A COMPUTATIONAL ANALYSIS OF THE OPTIMAL POWER FLOW PROBLEM

BAHA ALZALG, CATALINA ANGHEL, WENYING GAN, QING HUANG, MUSTAZEE RAHMAN, AND ALEX SHUM

Abstract. The optimal power flow problem is concerned with finding a proper operating point for a power network while attempting to minimize some cost function and satisfy several network constraints. In this report we analyze the optimal power flow problem subject to contingency constraints, which demands that there be enough power to meet demands in the network in the event of contingencies such as a fault in some transmission line or the loss of a generator. We develop numerical algorithms to handle this problem for the $e-1$ security constrained optimal power flow problem. We also investigate the relationship between the cost of the optimal power flow problem and network topology. We find that when the network topology is that of a small world graph or a scale-free graph, the optimal power flow problem is quite robust in terms of satisfying contingency constraints.

Contents

1. Introduction

Modern electrical networks are complex structures whose effective operation is vital towards tasks as minute as everyday chores to large scale industrial activities. As such it is necessary to design these networks so that they can meet demands even under various contingencies, such as the breakdown of a generator or a fault in some transmission line. The importance of being able to supply power under these contingencies was exemplified by the blackout in the North American power grid during 2003, the largest ever experienced in North America, that affected about 50 million people in parts of Ohio, Michigan, New York, Pennsylvania, New Jersey, Connecticut, Massachusetts, Vermont, as well as Ontario and Québec for 2 to 7 days. The reason for the blackout was a failure of only a handful of contingencies in the network [8].

In this article we address the issue of efficient transmission of power in power networks under such contingency constraints. We also focus on the effect of the network topology on the efficient transmission of power. The mathematical model for the transmission of power through a network, given by the power flow equations, is well understood. In Sections 2 we introduce the power flow equations along with the related optimal power flow problem that deals with transmission of power through a network under the objective of minimizing a cost function.

In Section 3 we deal with the optimal power flow problem under certain contingency constraints and demonstrate the effect of contingencies on the minimal cost. To this end we develop an algorithm that can compute the minimal cost for the optimal power flow problem under contingencies given the network data.

Section 4 is devoted to the study of the relationship between a power network's topology and the optimal power flow problem (with and without contingencies). We explore how the minimal cost of the optimal power flow problem varies depending on certain network parameters such as its average degree, average path length and algebraic connectivity.

Acknowledgements

This work was conducted at the IMA while the authors were participating in the workshop 'Mathematical Modeling in Industry XV'. The authors are grateful to Chai Wah Wu, the team's mentor, for his guidance throughout the project and for his insights into the problem. The authors would also like to thank the IMA and the organizers of the workshop for their hospitality.

2. Background

2.1. **The power flow equations.** A power network with n buses and e branches is represented by a weighted graph $\mathcal{N} = (V, E)$ where the buses in V are identified with $\{1, 2, \ldots, n\}$. Each bus is connected to a **load** and the first m buses are connected to **generators** as well. For each branch $e = uv$, there is a complex valued weight $y_{u,v} = y_{v,u}$ at e with the convention that $y_{u,v} = y_{v,u} = 0$ if there is no branch connecting bus u to v. The weights $y_{u,v}$ are called **admittances** and each bus u also has an admittance to the ground denoted by y_u. Given the admittances one defines the **admittance matrix** Y for the power network as

$$Y_{uv} = \begin{cases} -y_{uv}, & \text{if } u \neq v \\ y_u + \sum_{v:v\neq u} y_{uv}, & \text{if } u = v \end{cases} \tag{2.1}$$

Figure 2.1 shows the model of a power network with 4 buses and branches.

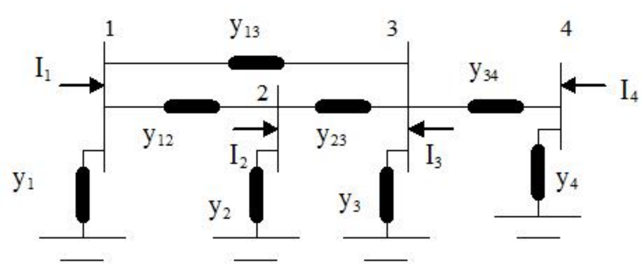

Figure 2.1. A power network with 4 buses and branches. Taken from `class.ece.iastate.edu/ee458/PowerFlowEquations.doc`.

Given a power network the following quantities describe the load and generator attributes at the buses. For each $l \in \{1, \ldots, m\}$ and $u \in \{1, \ldots, n\}$:

- P_u^d and Q_u^d are the real and reactive power demands at bus u respectively. These are given demands that must be met in order to have a proper power supply throughout the network.
- P_l^g and Q_l^g are the real and reactive powers generated at bus l respectively. These are variables.

The **voltage** at bus u is denoted by $V_u \in \mathbb{C}$, and $V = [V_1, \dots, V_n]^T$ is the column vector consisting of all the voltages. Similarly, the **current** injection at bus u is denoted by I_u and $I = [I_1, \dots, I_n]^T$ represent the vector of all currents in the network. The relation governing the currents and voltages is

$$I = YV\,. \tag{2.2}$$

One defines the **net power injection** at bus u to be $S_u = V_u I_u^*$ where $*$ denotes complex conjugation. Throughout this report $i = \sqrt{-1}$ denotes the imaginary unit. The **power flow** equations for the network state that

$$\begin{aligned} (P_l^g - P_l^d) + i(Q_l^g - Q_l^d) &= V_l I_l^* \quad \text{for} 1 \le l \le m && (2.3)\\ -P_u^d - iQ_u^d &= V_u I_u^* \quad \text{for} m+1 \le u \le n\,. && (2.4) \end{aligned}$$

The power flow equation (2.3) are often written using real variables. To this end write $V_u = |V_u|e^{i\theta_u} = |V_u|(\cos(\theta_u) + i\sin(\theta_u))$, $Y = G + iB$, and adopt the convention that $P_u^g = 0$ if $u > m$. Simplifying (2.3) gives

$$\begin{aligned} (P_u^g - P_u^d) + i(Q_u^g - Q_u^d) &= V_u\Big(\sum_{w=1}^{n} Y_{uw} V_w\Big)^* \\ &= |V_u|e^{i\theta_u} \sum_{w=1}^{n} (G_{uw} - iB_{uw})(|V_w|e^{-i\theta_w}) \\ &= \sum_{w=1}^{n} |V_u||V_w|e^{i(\theta_u - \theta_w)}\,(G_{uw} - iB_{uw}) \\ &= \sum_{w=1}^{n} |V_u||V_w|\,(\cos(\theta_u - \theta_w) + i\sin(\theta_u - \theta_w))\,(G_{uw} - iB_{uw}). \end{aligned}$$

Separating the equations into its real and imaginary parts one obtains

$$\begin{aligned} P_u^g - P_u^d &= \sum_{w=1}^{n} |V_u||V_w|\,(G_{uw}\cos(\theta_u - \theta_w) + B_{uw}\sin(\theta_u - \theta_w)) && (2.5)\\ Q_u^g - Q_u^d &= \sum_{w=1}^{n} |V_u||V_w|\,(G_{uw}\sin(\theta_u - \theta_w) - B_{uw}\cos(\theta_u - \theta_w))\,. && (2.6) \end{aligned}$$

In equations (2.5) and (2.6) there are 4 sets of variables, P_l^g, Q_l^g for $i \le l \le m$ and $|V_u|, \theta_u$ for $1 \le u \le n$. The goal is to solve these non-linear equations in the given variables. The approach for solving the power flow equations is to use an iterative algorithm. In commercial programs the Newton-Raphson algorithm is commonly used.

2.2. **The optimal power flow problem.** The objective in the optimal power flow (OPF) problem is to find an optimal operating point of the network that minimizes an appropriate cost subject to certain constraints on loads, powers, voltages variables and branch flows. The cost may be, for example, generation cost or transmission loss. The OPF problem has been extensively studied in literature beginning with the work of Carpentier [2] in 1962. Several algorithms have been developed to tackle this highly non-linear problem including Newton-Raphson, quadratic programming, non-linear programming, LP relaxations and various interior point methods (see [4, 5, 6] and the references cited therein). The OPF problem is in general NP-hard [7].

