PNNL-29662

U.S. DEPARTMENT OF
ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Connectivity, Centrality, and Bottleneckedness: On Graph Theoretic Methods for Power Systems

SG Aksoy JD Taft

January 30, 2020

Pacific Northwest
NATIONAL LABORATORY

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RLO1830

PNNL-XXXXX

Connectivity, Centrality, and Bottleneckedness: On Graph Theoretic Methods for Power Systems

SG Aksoy JD Taft

January 30, 2020

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Executive Summary

This report provides an introduction to selected graph theoretic topics with pertinence to the structural analysis of electric power grid and communication systems. We focus on methodologies for defining, scoring, and identifying connectivity, spectral, and bottleneckeness properties in graphs, as well as vertex and edge importance measures such as centrality. We apply these measures to power systems and communications graph data, discuss and visualize the results, and comment on computational aspects of these methods. We show that graph theoretic methods can provide useful insights into grid and communication network structure, leading to tools and methods that could be used by electric utility engineers to improve key grid and communication network characteristics, such as resilience and scalability.

Contents

Figures

Tables

1.0 Introduction

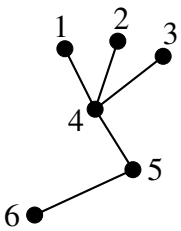
Grid Architecture is largely concerned with structures since we view the power grid as being comprised of a network of structures: electrical infrastructure, industry structure control structure, digital superstructure including communications networks, and convergent structures such as those for water, gas, and transport. These structures are interconnected in complex ways, and many of the characteristics of the grid that we wish to enhance or add derive directly from structure or are strongly influenced by it. Consequently it is important to have rigorous methods to analyze such structures and systematically modify them. These structures are far too complex to handled by inspection and hence the focus here on applying graph theory to grid structure problems.

Graphs are abstract structures that express pairwise relationships between entities. Because of their versatility and universality, graphs are a natural data structure for representing a myriad of complex systems. The burgeoning field of network science, for which graph theory serves as a mathematical scaffold, attests to the ubiquity and utility of graph theoretic analyses in far-ranging disciplines, including biology, chemistry, social science, and engineering [2]. In the case of power systems, the application of graph theoretic methods is far from new; see [12] for a survey of scientific literature on graph theoretic methods applied to the electric grid. Rather than provide a comprehensive survey here, we aim to provide a self-contained introduction to selected graph theoretic topics that may have increased pertinence in the structural analysis of power systems. In particular, we focus on methodologies for defining, scoring, and identifying connectivity, centrality, and bottleneckeness properties in graphs. We apply these measures to power systems graph data when possible, frequently present visualizations, and discuss the computational feasibility of these methods.

2.0 Preliminaries

A graph $G = (V, E)$ is a set V of elements called *vertices* and a set E of unordered pairs of vertices called *edges*. The terms "nodes" and "links" are sometimes used interchangeably with "vertices" and "edges", respectively. If the edge set E consists of *ordered* pairs of vertices, G is called a *directed graph* or *digraph*. For simplicity, we will be primarily concerned with undirected graphs, although topics we cover have analogous, albeit more complex, counterparts in the directed setting. Figure 2.1a illustrates an example of small graph with vertex set $V = \{1,2,3,4,5,6\}$ and edge set $E = \{\{1,4\},\{2,4\},\{3,4\},\{4,5\},\{5,6\}\}$.

For a given vertices $u, v \in V$, if $\{u, v\} \in E$, then we say u is *adjacent* to v. The set of vertices adjacent to u is called the *neighborhood* of u, and the number of such vertices is called the *degree* of u, which we denote d_u. The *degree sequence* of a graph, $(d_1, d_2, \ldots, d_n)$, is the list of degrees for each vertex in the graph. Degree sequences are often summarized more succinctly in the *degree distribution* of a graph, $(P(1), P(2), \ldots, P(\Delta))$. This is the sequence of the number of vertices for each possible degree, i.e., $P(i) =$ the number of vertices, v, with $d_v = i$, and Δ is the maximum degree in the graph. For example, the graph in Figure 2.1a has degree sequence $(1,1,1,4,2,1)$ and the degree distribution is $(4,1,0,1)$.



(a)

Graph Datasets		
	$\|V\|$	$\|E\|$
WECC: 115 kV	12997	15752
WECC: 138 kV	5691	6831
WECC: 230 kV	1066	1360
Poland: 110 kV	2856	3192
Poland: 220 kV	171	211
Poland: 400 kV	64	77
Texas: 115 kV	1275	1573
Texas: 345 kV	225	513

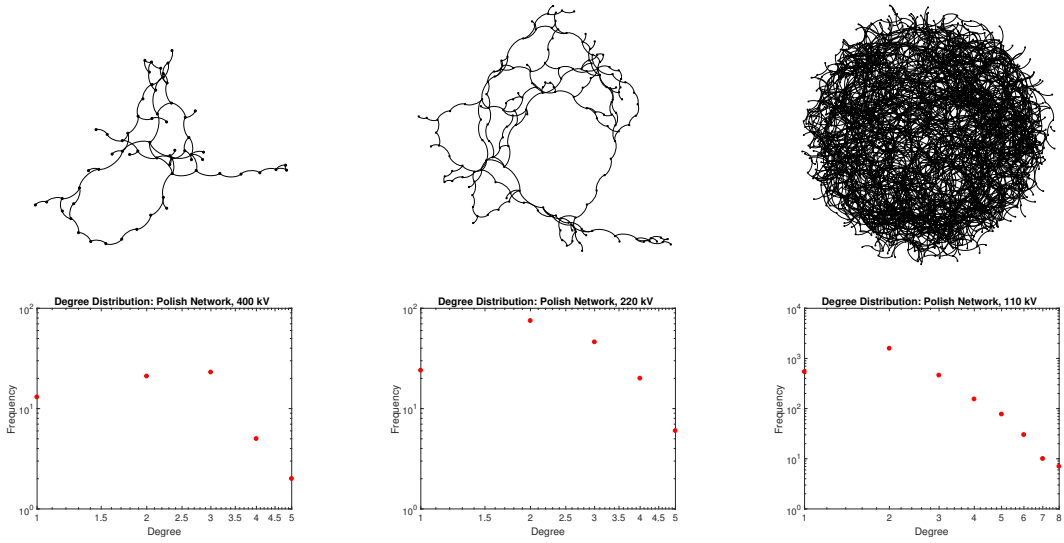
(b)

Figure 2.1. (a): A small graph on 6 vertices. (b) Vertex and edge counts for the WECC, Poland, and Texas 2000 networks.

2.1 Datasets

We consider three main datasets: (1) the WECC transmission network[(a)], (2) the Polish transmission network[(b)], and (3) and the Texas 2000 transmission network [4]. In the graphs derived from each of these networks, a vertex represents a generator, bus or substation, with given nominal voltage level. An edge represents a power line or, if linking two generators of different voltage levels, a transformer. We note the Texas 2000 network "is entirely synthetic, built from public information and a statistical analysis of real power systems"[(c)] and also features with synthetic geographic locations for each node, whereas the WECC and Polish data do not contain geographic data. Table 2.1b lists the datasets and presents the node and edge count for each graph.

Figure 2.2 (top row) presents visualizations of the three graphs derived from a Polish transmission network. Lacking any geographic data, the node position layout is determined by the



(a) **(b)** **(c)**

Figure 2.2. Visualizations (top row) and degree distribution plots (bottom row) of Polish 400 kV, 220 kV, and 110 kV networks.

