A generative graph model for electrical infrastructure networks

Sinan G. Aksoy∗  Emilie Purvine†  Eduardo Cotilla-Sanchez‡  Mahantesh Halappanavar§

December 1, 2017

Abstract

We propose a generative graph model for electrical infrastructure networks that accounts for heterogeneity in both node and edge type. To inform the design of this model, we analyze the properties of power grid graphs derived from the U.S. Eastern Interconnection, Texas Interconnection, and Poland transmission system power grids. Across these datasets, we find subgraphs induced by nodes of the same voltage level exhibit shared structural properties atypical to small-world networks, including low local clustering, large diameter and large average distance. On the other hand, we find subgraphs induced by transformer edges linking nodes of different voltage types contain a more limited structure, consisting mainly of small, disjoint star graphs. The goal of our proposed model is to match both these inter and intra-network properties by proceeding in two phases: we first generate subgraphs for each voltage level and then generate transformer edges that connect these subgraphs. The first phase of the model adapts the Chung-Lu random graph model, taking desired vertex degrees and desired diameter as inputs, while the second phase of the model is based on a simpler random star graph generation process. We test the performance of our model by comparing its output across many runs to the aforementioned real data. In nearly all categories tested, we find our model is more accurate in reproducing the unusual mixture of properties apparent in the data than the Chung-Lu model. We also include graph visualization comparisons, as well as a brief analysis of edge-deletion resiliency. Ultimately, our model may be used to generate synthetic graph data, test hypotheses and algorithms at different scales, and serve as a baseline model on top of which further electrical network properties, device models, and interdependencies to other networks, may be appended.

Keywords: power grid graph model, Chung-Lu model, network-of-networks.

1 Introduction

Graph theoretic approaches to modeling the electric power grid aim to leverage tools in network science to analyze its topological structure. This area of research spans a variety of applications, including vulnerability analysis [2, 17], controlled islanding [3, 33], locational marginal pricing [9], and the location of sensors [4]. In its most basic form, the electrical grid is represented as a graph, where the vertices represent generators, substations, or loads, while edges represent power transmission lines or a voltage transformer.

A fundamental obstacle to applying network science to the power grid stems from the restricted access and proprietary nature of power grid data. In order to better facilitate the application of network science tools to the power grid, researchers need generative graph models to create synthetic power grids graphs. Ideally, these models require few, compact inputs (either measured from real data or generated artificially) and produce graphs exhibiting meaningful structural properties that can be quantitatively tuned according to user input. In this way, generative graph models serve as null models for testing hypotheses and algorithms at different scales. In this paper, we investigate structural properties of power grid graphs, and then apply those observations to design such a generative graph model. However, before salient features worth modeling can be selected, one must determine a suitable underlying data structure for the model.

In regard to this issue, Halappanavar et. al. [15] argue that the inherently heterogeneous nature of the real-world power grid necessitates graph models which, at a minimum, account for different node and edge types. Accordingly,
they argue, it may be inappropriate to apply graph algorithms or metrics to the entire power grid graph in a way that treats vertices or edges within a network as homogeneous in type. As an alternative, the authors propose using a more nuanced power grid graph model in which each vertex has a nominal voltage rating, and each edge is either a power transmission line (connecting two vertices of the same voltage) or transformer edge (connecting vertices of different voltage levels). In this approach, a power grid graph is then viewed as a “network-of-networks,” composed of vertex or edge homogenous subgraphs. That is, power grid graphs consist of a collection of same-voltage subgraphs for each voltage level, which are connected to each other via transformer edges. In this framework, network science tools can then be used to study and model the structure of the same-voltage subgraphs themselves, as well as how these subgraphs connect to each other through transformer edges.

In this work, we develop this viewpoint by designing a generative random graph model for the power grid that accounts for node and edge type. Our model aims to match, according to user-specified input, meaningful structural properties encountered in real power grid graph data. The paper is organized as follows: in Section 2 we describe the graph theoretic basics underlying the model, which largely mirrors that which is presented in [15]. Then, in Section 3 we investigate characteristics of real-world power grid graphs. We analyze both the same-voltage subgraphs, as well as the transformer edges between them, in the U.S. Eastern Interconnection, Texas Interconnection, and Poland transmission network. As we will soon explain, we find that the same-voltage subgraphs contain a mixture of properties typical for real-world networks, such as a heavy-tailed degree distribution, while also exhibiting atypical structure, such as low clustering, high diameter and high average distance. Given that many existing graph generation models are not able to capture this combination of properties, we propose a new graph model, described in Section 4, based on the well-known Chung-Lu model, which we call the Chung-Lu Chain model. We also describe our process for generating the transformer edges, as well as how both phases of the model may be used in conjunction to output the aggregate graph. In Section 5 we test the performance of our model by comparing its output over many runs against the real data. This comparison not only includes a number of graph metrics, but also graph visualizations and resiliency to single edge failure analysis. Lastly, we conclude and briefly mention avenues for future work in Section 6.