The classical OPF problem is formulated as

$$\begin{array}{ll}
\min_{V,\,P_g,\,Q_g} \ \sum_{l=1}^{m} f_l(P_l^g) & \text{subject to} \\
 & \\
S_l I_l^* = (P_l^g - P_l^d) + i(Q_l^g - Q_l^d) & \text{for } 1 \le l \le m \\
S_u I_u^* = -P_u^d - iQ_u^d & \text{for } m+1 \le u \le n \\
 & \\
P_l^{min} \le P_l^g \le P_l^{max} & \text{for } 1 \le l \le m \\
Q_l^{min} \le Q_l^g \le Q_l^{max} & \text{for } 1 \le l \le m \\
V_u^{min} \le |V_u| \le V_l^{max} & \text{for } 1 \le u \le n \\
\theta_u^{min} \le \theta_u \le \theta_u^{max} & \text{for } 1 \le u \le n
\end{array}$$

Other constraints are possible as well. For example, we could impose bounds on the voltage angle differences: $|\theta_u - \theta_w| \le \theta_{\max}$ for all $1 \le u, w \le n$. The cost functions $f_l(P_l^g)$ are typically taken to be polynomials (often quadratic polynomials).

2.3. **The MATPOWER package.** The research and modeling of the power flow and OPF problems were done entirely using the MATPOWER toolbox version 4.0 in MATLAB [10]. The MATPOWER toolbox was created by Zimmerman and Sánchez to numerically handle the power flow equations and the OPF problem among other things. A detailed manual is available on their website at `http://www.pserc.cornell.edu/matpower/`. The toolbox is capable of modeling complex networks, providing properties of buses, branches, generators, generator costs and loads. These quantities are all given in matrices that can be easily modified. The power flow cost problem is solved using the Newton-Raphson method, while the OPF problem has several different solvers. The solver is set to the default MATLAB Interior-Point Solver (MIPS), and was used to solve all the OPF problems. The majority of the produced code focused on building the corresponding matrices in our problems for MATPOWER to solve.

A variety of different case files were provided with the toolbox that are representative of real networks. Several of them were tested and their results are shown in this report report. The format is simple enough to build our own examples. The following functions were used extensively to obtain the data shown throughout this report.

```
r = runpf(casefile,options)
```

The function *runpf* solves the power flow equation (2.3) given the values of P_l^g and Q_l^g for the variables $V_u = (|V_u|, \theta_u)$. The input *casefile* takes in a structure with the necessary matrices for buses, generators and branches. The output r is a structure that includes the values for the variable V_u. The *options* input controls various parameters. The reader is encouraged to refer to the MATPOWER manual to see its descriptions.

```
r = runopf(casefile,options)
```

The function *runopf* solves the OPF problem. The inputs and outputs are the same format as *runpf*. Instead, the P_g and Q_g values are unknown, and a minimization is performed according to the given cost function. The output of interest in the structure r, are the cost, as well as the optimal supplies for each generator.

2.4. **Graph theoretic notions.** In the report, we try to correlate the ratio of the OPF cost of a network to a subnetwork obtained by removing one branch with several topological properties of the network. We also try to do the same for the OPF cost with contingency constraints. Hence in this section, we will introduce several graph theoretic definitions that we will use later.

Let $G = (V, E)$ be a simple graph. The first notion is that of algebraic connectivity of G. The algebraic connectivity reflects how well connected the overall graph is, and has been used in analyzing the synchronizability of networks. Meanwhile, we suspect that power generation cost is highly related to the connectedness of the underlying network topology. For this reason we discuss the algebraic connectivity. For a simple graph G, the graph Laplacian L_G is defined as:

$$L_G(u,v) = \begin{cases} -1, & \text{if } u \text{ is adjacent to } v\text{and } u \text{ is not the same as } v \\ \deg(v), & \text{if } u = v \\ 0, & \text{otherwise} \end{cases}$$

The algebraic connectivity is defined as the **second smallest eigenvalue** of L_G. The algebraic connectivity of G is greater than 0 if and only if G is connected. Usually the bigger the algebraic connectivity the better the graph is connected.

The second definition we need is the average path length of G. For each pair of vertices u, v in G, the distance between them, denoted $d(u, v)$, is the length of the shortest path connecting u and v. If u and v are not connected, the distance is infinite. The average path length $l(G)$ is the average distance for all possible pairs of different vertices in G. It is a measure of the efficiency of information or mass transport on a network. If n is the number of vertices in G, then

$$l_G = \frac{\sum_{u,v\in V} d(u,v)}{n(n-1)} .$$

The third notion is the average vertex degree. The degree of a vertex v in G is the number of edges it is connected with, we can write it as $deg(v)$. The average degree is the average degree over all vertices:

$$AvD_G = \frac{\sum_{v\in V} \deg(v)}{n} = \frac{2e}{n} .$$

Here e is the number of edges in G and n is the number of vertices.

3. Security constrained optimal power flow

In the report thus far, the power flow equations and the OPF problem have been presented. We now consider the OPF under contingency constraints as advertised. Suppose we are given a power network as described in Section 2.1; let n and e denote the number of buses and branches in the network respectively. It is of interest to investigate the effects of cutting a branch from the network. When branches are cut, internal power values in the branches and buses change, potentially causing power failures or generator and load disconnections. Generators may have to change their power supplies, and startup and shutdown costs can be expensive. There are several new problems that arise.

(1) Can power still be supplied to the loads in the new network? If so, what are the optimal power generations?
(2) Suppose we are interested in a specific branch being cut. Can the optimal power flow problem be solved simultaneously for both the original network and the new network? That is, we wish to find power supply values that can satisfy both the original and the sub-network resulting from one of the branches being cut. Is this even feasible?
(3) A tougher problem: what if we want uninterrupted power supply in both networks no matter which branch is being cut (i.e. one can remove exactly one branch but is it not known which one in advance).

Limiting factors would include the exclusion of generators and limits in the power transferable through the branches. The first problem is simply solving the optimal power flow problem in the context of a new network. A solution to the branch cutting problem (the first problem above) does not necessarily yield a feasible point in the original OPF problem. Simply raising the generator values might not satisfy the constraints for both problems. The power flow equations are highly nonlinear and the internal properties of the network must be considered. The second and third problems are more difficult. Our goal is to prevent a power failure should a branch be cut. It may seem wasteful to provide unnecessary power that is not used. In the setting of large scale electrical grids however, additional power should be supplied since the load demands constantly change. Our efforts have been focused on the second and third problems. To reiterate, the second problem is **not** the optimal power flow with a branch cut from the original network. The last two problems are known in literature as the $e-1$ security-constrained optimal power flow (SCOPF) problem.

3.1. **The single $e-1$ security-constrained OPF.** Recall the optimal power flow problem given in Section 2.2. Since we are dealing with two problems simultaneously, it is necessary to construct a new set of variables $\mathbf{x}' = [P^{g'}, Q^{g'}, V']^T$ that corresponds to the new problem on the sub-network resulting from a branch being removed. Next, we require that this new set of variables $\mathbf{x}'$ to satisfy its own corresponding power flow equations for the sub-network. Finally, we would like the corresponding powers in the generators to be the same (resulting in uninterrupted power flow). Since they are the same, it is not necessary to change the cost function from the original problem. To summarize, the SCOPF problem has the following additional constraints:

$$
\begin{aligned}
g_1(\mathbf{x}') &= 0 \\
h_1(\mathbf{x}') &\le 0, \text{ and} \\
P_l^{g'} &= P_l^g.
\end{aligned}
$$

Here g_1 represents the power flow equation 2.3 for the new sub-network, and h_1 represent the inequality constraints for the OPF problem on said sub-network.

In terms of implementation in MATPOWER, we perform the following adjustments to the original case file.

(1) Bus matrix: We add one duplication to the original bus matrix but remark the bus number from $n+1$ to $2n$ in the duplication island.
(2) Generator matrix: Add one duplication to the original generator matrix.
(3) Branch matrix: Add one duplication to the original branch matrix, change the bus number accordingly, and delete the branch row we want to cut.
(4) Generator cost matrix: No change.

One more thing to mention is that MATPOWER usually picks the first generator bus as the reference bus. When we make duplication, we keep the reference bus type in the duplicated part. The two islands are actually regarded as unconnected, we set a reference bus in each island which helps convergence.