(a) The U.S. WECC power grid data was obtained through the U.S. Critical Energy Infrastructure Information (CEII) request process

(b) The Poland network is open-source data from MATPOWER `http://github.com/MATPOWER/matpower`

(c) `https://egriddata.org/dataset/texas-2000-bus-system-activsg2k`

Yifan Hu algorithm [9], a popular graph visualization tool frequently applied to large networks. Nonetheless, the size and density of the network in Figure 2.2 yields a cluttered visualization. As we will soon see, graph theoretic methods provide a critical capability in analyzing large and complex networks for which direct inspection is either infeasible or not informative. Figure 2.2 (bottom row) plots the degree distributions of the three Polish networks. Due to the heavy-tailed nature of these degree distributions, we present the plots in log-log scale.

A variety of network statistics can be discerned from the degree distribution. For example, the rightmost point in the degree distribution plot for the Polish 110 kV graph has x-value 8, meaning the highest number of power lines leaving a single generator in this is network is 8, and y-value 7, meaning there are precisely 7 such generators. Furthermore, the degree distribution can also tell us there are 3192 power lines in this network, 2.2 power lines leave a generator on average, only a small fraction of generators, about 9%, have more than 3 power lines leaving, and so on.

3.0 Connectedness and Connectivity

A fundamental concept in graph theory is that of *connectedness* and *connectivity*. Connectedness is based on the notion of a graph walk. A *walk of length k* on a graph is a sequence of $k+1$ successively adjacent vertices, $u_0, u_1, u_2, \ldots, u_k$, such that $u_i \neq u_{i+1}$ for all i, but vertices may otherwise be repeated. A set of vertices $S \subseteq V$ is *connected* if there exists a walk between every pair of vertices and a connected set S is called a *connected component* of G if S is not contained within a strictly larger, connected set of vertices. Every graph may be uniquely partitioned into its connected components which, loosely speaking, constitute the aggregate blocks of a network. For this reason, graph analyses are often conducted on a per-component basis, either by choice or technical necessity.

Graph connectedness is a binary concept: G is connected if V is a connected component and is disconnected otherwise. Although coarse, connectedness is fundamental the context of power systems represented as a graphs: power cannot flow between two nodes (representing generators or substation) that lie in two different connected components. However, connectedness does not address "how connected" a component is, which we refer to *connectivity*. Connectivity notions may be local in that they are associated with particular vertices or edges in the network, or global in reflecting the entire network. Connectivity metrics have utility in practice in providing a rigorous means by which to quantify or score structural properties, as well as locate portions of interest within the graph that may otherwise be difficult to identify.

In this rest of this section, we will review connectivity concepts, discuss the structural properties they reflect, and apply them to graphs associated with power systems. Proceeding in order of increasing conceptional nuance, we first explore cut edges, or single edges whose removal disconnects the graph. Broadening to the case of multiple edges whose simultaneous removal disconnect a graph, we then turn our attention to so-called *isoperimetric* measures, and in particular, a bottleneck metric called the Cheeger constant. Given the inherent difficulty of exact Cheeger constant computation in practice, we then discuss how Laplacian matrix eigenvalues and eigenvectors may be used to approximate this numerical measure, as well as locate sparse cuts in the graph.

3.1 Cut edges and k-edge connectivity

A connected graph is k-edge connected if it remains connected whenever any choice of k edges are removed. The largest k for which a graph is k-edge connected is called its *edge-connectivity*.

Hence, if a graph has edge-connectivity 3, this means disconnecting the graph by deleting any 3 edges is impossible, but the graph can be disconnected by deleting some choice of 4 edges. Although a seemingly natural way to quantify connectivity in graphs, edge-connectivity is too restrictive to meaningfully quantify connectivity for most real-world networks because of the presence of degree 1 vertices. That is, because real world networks tend to exhibit heavy-tailed degree distributions, they feature *many* degree 1 vertices, as we observed in Figure 2.2. The presence of a *single* degree 1 vertex ensures the graphs edge-connectivity is 1, as deleting the edge incident to that vertex disconnects the graph. An edge whose deletion disconnects the graph is referred to as a *cut edge*. In the graph in Figure 3.1, edges a and b are cut edges.

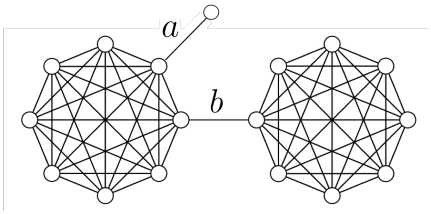


Figure 3.1. A graph with two cut edges, a and b.

Observe that, while both a and b are cut edges, they removal from the graph yields starkly different outcomes: when a is removed, the graph is separated into a large connected component with all but one vertex, and a small connected component with a single vertex. In contrast, when b is deleted, the graph is separated into roughly two pieces of equal size. Hence while the prevalence of degree 1 vertices in graphs associated with power systems make cut edges plentiful, this observation suggests a straightforward manner in which one can distinguish between cut edges. We make precise this distinction by defining the "weight" of a cut edge, as follows: for a connected graph G, define the weight of a cut edge e, denoted $w(e)$ as

$$w(e) = \frac{\min\{|C_1|, |C_2|\}}{|G|},$$

where C_1 and C_2 denote the two connected components of the graph $G \setminus e$.
Simply put, $w(e)$ is the fraction of the graph that is removed when edge e is deleted. In Figure 3.1, $w(a) = \frac{1}{17}$ or about 6% while $w(b) = \frac{8}{17}$ or about 47%. Turning our attention to the data, we compute the cut edge weight over all cut edges. Table 3.1 presents the 5 largest-weight cut edges for WECC networks.

WECC 115 kV Largest Comp.		**WECC 138 kV** Largest Comp.		**WECC 230 kV** Largest Comp.		**Entire WECC** Largest Comp.	
Edge e	$w(e)$	Edge e	$w(e)$	Edge e	$w(e)$	Edge e	$w(e)$
{42310, 42414}	38.85%	{65920, 66105}	44.44%	{60165, 60232}	5.00%	{50464, 51818}	0.74%
{42414, 42457}	38.80%	{60020, 65920}	44.23%	{60062, 60165}	4.84%	{51818, 51820}	0.74%
{42435, 42457}	38.75%	{60020, 61646}	39.62%	{60073, 69520}	2.50%	{50463, 51820}	0.73%
{42402, 42435}	38.64%	{61646, 61654}	39.41%	{66565, 69520}	2.34%	{50672, 50929}	0.41%
{42402, 45608}	38.54%	{61491, 61654}	38.99%	{66050, 66565}	2.17%	{50929, 80931}	0.40%

Table 3.1. The 5 largest weight cut edges for WECC networks. Edges are identified by the bus numbers of their endpoints, and cut edge weight $w(e)$ is reported for each edge.

We observe that, for the WECC 115 kV and 138 kV networks, it is possible to disconnect about 40% of the network's vertices from the main component by deleting a single edge, while for the aggregated WECC network (obtained by taking the union of graphs over all voltage levels and including transformer edges) less than a percent of the network can be removed by deleting a single edge. To provide a more comprehensive overview of cut edge weight, we consider random cut edge deletion. In Figure 3.2, we plot the probability that at least k% of the graphs

vertices are removed if a single edge is deleted uniformly at random. For example, if $(x,y) = (2,10)$, then this means that if an edge is deleted at random, there is a 10% chance that at least 2% of the graph will be separated from the largest connected component.