1.1 Prior Work

Before proceeding, we describe prior work on graph modeling of the electrical power grid. While in this paper we focus both on modeling the vertex and edge-homogenous subgraphs, as well as the aggregate graph, Hines et al. [16] considered the aggregate power grid graph associated with the IEEE 300-bus test case and the U.S. Eastern Interconnection power grid. They compared the structure of these aggregate graphs with comparably sized Erdős-Rényi, preferential-attachment, and small-world graphs and found these models lacked utility as stand-alone models for the power grid since they differed substantially in degree distribution, clustering, diameter and assortativity. As an alternative, they proposed a modification of the random geometric graph model called the “minimum-distance” model, which they showed outperformed the aforementioned models on the selected criteria. In [20] Wang, Scaglione and Thomas also analyze the structural properties of power grid graphs and propose their own generative model. They take a hierarchical approach to generating synthetic power grid graphs based on geographic zones; in contrast, our approach decomposes the network into vertex and edge-homogenous subgraphs based only on voltage rating and requires no geographic knowledge of the underlying power grid. For a broad, general survey on other research on modeling the power grid as a complex network, see [25]. Lastly, building off of work in [7], Birchfield et al. [6] puts forth an extensive list of 18 validation metrics for assessing the realism of synthetic power grid data. Some of these metrics, such as the number of substations per given voltage range, and the ratio of transmission lines to substations of a given voltage level, concern basic information that is directly built into our generative model. Other metrics mentioned, such as generator dispatch percentage and reactive power limits, require additional information that, while not included in our base generative model, may be later appended to its output data. Rather than studying how to realistically incorporate further electrical network information to power grid graph data, we emphasize that our focus here is on generating synthetic power grid graph data with accurate graph structure.

2 Preliminaries

2.1 Graph theory basics

A graph \( G = (V, E) \) is a set of vertices, \( V \) with \( |V| = n \), and set of edges, \( E \subseteq V \times V \), where each edge is an unordered pair of distinct vertices. Two vertices and \( i, j \in V \) are adjacent if \( \{i, j\} \in E \), and we call \( i \) and \( j \)

---

1 The U.S. power grid data used in this paper were obtained through the U.S. Critical Energy Infrastructure Information (CEII) request process. Poland open source data comes from MATPOWER: \[http://github.com/MATPOWER/matpower\]
neighbors. The degree of a vertex $i$, denoted $d_i$, is the number of vertices adjacent to $i$. The degree sequence of a graph, denoted $\vec{d} = (d_1, d_2, \ldots, d_n)$, is the list of degrees for each vertex in the graph.

A path of length $k$ between vertices $i$ and $j$ is a sequence of $k + 1$ distinct vertices, $i = i_0, \ldots, i_k = j$, where $i_\ell$ is adjacent to $i_{\ell+1}$ for $\ell = 0, \ldots, k - 1$. If $i = j$, then we call $i = i_0, \ldots, i_k = j$ a cycle of length $k$ and a cycle of length 3 is called a triangle. The distance between two vertices is the length of the shortest path between them, and is denoted $d(i, j)$.

A subset $S \subseteq V$ is connected if there is a path between all pairs of vertices in $S$. If $C \subseteq V$ is a maximally connected set (i.e. if there is no $S \supsetneq C$ which is also connected), we call $C$ a connected component of $G$. A tree is a connected graph with no cycles. The degree 1 vertices in a tree are called leaves. A k-star is a tree consisting of $k$ leaves connected to one degree $k$ vertex, called the center.

The subgraph induced by vertex set $S \subseteq V$ is the graph with vertex set $S$ and whose edge set consists of all edges in $E$ with both endpoints in $S$. Similarly, the subgraph induced by edge set $T \subseteq E$ is the graph with edge set $T$, and vertex set consisting of all endpoints of edges in $T$. We will denote these by $G[S]$ and $G[T]$ respectively. It will be clear from context whether we mean vertices or edges.