Our goal is to identify which branch is the most problematic, which requires checking each $e-1$ contingency case and the resulting cost. The largest optimal cost shows that the corresponding branch is the most problematic. We would like to illustrate this through two examples.

Example 3.1. 9-bus case. The 9-bus case comes from MATPOWER. The network is shown in Figure 2. There are three generator buses and three nonzero load buses. Table 1 shows that the original uncut model has the lowest optimal cost. This is trivial because when we consider contingencies regarding the removal of a branch, we add constraints to the original OPF problem. The feasible region decreases, which leads to the increase of optimal cost value. Among these contingencies, we see that the removal of branches $(1,4)$, $(3,6)$, and $(8,2)$ leads to relatively higher optimal costs. These branches are all connected to generator buses which become isolated after the corresponding branches are removed. Understandably, the isolation (i.e. loss) of a generator is expensive for a network, and so the removal of branches that do so are relatively more problematic. We also observe that the contingency corresponding to branch $(8,2)$ being cut has the largest optimal cost (the most problematic branch cut). This indicates that the generator attached to

Branch Cut	Optimal Cost
No cuts	**5296.7**
(1, 4)	6682.6
(4, 5)	5331.2
(5, 6)	5885.4
(3, 6)	6846.3
(6, 7)	5330.7
(7, 8)	5525.9
(8, 2)	7935.4
(8, 9)	5576.8
(9, 4)	5587.4

TABLE 1. Numerical results for 9-bus case.

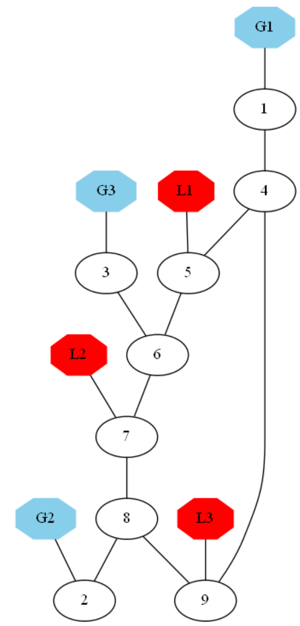

TABLE 2. Graph for 9-bus case.

bus 2 is most vital to this power network, and its removal incurs the most cost for meeting power demands.

Example 3.2. 14-bus case. The 14-bus case also comes from MATPOWER. The original network graph is shown in Figure 4. From the results in Table 3, we find that the removal of branches $(1, 5)$, $(2, 4)$, and $(4, 5)$ make the numerical solver for the SCOPF problem not converge (more precisely, the constraints added to satisfy the contingencies leads the MATPOWER solver to not converge). One possible reason is that there is no feasible solution satisfying all the constraints. For instance, the contingency of branch $(1, 5)$ being cut could make the network fail. The load demands can not be met simultaneously with the power flow constraints when this branch is cut. From the OPF solution of original non-contingency case, we have $P_{g,1} = 194.33$, $P_{g,2} = 36.72$, $P_{g,3} = 28.74$, $P_{g,6} = 0$, and $P_{g,8} = 8.5$. The power generation at bus 1 is the largest, which would explain why branch $(1, 2)$ being removed could lead to a higher cost.

3.2. **The full $e - 1$ security-constrained OPF.** In the previous part, we discuss the case with $e - 1$ contingencies of cutting a branch. Now we extend the idea to $e - 1$ contingencies of cutting any one branch, which is problem (3) mentioned at the beginning of this section. In reality, for a complex power network, it is possible that one of the branches may fail while we still need the power flow work properly.

The scheme of this problem is similar as the previous SCOPF model. But we extend to make e duplications since we want the system satisfying all $e-1$ contingencies. In terms of implementation of MATPOWER, the adjustment is shown as follows:

(1) Bus matrix: We add e duplications to the original bus matrix but remark the bus number starting from $n + 1$ to $n(e + 1)$ for these duplicated islands.
(2) Generator matrix: Add e duplications to the original generator matrix.
(3) Branch matrix: Add e duplications to the original branch matrix, change the bus number accordingly with bus matrix, and delete one different branch in each duplicated island respectively.
(4) Generator cost matrix: No change.

Branch Cut	Optimal Cost
No cuts	**8081.53**
(1, 2)	9126.1
(1, 5)	Does not converge
(2, 3)	8268.0
(2, 4)	Does not converge
(2, 5)	8110.3
(3, 4)	8081.6
(4, 5)	Does not converge
(4, 7)	8085.3
(4, 9)	8082.3
(5, 6)	8119.9
(6, 11)	8086.0
(6, 12)	8090.4
(6, 13)	8116.1
(7, 8)	8086.4
(7, 9)	8095.6
(9, 10)	8087.4
(9, 14)	8102.0
(10, 11)	8082.2
(12, 13)	8081.8
(13, 14)	8085.4

TABLE 3. Numerical results for 14-bus case.

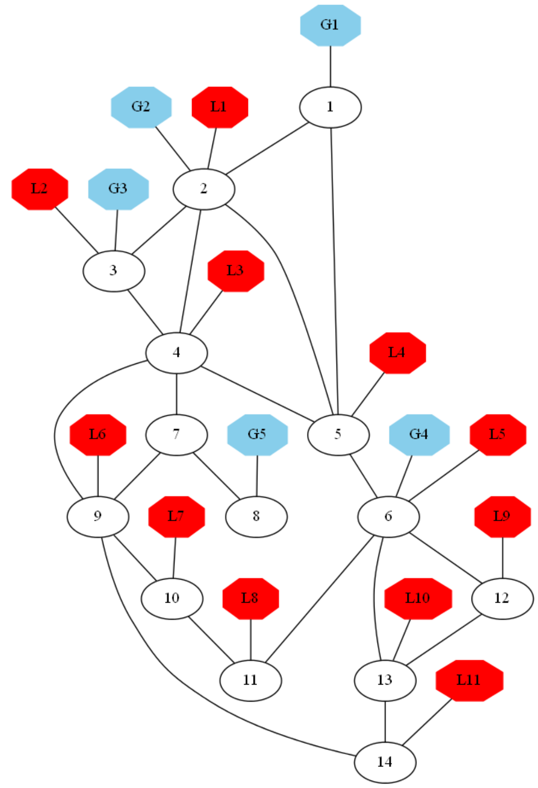

TABLE 4. Graph for 14-bus case.

Specifically speaking, suppose we have $n = 9$ buses, $e = 10$ branches and first three buses are generators. Let $P_{g,1}$, $P_{g,2}$, and $P_{g,3}$ denote the power injection for the three buses. We do the duplication for 10 times and delete one different branch in each duplication respectively to build new branch matrix. Note we use different bus number for each duplication. New constraints are added as $P_1 = P_{10} = \ldots = P_{91}$, $P_{g,2} = P_{g,11} = \ldots = P_{g,92}$, and $P_{g,3} = P_{g,12} = \ldots = P_{g,93}$. In our model, these constraints are relaxed to $P_{g,10} \le P_{g,1}, \ldots, P_{g,91} \le P_{g,1}$ and the same for the other above two constraints. This kind of relaxation is reasonable because if we satisfy that the original power injection of generator buses is no less than the power injection after cutting one branch, we can make sure the power flow network operates successfully. Also, we keep other constraints the same as the previous model.

We work with the network corresponding to the complete graph (see the last graph on Figure 4.1 for a complete graph with 6-buses). Such a network is fully connected and cutting one branch has the least effect on the network compared with other networks with the same number of buses.

Example 3.3. Suppose we have 6 buses, 3 of which are generator buses and 3 of which are load buses (see Figure 4.1). The loads are set as $P_{d,2} = P_{d,4} = P_{d,6} = 100$ and $Q_{d,2} = Q_{d,4} = Q_{d,6} = 35$. The objective cost function is $f(P) = \sum_{i=1}^{3} 0.11P_{g,i}^2 + 5P_{g,i} + 150$. Using MATPOWER, the solution of SCOPF problem is that $P_{g,1} = 100.02$, $P_{g,2} = 100.19$, $P_{g,3} = 99.79$ and the cost is 5250.1. Therefore, for this complete network, we could allocate the power injection so that the network could works no matter any of one branch fails.

