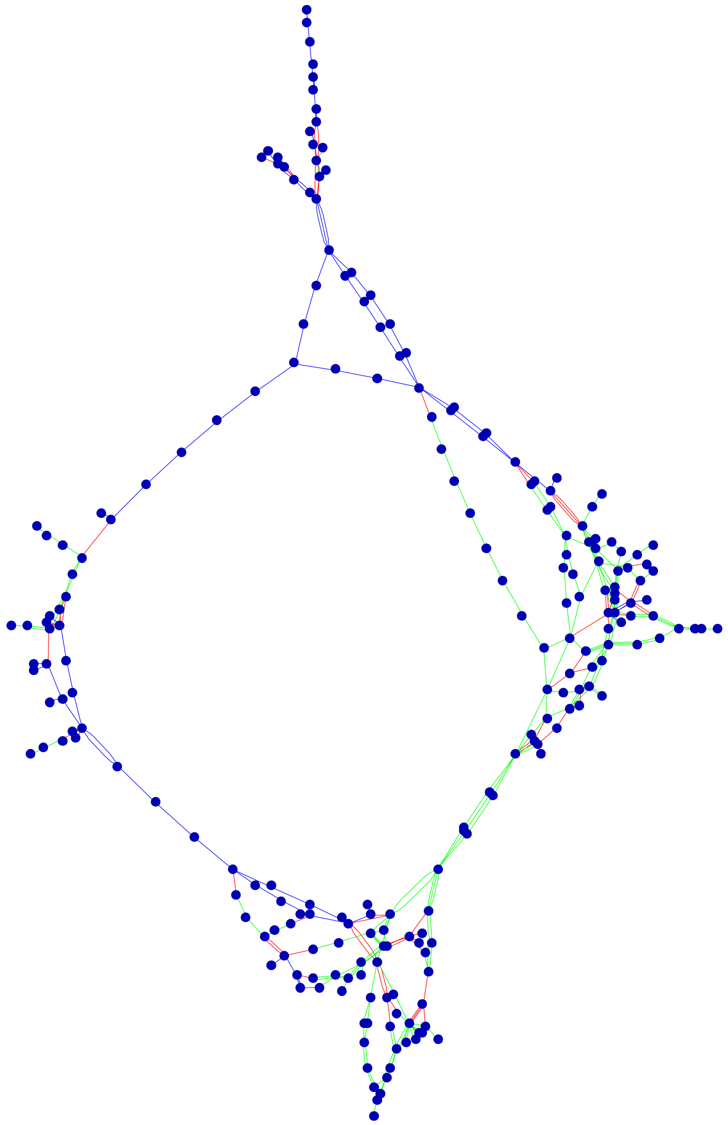
Lines being cut: 68



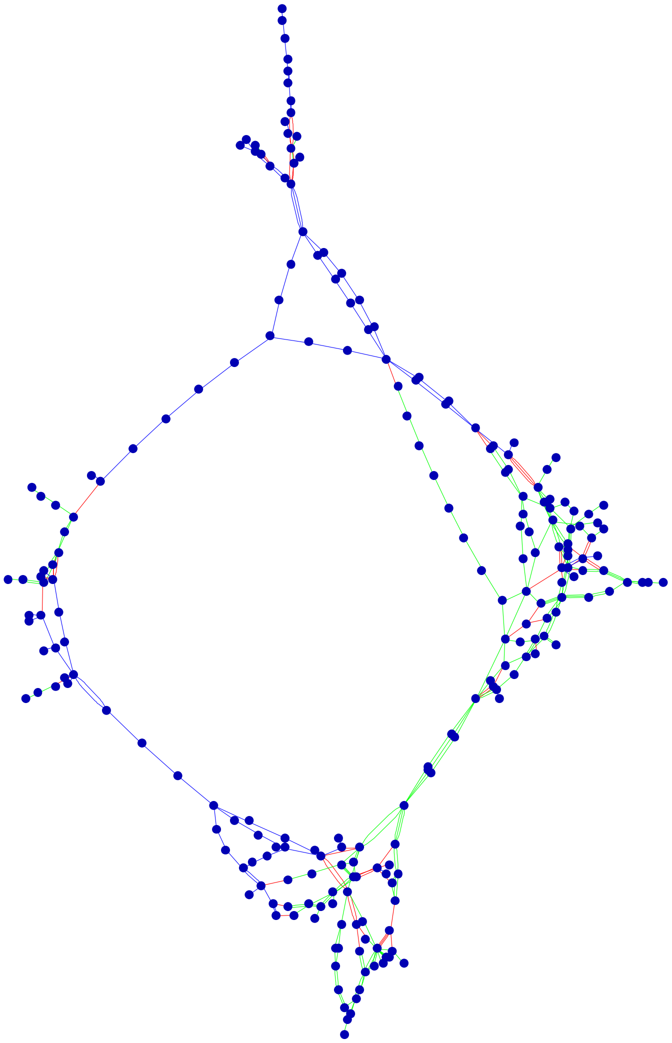
Lines being cut:55