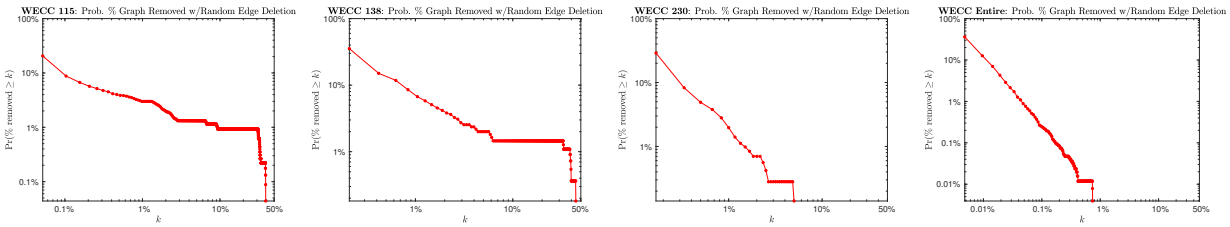


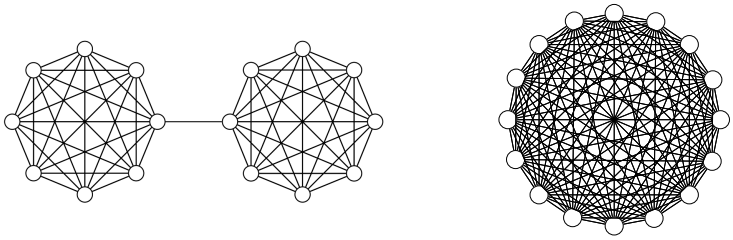
Figure 3.2. Probability at least k% of graph removed under random single edge deletion scenario.

The plots presented Figure 3.2 summarize cut edge weight behavior in these networks. For clarity, the plots are presented in log-log scale. The rightmost point in each plot represents the maximum fraction of the graph that may be disconnected via deletion of an edge, which we listed in Table 3.1. Similarly, the leftmost point in each plot is the percentage of edges in the graph which are cut edges. A steeper slope indicates the probability a larger proportion of the graph will be disconnected decreases more quickly. For instance, consider the rightmost plot corresponding to the aggregated WECC network (consisting of vertices representing buses of all voltage levels) There, the probability that 0.01% of the graph is disconnected because of random edge failure is roughly 10%, whereas the probability of disconnecting 0.1% of the graph (an order of magnitude more of the graph) decreases sharply to about 0.1% (two orders of magnitude less in probability).

We also note some of the plots exhibit pronounced horizontal segments, within which the probability of disconnecting at least k% of the graph remains fixed. For instance, observe the horizontal segment between about $x = 6\%$ and $x = 30\%$ for the WECC 138 kV graph. This means that while there are some cut edges whose removal disconnects about 6% of the graph, any cut edges disconnecting strictly more than this disconnect at least 30% of the graph. In this way, horizontal segments correspond to "jumps" in the percentage of the graph that could be disconnected by a single edge failure. Taken to the extreme, a particularly long horizontal segment indicates a random edge failure could either disconnect a "small" or "large" proportion of the graph, but not any size in between these endpoints.

The foregoing has a number of useful applications for Grid Architecture. For example: the slope of the line in the random edge deletion graphs (before the horizontal segment plateau or shelf) provides a means to quantify the resilience of a network such as a power grid or a communication system to random failures of the type that occur as part of normal operations in any Ultra-Large-Scale complex system. Such quantification is elusive under common approaches to grid analysis but is increasingly important due to the focus on grid resilience.

We remind the reader this analysis is centered on *single* edge deletions. Even in cases where it is impossible to disconnect much of a network through single edge deletions, it may still be the case that multiple edge deletions (the number of which may be relatively few compared to the size of the network) may result in disconnecting large portions of the network. In order to extend the above analysis to this case, we consider isoperimetric numbers of graphs in the next section, which are one way of quantifying such "bottlenecks".



(a) (b)

Figure 3.3. (a) A barbell graph on 16 vertices. (b) A complete graph on 16 vertices.

3.2 Isoperimetric numbers and bottlenecks

The classical isoperimetric problem in geometry is to find, among all curves of a given length, the maximum area-enclosing curve. Isoperimetric problems in graphs can be framed anal-ogously by measuring the "boundary" of a subset of vertices, taken to be the edges leaving that se, relative to some notion of the "size" of that set. For example, consider the so-called *barbell graph* depicted Figure 3.3a, taking either side of the barbell as our subset vertices yields a "large" set with a single edge leaving. This subset of vertices is called a *sparse cut*; we say this graph exhibits "bottleneckedness". In contrast, considering the complete graph depicted in Figure 3.3b, taking any subset of vertices yields "many" edges that leave that set; no such bottleneck exists here.

Formalizing this intuition, the *isoperimetric number*, also sometimes referred to as the *Cheeger constant* or *conductance* of a graph G is a numerical measure of graph bottleneckedness. This notion is based on the *Cheeger ratio of a vertex subset S*, denoted $h(S)$, given by

$$h(S) = \frac{e(S,\bar{S})}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}},$$

where $e(S,\bar{S})$ denotes the number of edges with one endpoint in S and the other not in S (i.e. the boundary size of S) and $\text{vol}(S)$ is the sum of the degrees of the vertices within that set (i.e. the size of S). The *Cheeger constant of a graph G* is the minimum Cheeger ratio of a set over all possible vertex subsets, i.e.

$$\Phi(G) = \min_{S \subseteq V} h(S).$$

A small Cheeger constant indicates the existence of a set with small Cheeger ratio; smaller values indicate tighter bottlenecks. In this sense, the Cheeger constant is an "extreme-case" measure reflecting the tightest bottleneck. Such analysis has application to determining potential congestion problems as well as cyber security vulnerability for grids and grid communications.
Example: Consider the graph in Figure 2.1a, and the vertex subset $S = \{1,2,4\}$. There are two edges with one endpoint in S and the other in the complement $\bar{S} = \{3,5,6\}$; namely, the edges $\{4,3\}$ and $\{4,5\}$. Hence $e(S,\bar{S}) = 2$. Furthermore, vertices 1, 2, and 4 have degree 1, 1, and 4, yielding $\text{vol}(S) = 6$, while vertices 3, 5, and 6 have degree 1, 2, and 1, yielding $\text{vol}(\bar{S}) = 4$. Putting this all together, we get that the Cheeger ratio of S is

$$h(S) = \frac{e(S,\bar{S})}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}} = \frac{2}{\min\{6,4\}} = \frac{1}{2}.$$

Observe that if a graph is disconnected, then $\Phi(G)$ is trivially zero; accordingly, one computes the Cheeger constant on a connected component of the graph. Furthermore, by definition,

the Cheeger ratio it is always between 0 and 1, regardless of the graph's size. This property facilitates easy cross-comparisons across multiple graphs of different sizes.