We conducted more tests for complete networks with the number of buses equal to 8, 10, 12, 14, and 16, and optimal solutions can be obtained in all these cases. For future work we will look into less complete cases or practical cases. But things become more complicated and it is possible that all $e-1$ contingencies of any one branch cut could make the network fail. For Example 3.2, we see that the network fails for $e-1$ contingencies for some branches. It is obvious that all $e-1$ contingencies could not make the network work. Also, we need to be careful when dealing with new isolated buses after cutting one branch. When a load bus with a load required becomes isolated after cutting one branch, then the system fails. When a generator bus becomes isolated, more power is needed from other generators which involves the maximum capacity of other generators. Depending on the network, more analysis is needed when dealing with SCOPF problems.

4. Effects of network topology

We try to relate the topology of the network and the optimal cost function. We are modeling the network as a graph where the buses are the nodes or vertices, and the branches are the edges. We would like to find features of the graph which would most affect the cost function in order to identify problematic contingencies.

4.1. **SCOPF for cycle, bipartite, and complete graphs.** Since it is difficult to isolate interesting features in complicated graphs, we begin by considering three simple graphs: the complete, bipartite and cycle graphs. We denote by C_n , $K_{n,m}$ and K_n the cycle with with n vertices, the bipartite graph with n vertices in one part and m vertices in the other part and the complete graph with n vertices respectively (see Fig. 4.1). We set the properties of these graphs to be as symmetric as possible; for instance there are as many generators as load, all generators have the same real power output and all the loads have the same power demand, all branches connecting loads to generators (resp. generators to loads and loads to loads) have the same attributes, etc.

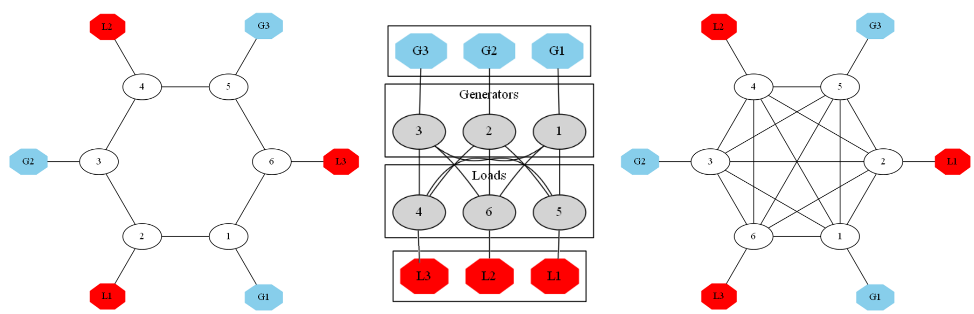

Figure 4.1. Toy networks: The cycle C_6, bipartite graph $K_{3,3}$, and complete graph K_6.

The first thing we need to stress is that there is no global connection between network topology and the minimal OPF cost. The reason is that there are two different kinds of nodes in a power network: generators and loads. But there is only one kind of vertex in a graph. If we change the type of nodes in a power network, the minimal cost will change but the network topology will remain the same. Here is an example.

Example 4.1. Consider the two 6-cycles in Figure 4.2. In the first power network, we let the 1,2,3 be generators and 4,5,6 be loads. But in the second power network, we let 1,3,5 be the generators and 2,4,6 be the loads. For the bus data, we induce all generators with the same data, and similarly for all the loads. The branch data is the same for all branches. As to the generator cost, we set it to be a quadratic polynomial; the same for all generators. All bus, branch, and generator data along with costs were taken from the 9-bus network in example 3.1. After running the two models, the minimal OPF cost in the first model is \$5449.83/hr while the optimal power cost in the second model is \$5385.84/hr. But the graph structures for the two graphs are the same. Hence, we should focus on the local connection between the topological structure and optimal power cost.

For all three toy networks, we consider the single $e-1$ SCOPF that results from removing a branding connecting a generator to a load. As would be expected, a branch cut has the greatest

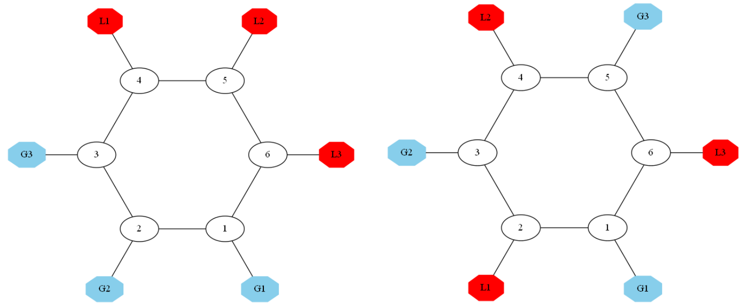

FIGURE 4.2. Two different models for C_6.

effect on the cost for the cycle graph (which becomes a path graph), and the least effect in a complete graph (where there still many ways to distribute power). Also, the ratio of optimal cost for the contingency case versus the original case tends to 1 for all graphs, with the slowest convergence for the cycle graph (see Figure 4.3).

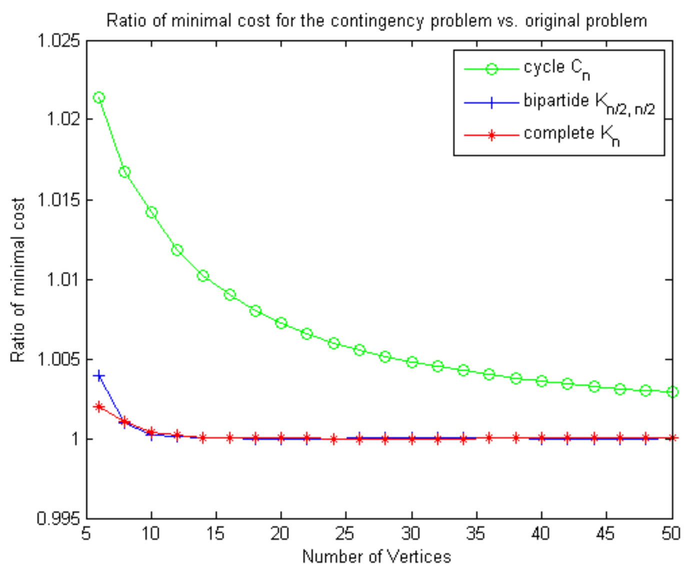

FIGURE 4.3. The ratio of optimal cost for the contingency case when one branch is dropped versus the original graph for cycle, bipartite, and complete graphs.

We can compare the ratio of minimal cost for the contingency case and the optimal cost for the original case with the ratio of the algebraic connectivity of the original graph and the graph with one branch removed (see Figure 4.4).

For the complete graph K_n and the complete bipartite graph $K_{n,n}$, the bigger n becomes the better the graph is connected, and the smaller the difference between the algebraic connectivity of the original graph and the sub-graph with one branch removed. As $n \to \infty$, the ratio of the algebraic connectivity of the original graph to the sub-graph with one branch removed decreases to 1. But the cycle graph C_n is comparably easier to disconnect via edge removal for large n. As $n \to \infty$, the ratio of the algebraic connectivity of the original graph to the sub-graph with one branch removed is increasing. After some calculation, it is easy to prove that the limit is 4. After comparing the two groups of pictures, we can see the trend in the change of algebraic connectivity and the change of optimal power costs will not remain the same for different graphs.