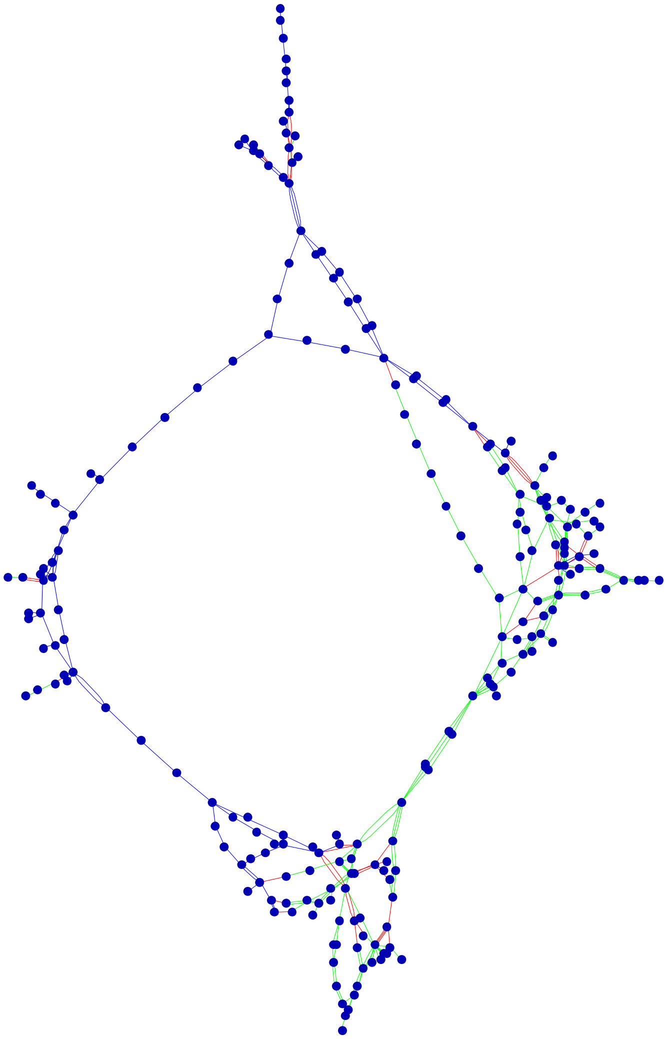
Lines being cut: 59



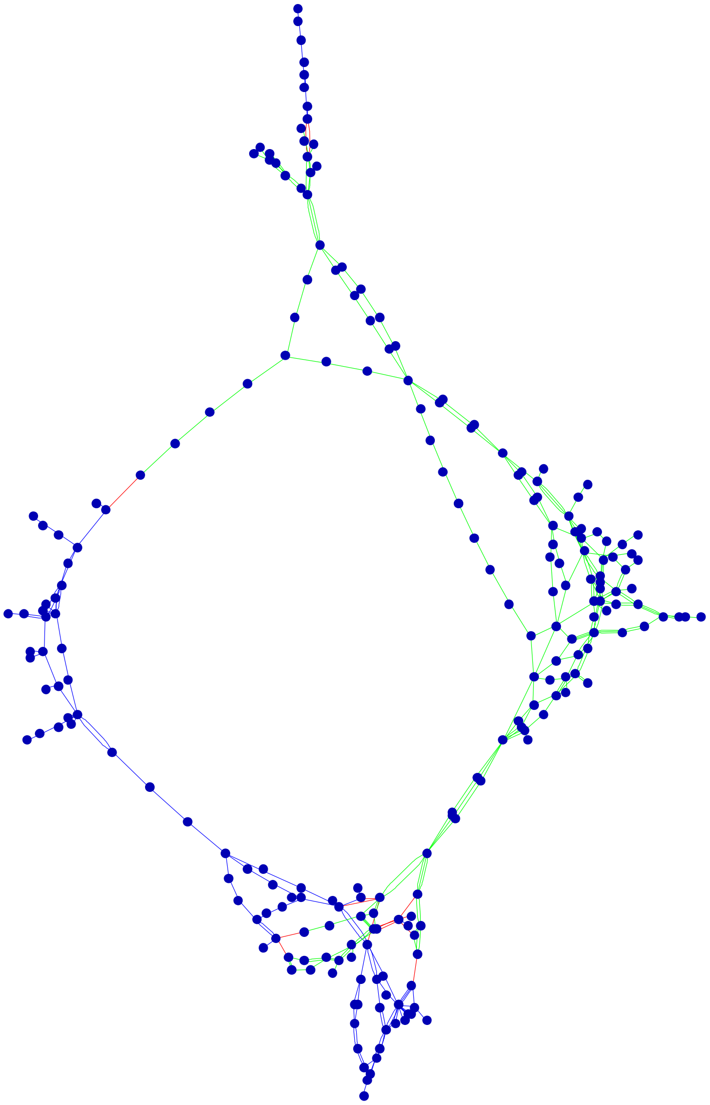
Lines being cut: 50