While computing the Cheeger ratio of a particular subset is easy, computing the Cheeger constant of a graph is much more difficult. A brute force way to try to compute the Cheeger constant of a graph is to compute the Cheeger ratio of each possible subset, and then take the minimum. However, there are *many* vertex subsets of a graph. A graph with 10 vertices has $2^{10} = 1024$ vertex subsets, a graph with only 30 vertices has over a billion vertex subsets! In general, computing the Cheeger constant of a graph exactly is NP-hard [11] and is thus rarely done in practice, except for very special classes of graphs or very small graphs. However, as we explain below, tools from a mathematical discipline called *spectral graph theory* allow us to find approximate solutions efficiently.

3.3 Graph eigenvalues and finding bottlenecks

Spectral graph theory concerns how graph properties are encoded in linear-algebraic quantities called eigenvalues, which are derived from matrices associated with a graph. For a more comprehensive survey of such results, refer to [3, 5, 8, 7]. However, for readers unfamiliar with linear algebra and spectral graph theory, we provide a brief description of the necessary fundamentals. Given a matrix A, the vector x is called an *eigenvector* and the number λ an *eigenvalue* if

$$Ax = \lambda x.$$

That is, an eigenvector is a vector for which multiplication by the matrix does not rotate the vector but only scales it by a constant factor (i.e. the eigenvalue). There are a variety of different matrices one could associate with a graph. Perhaps the most well-known is the adjacency matrix A, which is defined element-wise by

$$A(i,j) = \begin{cases} 1 & \text{if } \{i,j\} \in E \\ 0 & \text{otherwise} \end{cases}.$$

Two other important matrices, which we will use to approximate the Cheeger constant, are the Laplacian matrix L and normalized Laplacian matrix $\mathscr{L}$,

$$\begin{aligned} L &= D - A, \\ \mathscr{L} &= D^{-1/2} L D^{-1/2}, \end{aligned}$$

where A is the (unweighted) adjacency matrix defined above and D is the diagonal vertex degree matrix ($D(i,i) = d_i$ and $D(i,j) = 0$ whenever $i \neq j$). The eigenvalues of each of A, L and $\mathscr{L}$ are labeled in increasing order and denoted by λ_i,

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

The set of eigenvalues for a matrix is called the *spectrum*. The matrix underlying a particular eigenvalue may be specified by writing, for example, $\lambda_i(L)$ to denote that λ_i is the ith eigenvalue of the Laplacian matrix L; however, we drop the subscript when clear from context. For an illustration of a graph, its normalized Laplacian, and normalized Laplacian spectrum see Figure 3.4.

Having established the necessary background in spectral graph theory to proceed, we return to our investigation of bottleneckedness in graphs. Recalling that computing the Cheeger constant

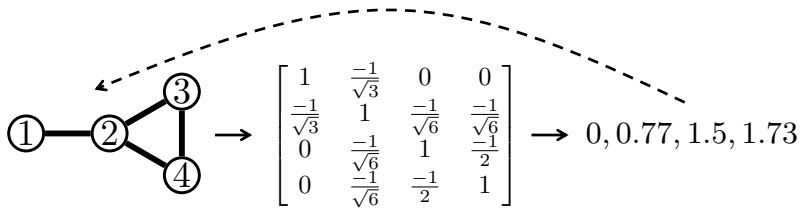


Figure 3.4. From left to right: a graph, its normalized Laplacian matrix, and normalized Laplacian eigenvalues. Spectral graph theory establishes relationships between eigenvalues or eigenvectors, and the graph they are derived from, represented in the diagram by the dashed line.

is infeasible in practice, *Cheeger's inequality* [8] relates the second eigenvalue of the normalized Laplacian, λ_1, with the Cheeger constant $\Phi(G)$, via upper and lower bounds:

$$2 \cdot \Phi(G) \geq \lambda_1 \geq \frac{\Phi(G)^2}{2}.$$

Put equivalently, this means

$$\sqrt{2\lambda_1} \geq \Phi(G) \geq \frac{\lambda_1}{2}.$$

In this sense λ_1 approximates $\Phi(G)$. As an immediate consequence of Cheeger's inequality, we can interpret λ_1 of a graph in the following way: assuming that a vertex subset S isn't too "large" and satisfies $\text{vol}(S) \leq \frac{1}{2}\text{vol}(G)$, then

- for any such subset S, there are at least $\lambda_1 \cdot \text{vol}(S)/2$ edges leaving S.
- there exists such a subset S for which there are at most $\sqrt{2\lambda_1} \cdot \text{vol}(S)$ edges leaving S.

Turning our attention back to our data, we compute the λ_1 on the largest connected component of the WECC networks. Additionally, we compare this value with that of random graphs generated by the Chung-Lu (CL) random graph model. We refer interested readers to [6] for further details on the CL model, and to [1] for work applying the CL model to synthetically generate power systems. For our purposes here, it is sufficient to note the CL model, when fitted to given graph data, outputs a graph in which vertices have the same degree as in the original graph, in expectation. Thus, up to some noise, the original and random CL networks have the same degree distributions. Consequently, to the extent that λ_1 differs in the original and random networks, this provides experimental evidence that bottleneckedness reflected by λ_1 can't be explained as a sole consequence of degree distribution. We present the results in Table 3.2.

The eigenvalue λ_1 in WECC		
	Original	Random CL
WECC: 115 kV	3.21e–5	2.21e–2
WECC: 138 kV	2.36e–4	2.60e–2
WECC: 230 kV	1.55e–3	3.12e–2
Entire WECC	4.56e–5	1.98e–2

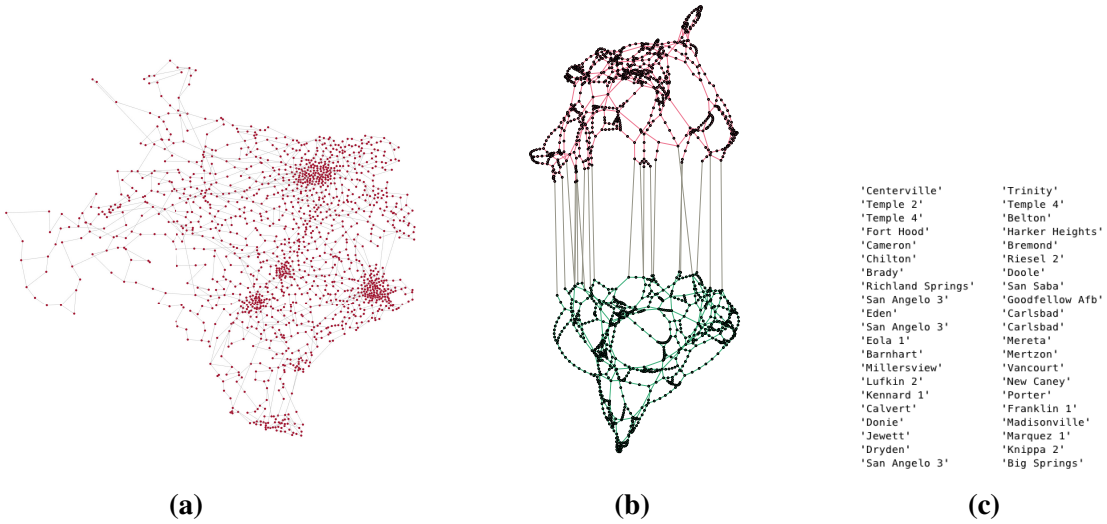
Table 3.2. The eigenvalue λ_1 in the original WECC data vs random Chung-Lu WECC networks.