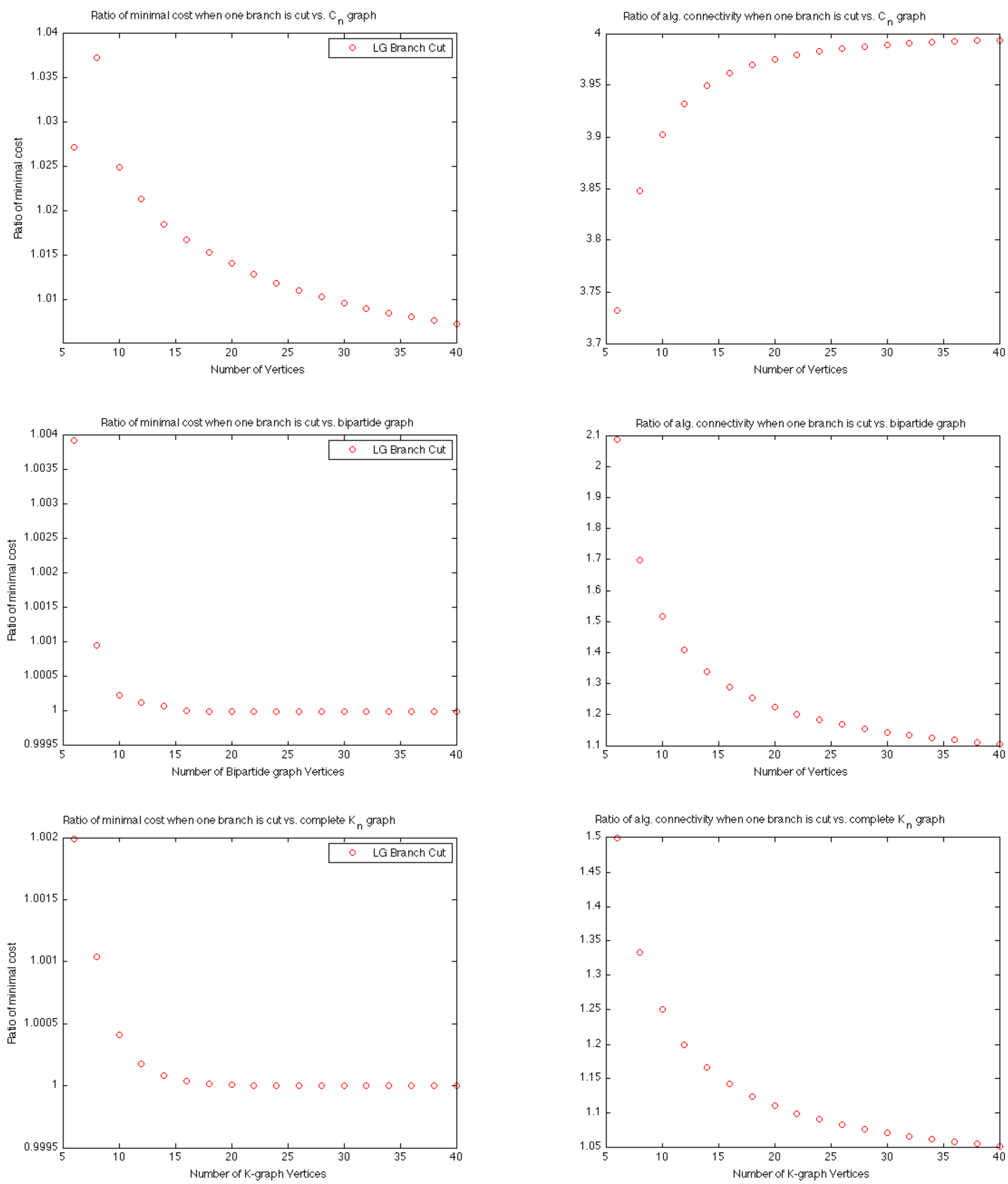

FIGURE 4.4. Optimal cost and algebraic connectivity ratios for the contingency case of one branch being removed v.s. the full graph, v.s. the number of buses.

4.2. **OPF on small world graphs.** We now investigate the OPF problem on the so-called small world graphs that were introduced by Watts and Strogatz [9]. A small world graph is constructed upon a connected graph with a large graph diameter (the largest distance between any two vertices). For example, the P_n and C_n graphs have graph diameters $n-1$ and $\lfloor \frac{n}{2} \rfloor$ respectively. By randomly adding a small number of edges to these graphs, the graph diameter decreases significantly. We expect that as connectivity increases in the graph, the optimal power flow cost will decrease. From our previous discussion, it was found that load-generator connections produced the largest change in OPF, while the other two types of connections produced little change in comparison. We limit our discussion to the addition of load-generator branches.

Construction. We start with a C_n graph that is built with alternating generator and load buses as described previously. Given a probability $0 \leq p \leq 1$, we add $\lceil p(LG) \rceil$ randomly chosen load-generator branches, where LG is the total number of possible load-generator branches that can be added. These branches are added to the network, and the OPF is solved. To produce a collection of data, p is chosen to be different values and the problem is solved multiple times. We note that this is not a general configuration, but we expect similar results for C_n graphs with different load/generator configurations. Since average degree is proportional to the number of branches LG that we can add, it is necessary to consider only one of these variables. We choose to investigate the relationship between OPF cost with the measures of average degree, average path length and algebraic connectivity. We note again, the problem considers only a local measure of algebraic connectivity, because it is specific to the layout for our cycle graph.

Results. The branch data was discovered to play a major role in the data. To have control over the data set, the same branches and same buses were chosen to connect the network. In an ideal network, where the branches have no resistive (real) component r. The reactive (imaginary) component x is held as a constant in all these data sets at $x = 0.2$. All electrical values are given per unit. No power is drawn in the wires, and changing any of the three measures have no effect (see Figure 4.5). This is not surprising because there is no voltage drop across the branches, so power is free to flow throughout any of the branches. Increasing connectivity does not decrease cost because the cycle is already an optimal configuration. This is however not realistic of branches in application. Manufactured branches contain a complex load $y = r + ix$, with $r > 0$, as well as additional line charging susceptance (reactive) components that are connected in parallel and described by a variable b.

We find that introducing a resistive load while keeping the factor b small in the branches gives the results we expected (see Figure 4.6). Increasing the connectivity (both in algebraic connectivity and the average node degree) results in a lower cost. The cost increases as the average path length increases, as expected.

An interesting phenomenon occurs when b is increased. The same effects occur with average degree and algebraic connectivity, so the discussion will be limited to average degree (see Figure 4.7). While keeping $r > 0$, with $r << x$, the value of b is increased. The optimal power flow cost now decreases until a certain point, before increasing again. This is due to the reactive power demand required by the wires. Adding branches to the C_n graph configuration assists connectivity, and thus decreases the cost until an "optimal connectivity' is reached (the minimum seen when $b = 0.25$ and $b = 0.4$). Adding more branches to the network beyond this optimum only adds more loads through wire connections, increasing the cost. Recall that increasing b yields a higher reactive power demand which results in a higher generator power demand. It can be seen that increasing b (as seen when $b = 0.7$) too much will make this minimum disappear, and any additional branches will simply raise the cost.

The opposite effect occurs for an average path length comparison (see Figure 4.8). For large enough b, increasing average path length increases the cost, which is characteristic of the plots seen with average degree. Higher average path length corresponds to lower connectivity. These trends can be modeled and can be fitted to functions.

This approach was performed for the 9-bus case file provided by MATPOWER representing a real-world network. Though it is hard to perform the same experiment, given that the bus data

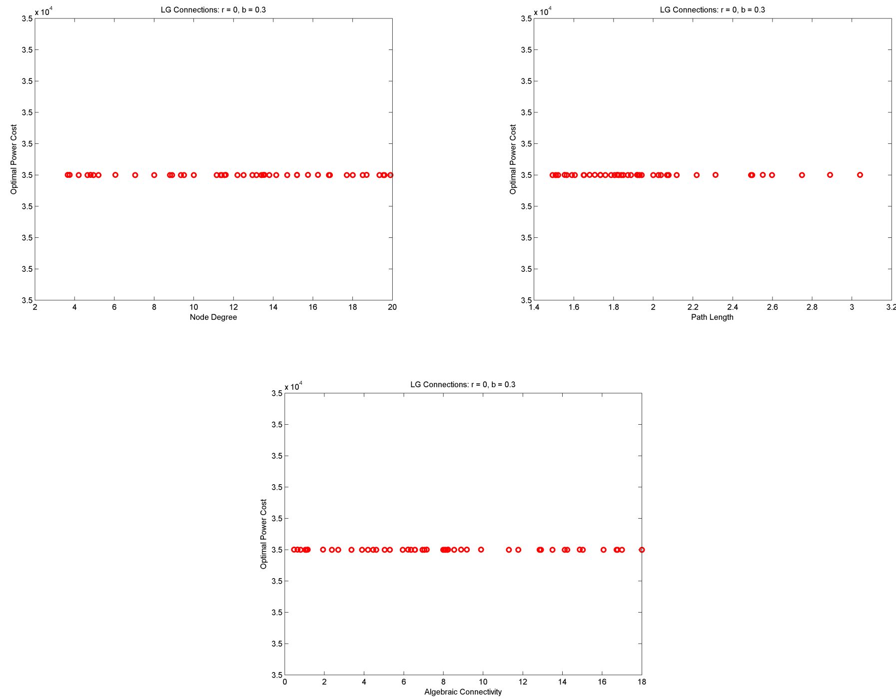

FIGURE 4.5. Ave. degree (left), aver. path length (right), and algebraic connectivity for zero-resistive element in branches