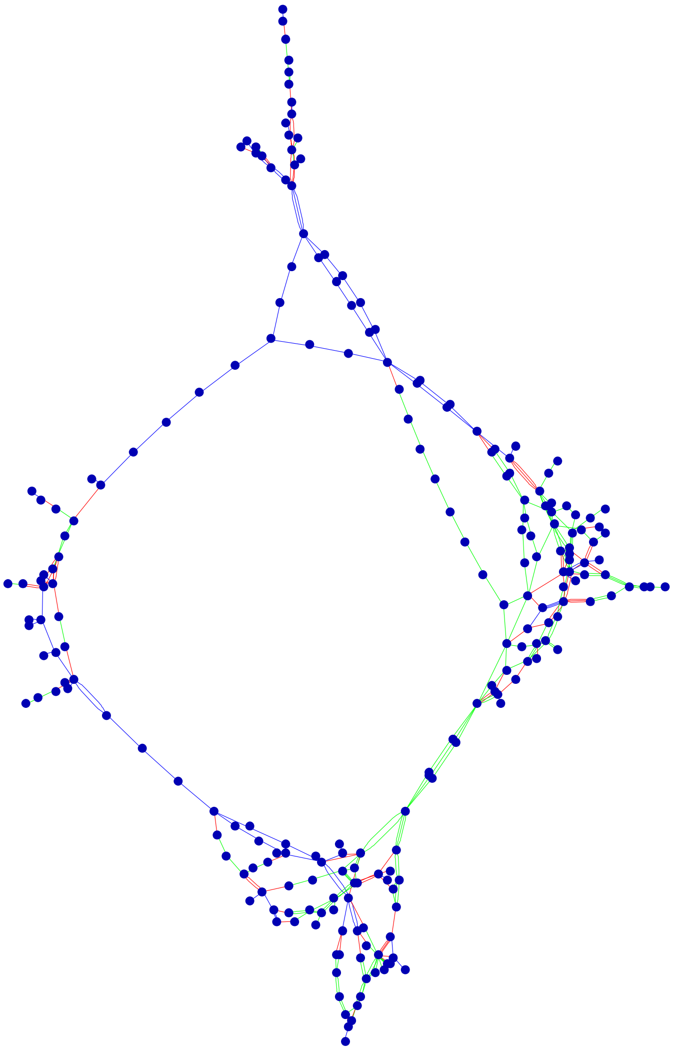
Lines being cut: 42



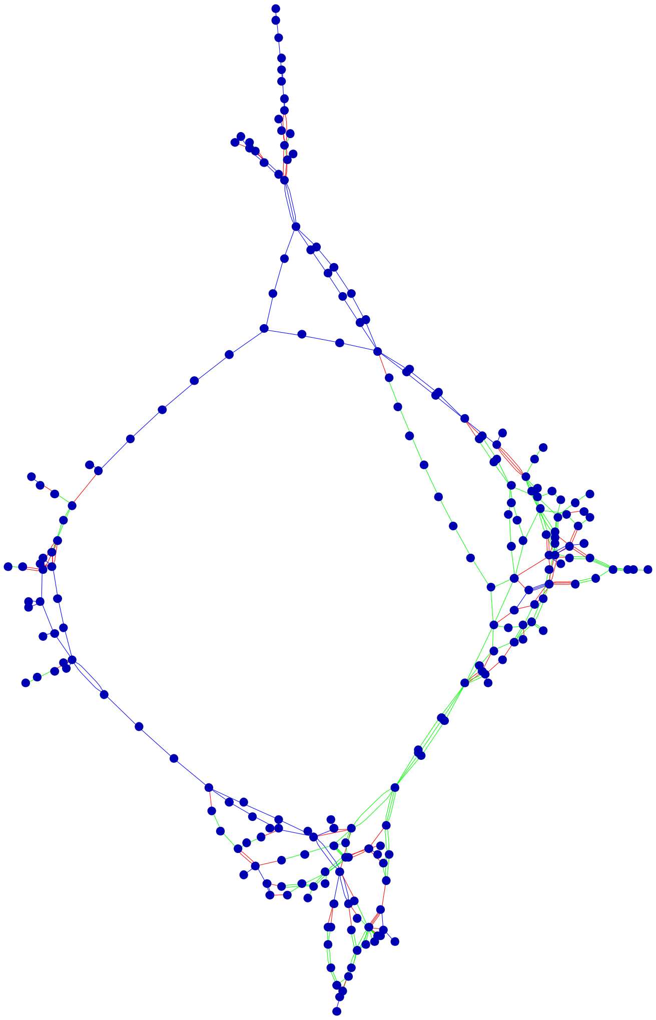
Lines being cut: 13