We observe that the original networks have smaller values of λ_1, by between 1 to 3 orders of

WECC 230kV
Eigenvector Sweep Statistics
Cheeger Ratio: 1.12e–2
vol(S)/vol(G): 37%
Edges Crossing Cut
1. {46645, 40011}
2. {46156, 40021}
3. {45075, 40092}
4. {46486, 40671}
5. {46609, 40673}
6. {46629, 42505}

(a) **(b)** **(c)**

Figure 3.5. (a) the largest connected component of the WECC 230kV network, (b) the minimum weight cut found by the eigenvector sweep method, (c) a list of the 6 edges crossing this cut, as identified by the bus numbers of their endpoints.



'Centerville'	'Trinity'
'Temple 2'	'Temple 4'
'Temple 4'	'Belton'
'Fort Hood'	'Harker Heights'
'Cameron'	'Bremond'
'Chilton'	'Riesel 2'
'Brady'	'Doole'
'Richland Springs'	'San Saba'
'San Angelo 3'	'Goodfellow Afb'
'Eden'	'Carlsbad'
'San Angelo 3'	'Carlsbad'
'Eola 1'	'Mereta'
'Barnhart'	'Mertzon'
'Millersview'	'Vancourt'
'Lufkin 2'	'New Caney'
'Kennard 1'	'Porter'
'Calvert'	'Franklin 1'
'Donie'	'Madisonville'
'Jewett'	'Marquez 1'
'Dryden'	'Knippa 2'
'San Angelo 3'	'Big Springs'

(a) **(b)** **(c)**

Figure 3.6. (a) the Texas 2000 network, (b) the minimum weight cut found by the eigenvector sweep method, (c) a list of the 21 edges crossing the cut.

magnitude, than comparable random graphs. This suggests that these networks have tighter bottlenecks than one would expect from random networks with comparable vertex degrees. These values of λ_1 also provide bounds on the Cheeger constant via the Cheeger inequality: for example, in the WECC 115 kV network, the value $\lambda_1 = 3.21\text{e–}5$ tells us the true Cheeger constant is in the interval [1.60e–5, 8.01e–3].

While λ_1 provides a computationally efficient manner in which we can approximate bottleneckeness via the Cheeger constant, it may also be desirable to explicitly identify bottlenecks in graphs. Our work in the previous section focused on finding cut edges that come as close as possible to dividing the graph in half, but this analysis does not fully capture bottleneckedness because it only considers the effect of disconnections resulting from *single* edge deletions. In some cases, it may be impossible to disconnect a large portion of the graph by deleting a single edge but possible by deleting several edges. However, due to combinatorial explosion, checking all combinations of 2, 3, or more edges, quickly becomes infeasible. Luckily, spectral graph theory again provides tools we can leverage to find an approximation of the tightest bottleneck. For example, the so-called "eigenvector sweep" method utilizes the eigenvector associated with λ_1 in order to identify a set of vertices S with small Cheeger ratio. This method derives from the proof of the Cheeger inequality and we omit the details; interested readers may refer to [8].

We apply the method to two of our datasets:

- **WECC 230 kV**: Figure 3.5a presents a visualization of the largest connected component of the WECC 230 kV graph. We note the positions of the vertices in the visualization do not reflect the physical geography of buses; they were determined by the Yifan Hu layout. Our previous analysis in Table 3.1 showed that single edge deletions could only remove about 5% of the graph. However, using the eigenvector sweep method, we find a cut with 6 edges divides the graph roughly in half with 230 vertices on one side and 369 vertices on the other side. The set has Cheeger ratio 1.12e–2. Figure 3.5b presents a visualization of this cut in which the vertices on each side of the cut are colored differently. The 6 edges crossing this cut are listed in Figure 3.5c.

- **Texas 2000**: Figure 3.6a presents of a visualization of the entire Texas 2000 [4] network. Using the eigenvector sweep method, we find a cut with 21 edges crossing that divides this network roughly in half. We note this network was synthetically generated and comes equipped with artificial geographic locations for vertices. This geography is reflected in Figure 3.6a whereas Figure 3.6b uses a different layout to emphasize the cut and make the edges crossing the cut easily discernible. In Figure 3.6c, we list the 21 edges crossing this cut.

4.0 Centrality and Path Counting

In the previous sections, we've outlined how to characterize *connectivity* in graphs, as well as score and identify bottlenecks in the graph. While this affords us tools for quantifying connectivity for the graph as a whole, one may also want to investigate connectivity on a more granular level – down to either for a single vertex, edge, or a pair of vertices. For instance, one may ask whether there are many ways to reach one vertex from another, which vertices are "centrally located" and which are on the "periphery" of the network, and whether, in the event of vertex or edge "failures", which pairs of vertices have high likelihood of remaining connected via some path. Centrality metrics and path counting represent ways of formalizing these questions. Such issues arise in the consideration of grid and communication resilience and cybersecurity.

4.1 Path Types and Enumeration

There are a plethora of different types of paths and walks in a graph. The most basic notion is that of a *walk*. A walk of length k is a sequence of $k+1$ successively adjacent vertices, i.e., a sequence $u_0, u_1, u_2, \ldots, u_k$, such that $\{u_i, u_{i+1}\}$ is an edge, and $u_i \neq u_{i+1}$ for all i, but vertices may otherwise be repeated. A *path* is a special type of walk in which all vertices and edges traversed are distinct. The length of the shortest path between two vertices u, v, generally denoted $d(u,v)$, is called the *distance* between u and v.

A number of centrality metrics are based on graph distance, as well as path enumeration, for different types of paths. We note that since vertices in a walk may be repeated, there are an infinite number of walks between two vertices in a connected graph. Below, we list and comment on simple variations for counting paths between two vertices, u and v.

1. The number of paths between u and v.
    - This is generally not computationally tractable, even for relatively small examples. Graphs arising from real data tend to have many long paths, often too numerous to count. This motivates putting a threshold limit on path length.

2. The number of shortest paths between u and v.

3. The number of paths between u and v of length $d(u,v)+k$ for a non-negative integer k.
   - If we take $k=0$, then this is just the number of shortest paths.
   - Choosing some small value of k means you're allowing for slightly longer paths which might include a few extra hops than the shortest possible.
   - One can generalize this further by choosing k to depend on other properties of u and v.

4. The number of paths between u and v of length $\leq k$, for some fixed constant k.
   - This makes the allowable path length the same for all pairs of vertices and doesn't make it depend on the vertices in question.

5. The number of edge-disjoint or vertex-disjoint paths between u and v.
   - Two paths are edge (resp. vertex) disjoint if none of the edges (resp. vertices) traversed in one path appear in the other path.
   - Note that every vertex-disjoint path is an edge-disjoint path, but not vice-versa. Hence, vertex-disjoint paths are rarer in the sense there are at least as many (if not more) edge-disjoint paths between u and v as there are vertex-disjoint paths.
   - Menger's theorem [10] relates the number of edge and vertex-disjoint paths to the size of the minimum edge and vertex cut between u and v. That is:
     - The fewest number of edges you'd need to delete to disconnect u from v is the same as the number of edge-disjoint paths between u and v.
     - The fewest number of vertices you'd need to delete to disconnect u and v is the same as the number of vertex-disjoint paths between u and v.