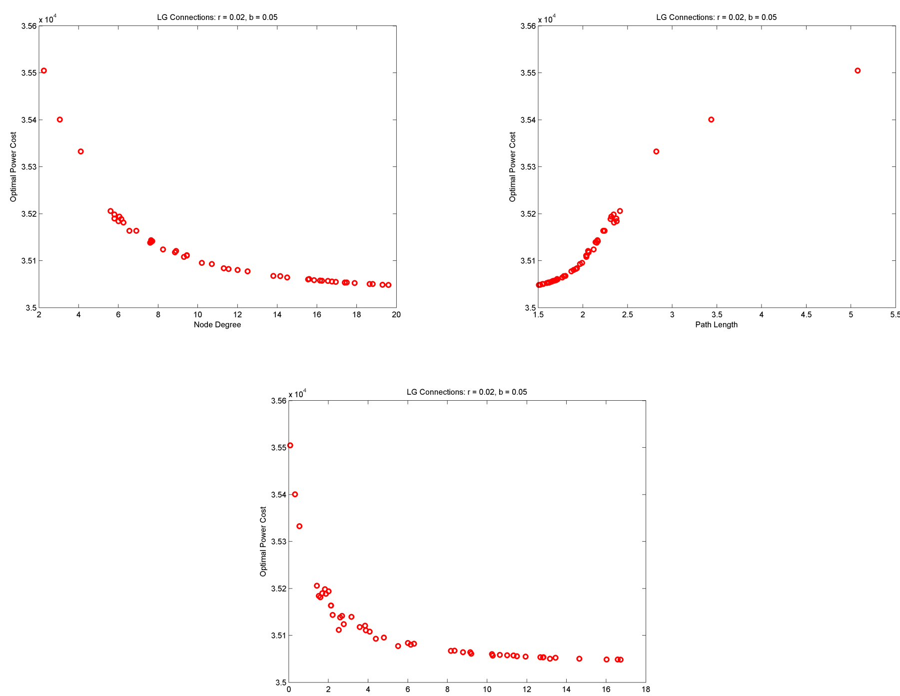

FIGURE 4.6. Ave. degree, ave. path length, and algebraic connectivity for r = 0.02, b = 0.05.

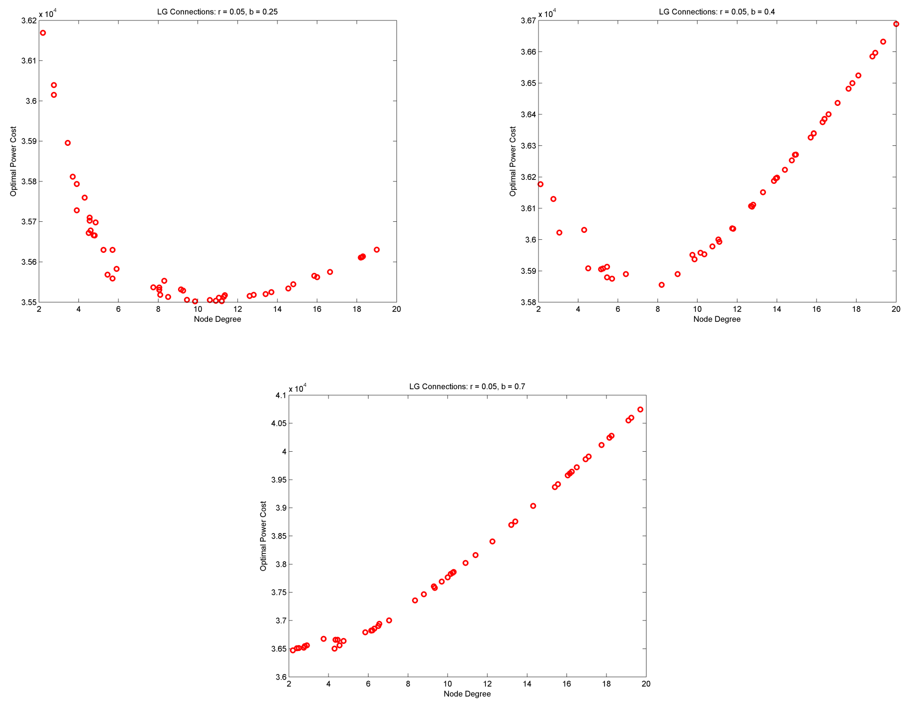

FIGURE 4.7. Ave. degree for $r = 0.05$, and $b = 0.25$ (left), 0.4 (right), and 0.7 (bottom).

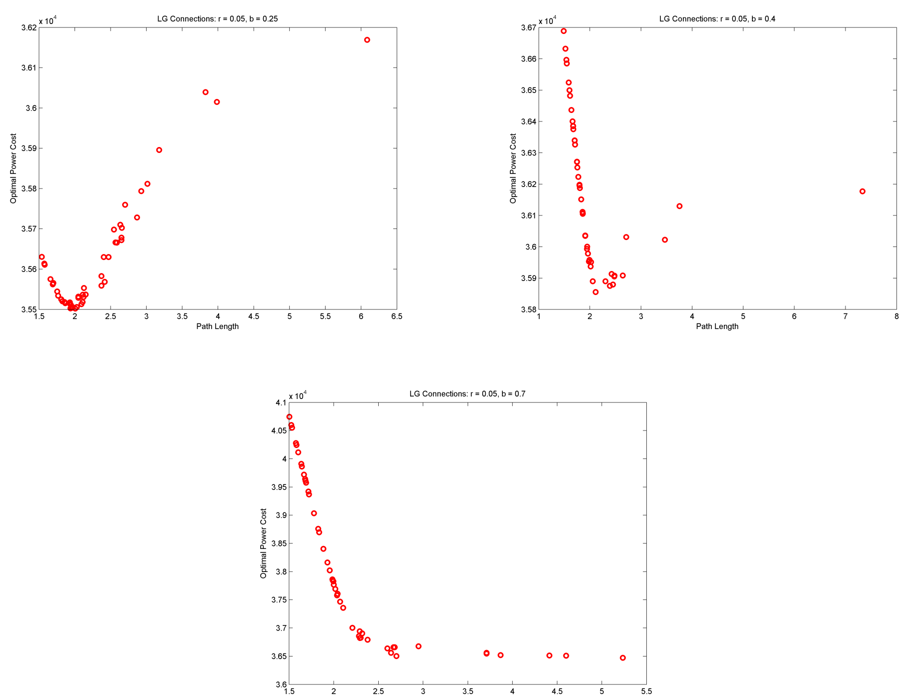

FIGURE 4.8. Ave. degree for $r = 0.05$, and $b = 0.25$ (left), 0.4 (right), and 0.7 (bottom).

is different for each bus, the general trends that occurred before seem to appear again (see Figure 4.9).

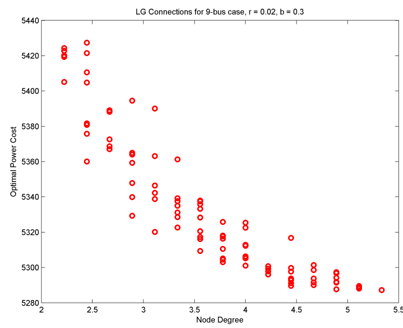

FIGURE 4.9. Ave. degree, for 9-bus case, with $r = 0.05$, and $b = 0.3$.

4.3. **OPF on scale free graphs.** A scale-free graph refers to a sequence of graphs with the property that the limiting degree distribution of the vertices follows a power law. More precisely, given a sequence of graphs $\{G_n\}_{n\geq 0}$ (usually random) with $\#V(G_n) \to \infty$, let

$$P_n(d) = \frac{\#\{v \in V(G_n) : \; deg(v) = d\}}{\#V(G_n)}\,.$$

The sequence of graphs $\{G_n\}$ is scale-free if for some $\gamma > 0$

$$\lim_{n\to\infty} P_n(d) = cd^{-\gamma} \text{ for all sufficiently large } d\,.$$

Many large real-world graphs have such a power law degree distribution where γ is usually in the range $2 < \gamma < 3$. Examples include the degree distribution of internet sites, social and collaboration networks, and sexual partners in humans. More relevantly, the electrical power grid of the United States may be modeled as a graph with 4941 vertices, corresponding to the generators, transformers and substations, and the edges corresponding to the high-voltage transmission lines. The degree distribution follows an approximate power-law with exponent $\gamma \approx 4$ [1, 3].