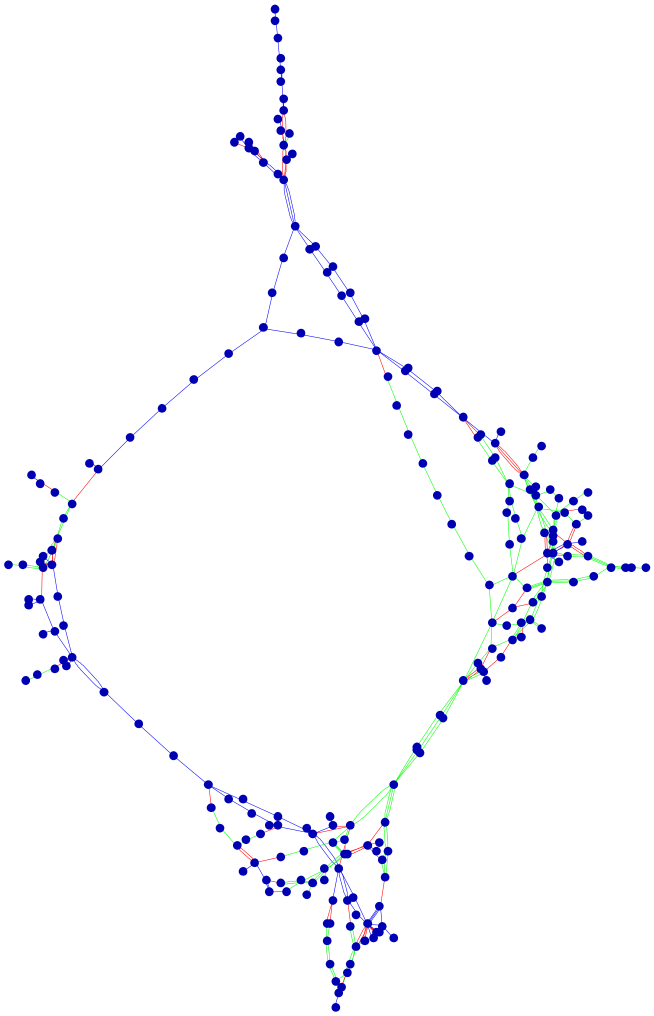
Lines being cut: 76



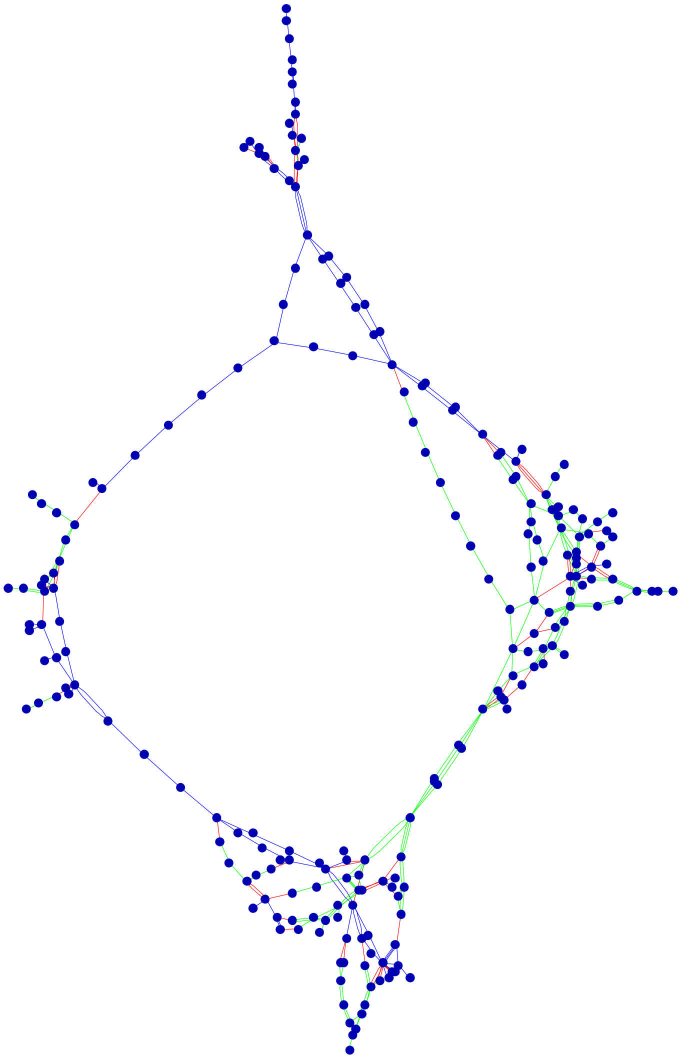
Lines being cut: 72