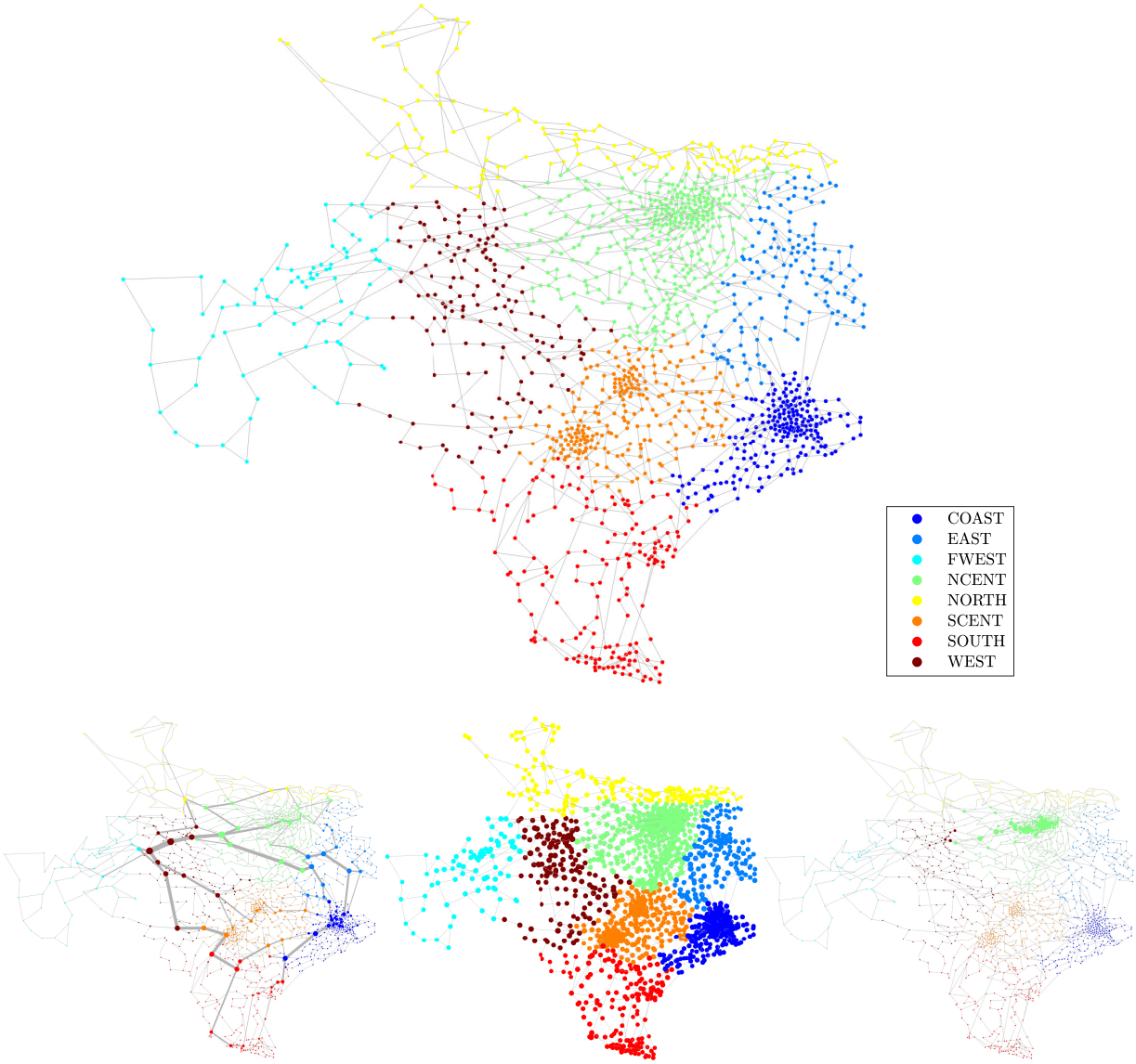
4.2 Centrality

We consider three different types of graph vertex-centrality measures. Each of these measures how "central" a vertex is within the graph, but based on different criteria. Below, we first give an intuitive description of the sense in which each of these measures centrality, and then follow this with a formal definition.

- **Betweeness Centrality**: *how frequently is this vertex on a shortest path between pairs of other vertices?* For $x,y \in V$, let $S(x,y)$ be the number of shortest paths between x and y, and for $u \neq x,y$, let $S_u(x,y)$ denote the number of those shortest paths which contain u. The betweenness centrality of $u \in V$ is

$$\sum_{\substack{x,y \in V \\ u \neq x,y}} \frac{S_u(x,y)}{S(x,y)}.$$

- **Eigenvector Centrality**: *if we consider at all possible walks (not just shortest paths), how frequently do we encounter this vertex?* Let $G=(V,E)$ be a connected graph, A its adjacency matrix, and λ the largest eigenvalue of A. The eigenvector centrality of $u \in V$ is



	Betweenness		Closeness		Eigenvector	
Rank	*Node*	*Area*	*Node*	*Area*	*Node*	*Area*
1	Sterling City 3	WEST	Eastland	NCENT	Fort Worth 10	NCENT
2	Eastland	NCENT	Abilene 2	WEST	Fort Worth 16	NCENT
3	Blackwell 2	WEST	De Leon	NCENT	Dallas 8	NCENT
4	Abilene 2	WEST	Clyde	NCENT	Euless	NCENT
5	De Leon	NCENT	Blackwell 2	WEST	Gordon	NCENT

Figure 4.1. Visualization of the Texas 2000 network (top) and underneath, the same network with vertex sizes proportional to their betweenness (left), closeness (middle), and eigenvector (right) scores. The table (bottom) lists the top 5 rated vertices under each notion of centrality.

the u^{th} entry of the eigenvector x associated with λ, i.e.

$$Ax = \lambda x.$$

- **Closeness Centrality**: *on average, how close is this vertex (in terms of shortest path length) to other vertices?* For a graph $G = (V,E)$, let $d(x,y)$ denote the distance between

$x, y \in V$. The closeness centrality of $u \in V$ is

$$\frac{|V|-1}{\sum_{x \in V} d(u,x)}.$$

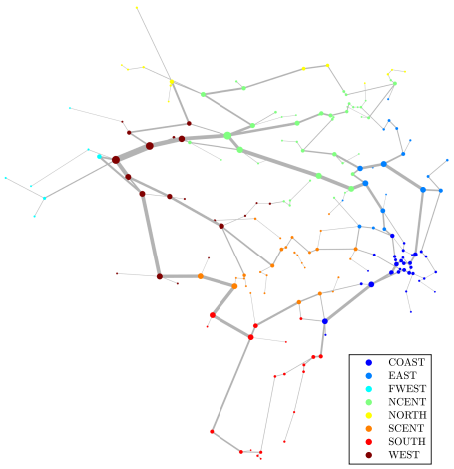
Each of the above vertex-based measures could translate to edge-based measures in a simple way: for instance, if the centrality of vertex u is X and that of vertex v is Y, then one could define the edge-centrality of the edge $\{u,v\}$ as the product of these values, $X \cdot Y$. Nonetheless, we consider one additional notion of edge-centrality defined natively for edges:

- **Edge-betweeness Centrality**: *how frequently is this edge on a shortest path between pairs of other vertices?*

The formal definition of edge-betweenness centrality is similar to that of vertex-centrality and is omitted. We note that the above metrics have analogs and closely related variants suitable for disconnected and directed graphs. For instance, in the case of eigenvector centrality, Katz centrality and PageRank are two closely related variants; in the case of closeness centrality, the so-called "harmonic centrality" is a suitable alternative for disconnected graphs.