Albert and Barabasi introduced a random graph model, called the preferential attachment model, that generates graphs with the scale-free property [1]. We use their procedure to construct examples of power networks with the scale-free property, and consider the OPF problem on these networks.

Construction.

- Input a small graph G_0, such as a cycle, biparite or complete graph from Section 4.1 and fix an integer parameter $k \geq 1$ corresponding to the number of edges to be added at each stage of the construction.
- We alternate between adding generator and load buses to the graph in order to expand it. So having constructed G_n for $n \geq 0$, we add a new generator bus to G_n at stage $n+1$ if at stage n a load bus was attached to it; otherwise we attach to load bus to G_n. For concreteness, we attach a generator to G_0.
- Having added a new bus to G_n, say a generator for concreteness, we attach k branches from the new bus to the load buses in G_n using preferential attachment (the branches are attached to generator buses of G_n if the new bus is a load). Preferential attachment is done as follows. The endpoint of the first branch is vertex u_1 of G_n with probability proportional to $deg_{G_n}(u_1)$. Similarly, the second branch has endpoint $u_2 \neq u_1$ with probability proportional to its degree in G_n. To ensure that $u_2 \neq u_1$, we sample the vertices of G_n repeatedly

according to the aforementioned probability until we pick $u_2 \neq u_1$. The endpoints $u_3, \ldots, u_k$ are chosen analogously by sampling repeatedly to ensure $u_i \notin \{u_1, \ldots, u_{i-1}\}$.

- After the k branches are attached we call the resulting graph G_{n+1}. We then iterate this construction on G_{n+1} to produce the desired sequence of random graphs.

Note that at present we have added only load-generator (L-G) branches, which seemed to have the most effect on power flow, but load-load and generator-generator branches should also be considered. Also due to the addition of only L-G branches, as the number of iterations increase, the resulting random graphs become increasingly more bipartite in structure. The bus and branch information was the same as used previously in creating the toy networks in Section 4.1.

As seen in Figure 4.10, the logarithmic plot of vertex degree distribution seems to be linear, hence the resulting graph is scale-free.

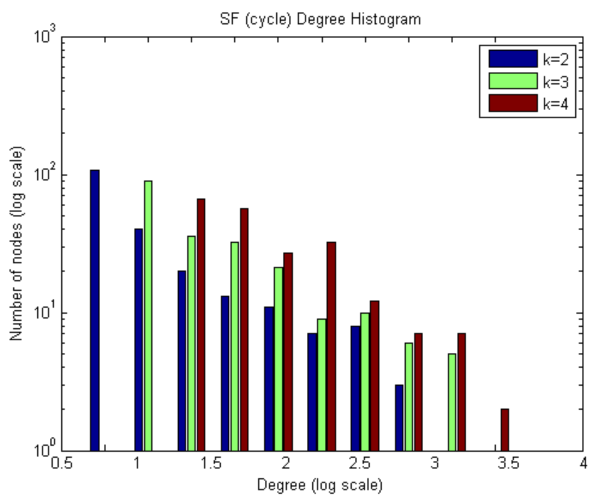

FIGURE 4.10. A logarithmic histogram of the distribution of vertex degree in a scale free graph created from a cycle graph with 6 vertices.

The average degree for a scale-free graph tends to $2k$ as the number of buses tend to infinity, where k is the number of branches added at each stage. Letting n_0 be the total number of buses at the initial stage, the number of buses at the nth stage will be $\#V(n) = n_0 + n$.

In order to count the number of branches, $\#E(n)$, it is easier to consider the sum of the degrees over all buses, which will give us $2\#E(n)$. We have

$$2\#E(n) = 2\#E(n-1) + 2k = 2e_0 + 2nk.$$

where e_0 is the original number of branches. The average degree is then

$$\frac{2\#E(n)}{\#V(n)} = \frac{2e_0 + 2nk}{n_0 + n} \to 2k \text{ as } n \to \infty.$$

Results. We investigate the minimal cost for the OPF problem on a scale-free graph for various values of k, and a large number of vertices (typically 200). More precisely, we set the initial graph G_0 to be one of either a small cycle, bipartite or complete graph. Then for fixed k we produce the graphs G_n for $1 \leq n \leq N$ (N fixed) using preferential attachment, and run the OPF problem on each G_n. Then we compute the minimal OPF cost per generator for each G_n with n odd (i.e. we compute [OPF cost (G_n)/ #(generators in G_n)] at each stage n where a generator is attached). Finally we plot the cost per generator against network statistics such as average degree, average path length and algebraic connectivity. For brevity we only present the results where G_0 is a cycle for values of $k = 3$ and 4. The trends shown in these results accompany the other graphs as well.

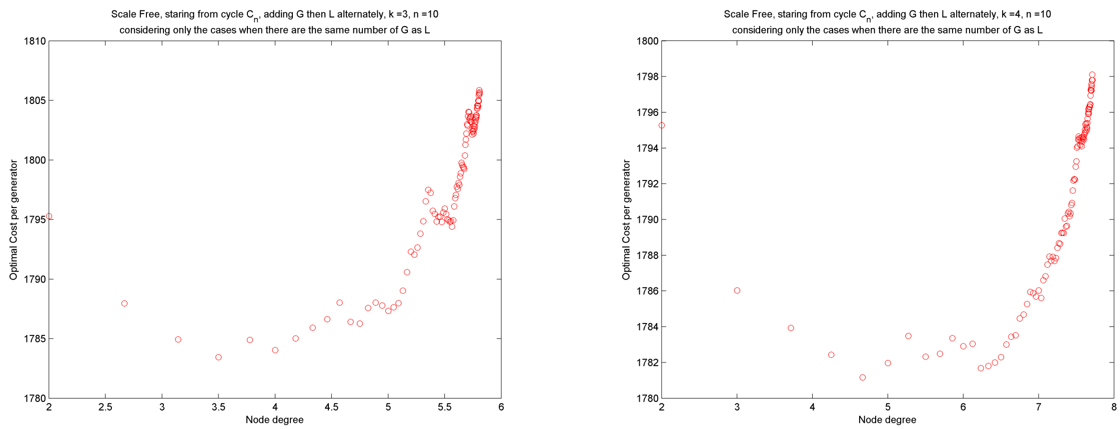

FIGURE 4.11. Cost per generator v.s. ave. degree for $G_0 = C_{10}$, $k = 3$ (left), $k = 4$ (right), and 200 iterations.

Notice from Figure 4.11 that as expected the average degree of these graphs cluster around $2k$. On the other hand, the cost per generator appears to cluster around a limiting value as well. The following is a heuristic explanation as to why the cost per generator should converge to a limiting value. Recalling that the cost is a polynomial $f(P^g) = \sum_{j=0}^{d} c_j (P^g)^j$ for all generators, the cost per generator as stage n can be expressed as

$$\text{Cost}(n) = \sum_{j=0}^{d} c_j \, \frac{1}{m_n} \sum_{l=1}^{m_n} (P_l^g)^j$$

where $m_n = 5 + \frac{n}{2}$ is the number of generators at stage n. The P_l^g are random variables, which are not necessarily independent, but are bounded, non-negative, and roughly identically distributed. Thus, by a law of large number type of argument, one would conclude that

$$\frac{1}{m_n} \sum_{l=1}^{m_n} (P_l^g)^j \to \mu_j \text{ as } n \to \infty \, .$$

Provided this is true, it follows that the cost per generator will have a limit as $n \to \infty$. We are confident that one can turn this heuristic argument into a rigorous proof through some careful analysis.

Notice also that the cost per generator has an increasing trend until it begins clustering. We believe this happens due to the 'cost' associated to adding branches to a network as was described in the Section 4.2. As we add more branches at each stage, the 'load' on the new branches causes some power loss and this in turn drives up the minimal cost of the corresponding OPF problem on the graphs.