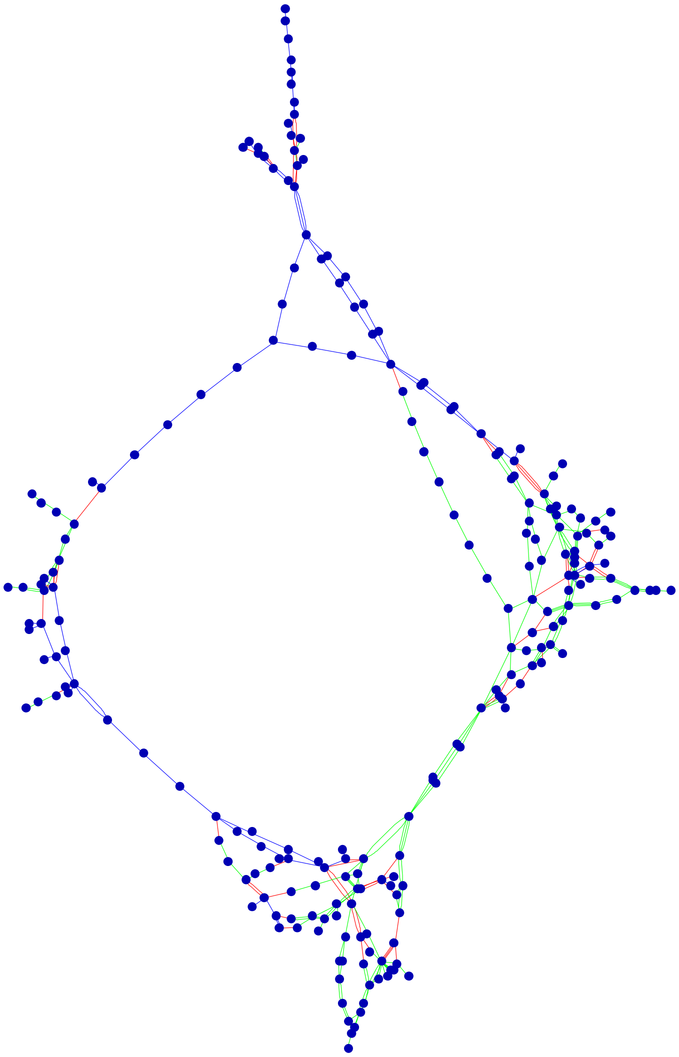
Lines being cut: 65



Lines being cut: 61



Lines being cut: 58



Lines being cut: 46