We now apply these metrics to the Texas 2000 dataset. Figure 4.1 (top) visualizes this network. Note the vertices are colored according to their grouping into one of 8 geographic regions. We present the centrality scores by drawing vertices proportionally to their centrality score, according to each of the three centrality metrics. Figure 4.1 (bottom) presents these visualizations. In the case of edge-betweenness centrality, we visualize these results in the same visualization as for vertex betweenness centrality, drawing edges proportionally to their edge-betweenness score. Additionally, we list the top 5 ranked vertices according to each centrality measure and their corresponding location.

From the visualization, a number of qualitative differences between these centrality metrics are immediately apparent. Although all three of these metrics reflect "centrality", this illustrates how nuanced choices in defining "centrality" can result in dramatic differences when applied to data. First, the stark differences in vertex sizes for the eigenvector centrality score visualization (right) suggest the distribution of score values are heavily skewed. Here, vertices in the NSCENT region receive the highest scores while those in other regions have relatively minus-



(a)

	Edge Betweeness	
Rank	*Edge*	*Areas*
1	{Sterling City 3, Blackwell 2}	{WEST, WEST}
2	{Abilene 2, Blackwell 2}	{WEST, WEST}
3	{Carlsbad, Sterling City 3}	{WEST, WEST}
4	{De Leon, Waco 2}	{NCENT, NCENT}
5	{Eastland, Abilene 2}	{NCENT, WEST}

(b)

Figure 4.2. (a) Subgraph of the Texas 2000 network induced by edges with betweenness centrality score in at least the 90th percentile, (b) top 5 ranked edges according to edge-betweenness

cule scores. The closeness centrality scores (middle) appear less discriminative, and exhibit a smoother distribution reflected by the roughly equally sized vertices in the figure. The betweenness centrality scores are the most diverse across geographic regions; vertices with betweenness score in the 90th percentile appear in nearly every region. We also observe, unsurprisingly, that vertices with high betweenness tend to be incident to edges with high betweenness. In Figure 4.2, we restrict this network to the subgraph induced by edges with betweenness score in at least the 90th percentile and list the top 5 rated edges. This emphasizes the critical edges that form the "betweeness centrality skeleton" of the network.

5.0 Tier Bypassing: Formulation & Correction

As a final example of how graph theoretic formulations aid our understanding of power systems, we formalize the problem of "tier bypassing" and use this to illustrate a simple correction algorithm. The tier bypassing problem and its solution are central to the Transmission/Distribution coordination problem and is closely related to layered decomposition-based grid coordination and DSO models [13]. Our system is represented by a triple (V,E,f), where $G=(V,E)$ is a graph and $f:V\to\{1,\ldots,T\}$ is a surjective function assigning a natural number to each vertex denoting its tier. Tier bypassing occurs in (V,E,f) if there exists an edge $\{u,v\}\in E$ such that $|f(u)-f(v)|\geq 2$. We seek an output triple (V',E',f') that satisfies the following:

1. (V',E',f') does not tier bypass.

2. Edges link different tiers, i.e. E' does not contain any edges $\{u,v\}$ such that $f(u)=f(v)$.

3. There is exactly one vertex of Tier 1.

4. Every vertex (except for the Tier 1 vertex) must be connected to *exactly* one vertex of lesser tier.

If a given system (V,E,f) does not satisfy these conditions, it may be easily modified so that it does, without changing the vertex tier assignments. We describe a simple algorithm for this below.

We illustrate this algorithm with an example in Figure 5.1. The upper left plot illustrates an input to this algorithm: a randomly generated graph with vertex tier assignments indicated by the color listed in the legend. After executing lines 4-8 of the algorithm, intra-tier edges and tier bypassing edges are removing, yielding the network depicted in the upper right corner. At this point, we note the network may contain both "parentless" nodes (which are not linked to any node of lesser tier) and nodes with multiple such parents. Lines 12-14 remove multiple parents, yielding the network in the lower left corner. Finally, Lines 15-18 insert edges that link a (non-tier bypassing) parent node to any nodes without a parent, yielding the final network depicted in the lower right. We note that, in practice, any number criteria could be used to decide which parent nodes to remove and which to add to given nodes.

Algorithm 1 Tier bypassing correction. **Input:** (V, E, f), where $G = (V, E)$ is a graph and $f : V \to \{1, \ldots, T\}$. **Output:** E' such that (V, E', f) does not tier bypass.

procedure TIERFIX(V, E, f)
  $E' \leftarrow E$
  Enforces Conditions 1 and 2.
  for $e = \{u, v\} \in E$ **do**
    if $|f(u) - f(v)| > 1$ **or** $f(u) = f(v)$ **then**
      $E' \leftarrow E' \setminus \{u, v\}$
    end if
  end for
  Enforces Condition 4.
  for $v \in V$ **do**
    $S = \{\{u, v\} : \{u, v\} \in E', f(u) < f(v)\}$
    if $|S| > 1$ **then**
      Choose $C \subseteq S$ with $|C| = |S| - 1$
      $E' \leftarrow E' \setminus C$
    else if $|S| = 0$ **then**
      Choose $u \in V'$ with $f(v) - f(u) = 1$
      $E' \leftarrow E' \cup \{u, v\}$
    end if
  end for

return E'
end procedure

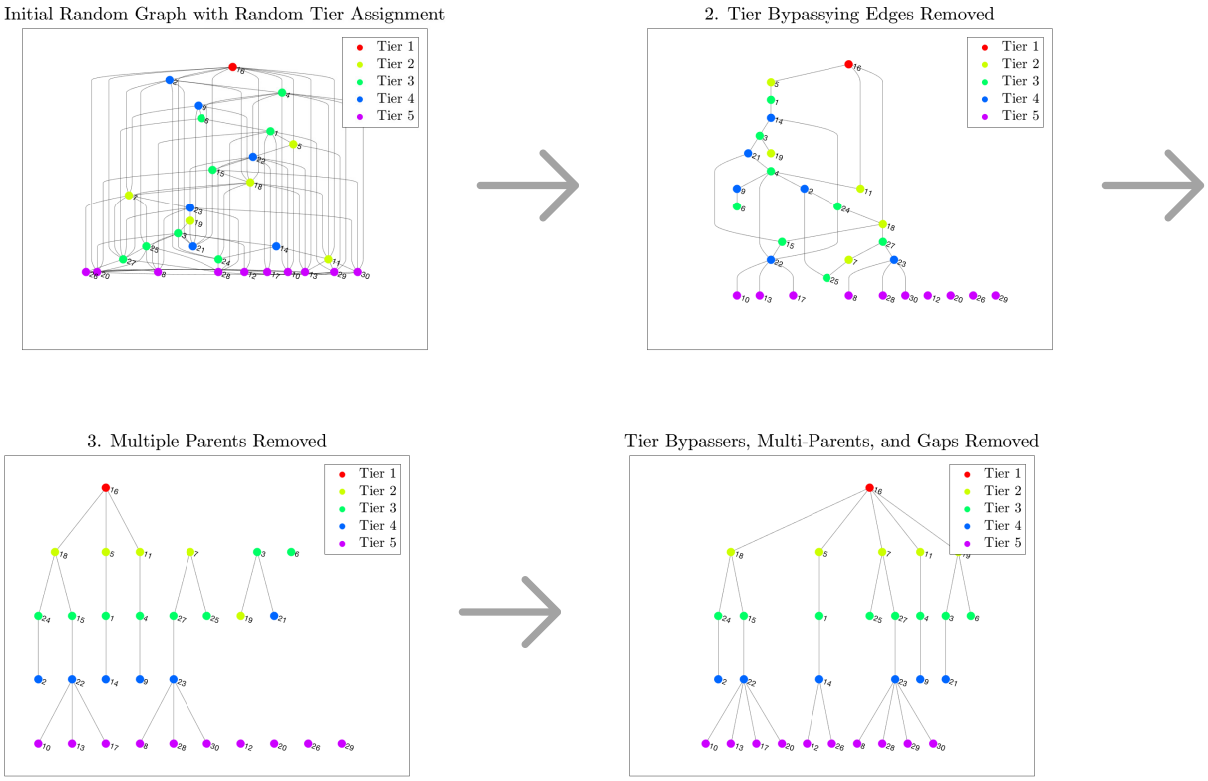


Figure 5.1. An example of modifying a network with random tier assignment to fix tier bypassing.