From Figure 4.12 we notice that the cost per generator is generally an increasing function of the average path length for scale-free graphs. This is to be expected as a large distance between loads and generators (recall we only add L-G branches) will increase the cost of producing power for the network. We also see clustering of the cost per generator as is to be expected by the aforementioned heuristic.

Finally in Figure 4.13 we see that the cost per generator is generally an increasing function of algebraic connectivity on scale-free graphs. Although this may seem unintuitive, note that a large algebraic connectivity implies that the corresponding graph has very many branches. Since the

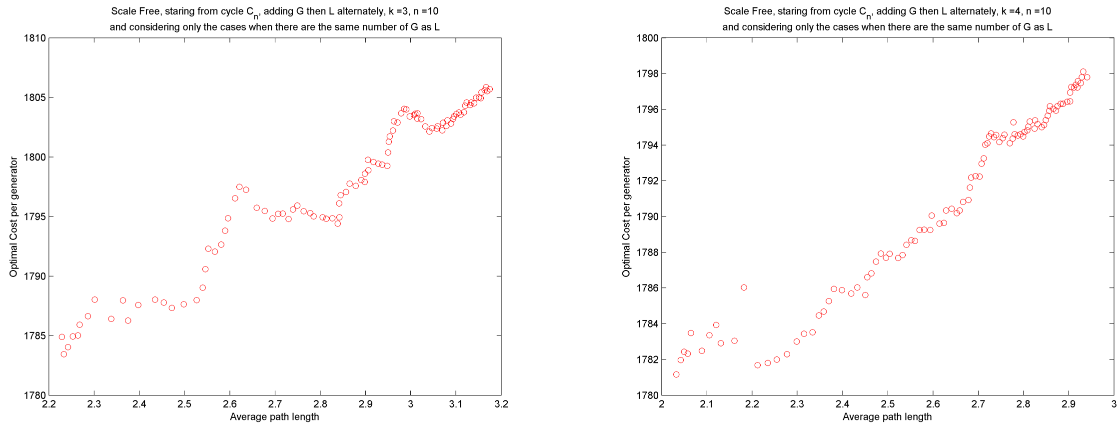

FIGURE 4.12. Cost per generator v.s. ave. path length for $G_0 = C_{10}$, $k = 3$ (left), $k = 4$ (right), and 200 iterations.

addition of branches can often increase the cost for the OPF problem as seen in Section 4.2, this trend is not surprising. We also observe the clustering phenomenon in the figures.

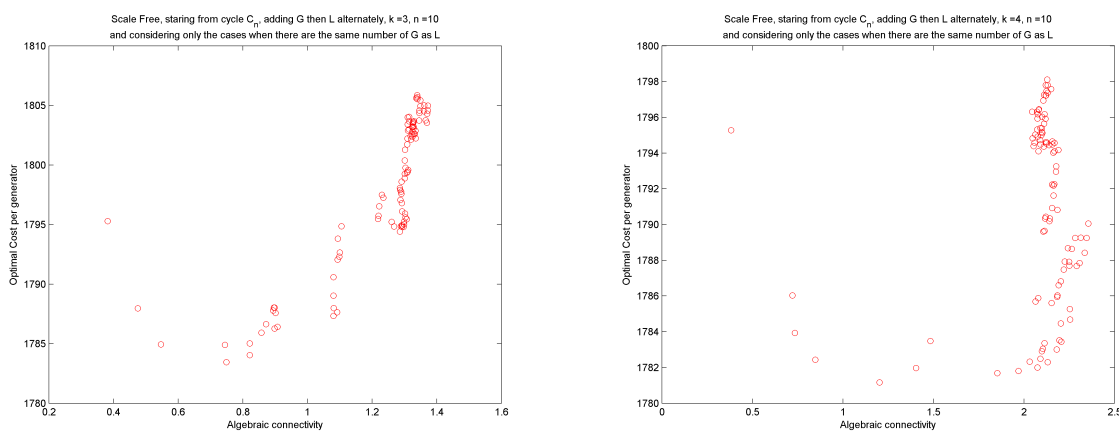

FIGURE 4.13. Cost per generator v.s. algebraic connectivity for $G_0 = C_{10}$, $k = 3$ (left), $k = 4$ (right), and 200 iterations.

4.4. **Comparison of costs of SCOPF for various graphs.** In this section we consider the robustness of the small world and scale-free networks under the single $e-1$ SCOPF problem. It turns out that similar to the complete graphs (and unlike the cycles), both the small world and scale-free graphs are surprisingly robust under contingency constraints. In particular, the ratio of the minimal cost for the OPF on these networks to the minimal cost for the single $e-1$ SCOPF is 1. However, the minimal cost for the power flow problems is significantly lower on the small world and scale-free networks compared to the cost for the corresponding complete graphs with the same number of buses.

In table 5 below, we show the cost of the OPF problem and the single $e-1$ SCOPF on the small world, scale-free, cycle, and complete networks with 100 buses each. The scale-free graph was constructed with $k = 3$ starting with the 6-cycle in Figure 4.1. The small world graph was constructed from a 100-cycle to have the same number of branches (288 branches) as the scale-free graph. The bus, branch, generator and cost data were the same for all networks. The results in table 5 are averaged over 10 trials.

Network	OPF cost	SCOPF cost
Small World	88186	88186
Scale-Free	88559	88559
C_{100}	88646	88820
K_{100}	98185	98185

Table 5. Cost for the OPF and SCOPF for various networks.

The small world and scale-free networks have significantly smaller number of branches than the complete graph based networks. The fact that these two network are as robust as the complete graph based networks for the SCOPF problem is startling. In particular, when designing real-world power networks, one is advised to design the underlying topology of the network based on the small world or scale-free topologies in order to reduce costs and maintain stability under contingencies.

References

[1] Reka Albert and Albert-Laszlo Barabasi, Emergence of Scaling in Random Networks, *Science*, vol. 286, pp. 509–512, 1999.

[2] J. Carpentier, Contribution to the economic dispatch problem, *Bulletin Society Francaise Electriciens*, vol. 3, no. 8, pp. 431–447, 1962.

[3] S. Martin, R.D. Carrm, and J.-L. Faulon, Random removal of edges from scale free graphs, *Physica A*, vol. 371, pp. 870–876, 2006.

[4] J. A. Momoh M. E. El-Hawary, and R. Adapa, A review of selected optimal power flow literature to 1993. Part I: Nonlinear and quadratic programming approaches, *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 96–104, 1999.

[5] J. A. Momoh M. E. El-Hawary, and R. Adapa, A review of selected optimal power flow literature to 1993. Part II: Newton, linear programming and interior point methods, *IEEE Transactions on Power Systems*,vol. 14, no. 1, pp. 105–111, 1999.

[6] K. S. Pandya and S. K. Joshi, A survey of optimal power flow methods, *Journal of Theoretical and Applied Information Technology*, vol. 4, no. 5, pp. 450–458, 2008.

[7] P. M. Pardalos and S. A. Vavasis, Quadratic programming with one negative eigenvalue is NP-hard, *Journal of Global Optimization*, vol. 1, no. 1, pp. 15-22, 1991.

[8] U.S.-Canada Power System Outage Task Force, Final Report on the August 14th Blackout in the United States and Canada, 2004. `https://reports.energy.gov/`

[9] Watts, D.J. and Strogatz, S.H., Collective dynamics of 'small-world' networks, *Nature*, vol. 393, pp. 440–442, 1998.

[10] R.D. Zimmerman, C.E. Murillo-Sanchez, and R.J. Thomas, MATPOWER: Steady-State Operations, Planning and Analysis Tools for Power Systems Research and Education, *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.

Department of Mathematics, Washington State University, Pullman, WA, U.S.A
E-mail address: balzalg@math.wsu.edu

Department of Mathematics, University of Toronto, ON, Canada
E-mail address: catalina.anghel@utoronto.ca

Department of Mathematics, University of California Los Angeles, Los Angeles, CA, U.S.A
E-mail address: wgan@math.ucla.edu

School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ, U.S.A
E-mail address: qhuang11@asu.edu

Department of Mathematics, University of Toronto, ON, Canada
E-mail address: mustazee.rahman@utoronto.ca

Department of Applied Mathematics, University of Waterloo, Waterloo, ON, Canada
E-mail address: a5shum@math.uwaterloo.ca