6.0 References

[1]Sinan G Aksoy, Emilie Purvine, Eduardo Cotilla-Sanchez, and Mahantesh Halappanavar. A generative graph model for electrical infrastructure networks. *Journal of Complex Networks*, aug 2018.

[2]Albert-Laszlo Barabasi. *Network Science*. Cambridge University Press, 2016.

[3]Norman Biggs. *Algebraic Graph Theory*. Cambridge University Press, 1974.

[4]Adam B. Birchfield, Ti Xu, Kathleen M. Gegner, Komal S. Shetye, and Thomas J. Over-bye. Grid structural characteristics as validation criteria for synthetic networks. *IEEE Transactions on Power Systems*, 32(4):3258–3265, jul 2017.

[5]Andries E. Brouwer and Willem H. Haemers. *Spectra of Graphs*. Springer New York, 2012.

[6]Fan RK Chung and Linyuan Lu. *Complex graphs and networks*. Number 107 in CBMS Regional Conference Series in Mathematics. American Mathematical Society Providence, 2006.

[7]Chris Godsil and Gordon Royle. *Algebraic Graph Theory*. Springer New York, 2001.

[8]Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[9]Y. F. Hu. Efficient and high quality force-directed graph drawing. *The Mathematica Journal*, pages 37–71, 2005.

[10]Karl Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10(1):96–115, 1927.

[11]Bojan Mohar. Isoperimetric numbers of graphs. *Journal of Combinatorial Theory, Series B*, 47(3):274–291, dec 1989.

[12]Giuliano Andrea Pagani and Marco Aiello. The power grid as a complex network: A survey. *Physica A: Statistical Mechanics and its Applications*, 392(11):2688 – 2700, 2013.

[13]Jeffrey D Taft. Architectural basis for highly distributed transactive power grids: Frameworks, networks, and grid codes. Technical Report PNNL-25480, Pacific Northwest National Lab (PNNL), Richland, WA (United States), 2016. Available online: https://gridarchitecture.pnnl.gov/media/advanced/Architectural%20Basis%20for%20Highly%20Distributed%20Transactive%20Power%20Grids_final.pdf

Appendix A

Table of Examples

Appendix A – Table of Examples

In order to help garner intuition about basic graph properties, Table A.1 provides of some sample networks and metrics computed on those networks. Specifically, each row in the table corresponds to the graph illustrated, and each column corresponds to a particular network metric. These metrics are:

- $|E|/|V|$: the ratio of the number of vertices to edges. For *sparse* graphs, this ratio is smaller and close to 1, for *dense* graphs, this ratio is closer to $|V|$.
- Diameter: The length of the longest shortest path in the graph. Informally, the most hops you'd ever need to take to get from one vertex to another (e.g. "six degrees of separation" means the graph diameter is 6).
- Average Distance: The average shortest path length between pairs of vertices.
- Clustering coefficient (CC): The clustering coefficient of a vertex is the proportion of its neighbors which are linked. The clustering coefficient of a graph is the average clustering coefficient over all vertices.
- Assortativity coefficient: The Pearson correlation coefficient of degree between pairs of linked nodes. This is a numerical measure, between -1 and 1, of the tendency of vertices of a certain degree to connect with vertices of the same degree. If a network is disassortative, then its assortativity coefficient is negative, meaning that high degree vertices tend to link to low-degree vertices.
- λ_1: The second eigenvalue of the normalized Laplacian matrix of the graph. This is a numerical measure of bottleneckedness that approximates the Cheeger constant. For more, see Section 3.3.

The example graphs considered include highly structured examples, such at the barbell graph, complete graph, ternary tree, and star graph (constituting the first 4 rows) as well as less symmetrical networks. The last two networks depicted are sample two-tier networks, such as are common in communication systems. For clarity, in each column we highlight the cell with the maximum and minimum values for that metric amongst the graphs in this table, in red and blue, respectively. For instance, we observe that the star graph (the fourth row) is the sparsest graph, and has the lowest clustering coefficient and assortativity coefficient.

While we have started the process of providing interpretations of these metrics in the context of power systems, work remains to be done to turn these into a regular tool set and methodology for evaluation and modification of grid structures.

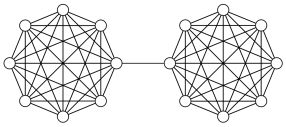
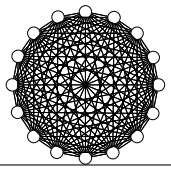
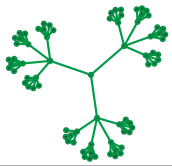
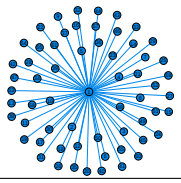
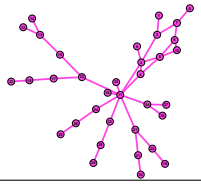
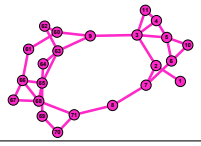
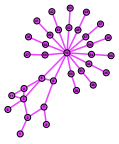
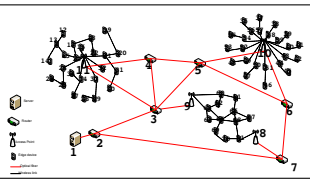
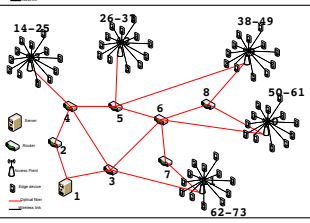
	$\lvert E\rvert/\lvert V\rvert$	Diam	Dist	CC	Assort.	λ_1
	3.5	3	1.93	0.9687	-0.02	0.0288
	7.5	1	1	1	–	1.07
	0.99	6	4.69	0	–0.81	0.0162
	0.98	2	1.97	0	-1	1
	1.09	8	3.89	0.06	-0.08	0.045
	1.43	6	3.21	0.3	-0.11	0.0614
	1.09	6	3.19	0.13	-0.82	0.0727
	1.14	10	5.03	0.098	-0.16	0.0219
	1.05	6	4.13	0.0342	-0.9	0.027

Table A.1. A table of graphs and selected graph metrics. The largest and smallest values for each metric are noted in each column by highlighting that cell in red and blue, respectively.

U.S. DEPARTMENT OF
ENERGY

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)
www.pnl.gov