2.2 Graph metrics

In applications of graph theory one is often concerned with different quantities you can measure about the graph as a whole, or about each of the vertices. We call these metrics and will describe a few that are important in the context of power networks.

The degree distribution of a graph, $\vec{P} = (P(1), P(2), \ldots, P(\Delta))$, 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 $\Delta$ is the maximum degree in the graph. This is related, but not to be confused with the degree sequence as defined above in Section 2.1.

Two basic measures of distance in a graph are diameter and average distance. The average distance in a connected graph $G$ is the average distance between all pairs of vertices. The diameter of a connected graph $G$, denoted $\delta$, is the maximum distance over all pairs of vertices.

$$\text{avgDist}(G) = \frac{1}{|V|(|V| - 1)} \sum_{i \in V} \sum_{\substack{j \in V \atop j \neq i}} d(i, j), \quad \delta(G) = \max_{i, j \in V \atop i \neq j} d(i, j)$$

If a graph contains multiple connected components, we define the diameter as the maximum over the diameters of all connected components. Similarly, the average distance does not take into account pairs of vertices in two different connected components since there is no path between them.

The local clustering coefficient of a vertex (LCC) is used to measure how tightly connected the neighborhood of a vertex is. It is defined as

$$\text{lcc}(i) = \frac{\# \text{ triangles incident to vertex } i}{\binom{d_i}{2}}$$

Stated equivalently, the numerator is the number of edges that connect between neighbors of vertex $i$ while the denominator is the number of edges that could possibly connect neighbors of $i$. A vertex with a high LCC has very tightly connected neighbors, while a low LCC means that neighbors are not well-connected. Note that the local clustering coefficient of a degree 1 vertex is undefined. The local clustering coefficient of a graph is the average local clustering coefficient over all vertices for which the LCC is defined.

2.3 The power grid as a graph

We associate a graph with a power grid in the same manner as described in [15]. Namely, power-grid graphs are undirected graphs in which each vertex additionally has a “type” according to its voltage level. More formally, a power-grid graph is $G = (V, E, f, X')$ where $V$ is a set of vertices representing power stations, buses, generators, loads, etc., $E$ is a set of edges representing power lines connecting the vertices, $X'$ is a set of possible voltage levels, and $f : V \rightarrow X'$ assigns a voltage level to each vertex. We further call an edge $\{i, j\}$ a transformer edge if $f(i) \neq f(j)$.

As introduced in [15], a power grid graph with multiple voltage levels can be decomposed as the union of same-voltage subgraphs and transformer edges. More precisely, the subgraph induced by voltage level $X \in X$ consists of all vertices of voltage $X$ and all edges between them. With slight abuse of notation, we denote this by $G[X]$, where $X$ denotes the set of vertices of voltage level $X$. Though $X$ is not a set of vertices, it can be uniquely identified with the set of vertices for which $f(i) = X$. Transformer edges are thus edges between two vertices in different same-voltage subgraphs. In this way, the entire power-grid graph is given by the union of all same-voltage subgraphs and the set of transformer edges.
In Table 1 we present the basic notation we will use to refer to power-grid graphs. In the remainder of the paper, whenever the voltage level \( X \) is clear from context, we will drop the superscript (e.g. write \( d \) instead of \( d^X \)). In Figure 1 we show an example of a power grid graph with three voltage levels, \( X = \{X_1, X_2, X_3\} \). The vertices for the three different same-voltage subgraphs are shown in red, green, and orange and the transformer edges are in blue. Notice that each same-voltage subgraph may have multiple connected components.

![Figure 1: A sample power grid graph with three voltage levels. Vertices in \( G[X_1] \) are in red, in \( G[X_2] \) are in green, and in \( G[X_3] \) are in orange. The transformer edges between two different voltage levels are in blue.](image)

### 3 Characteristics of power grid graphs

In this section we will present summaries of data from three real-world power grid graphs: U.S. Eastern Interconnection, Texas Interconnection, and the Poland transmission system. In each case we compute the degree distribution, average distance, diameter, and average local clustering coefficient for all of the same-voltage subgraphs as well as the transformer edges graph. We will show these results, summarize the trends, and provide a justification as to why power grid graphs might have these common properties.

#### 3.1 Same-voltage subgraphs

In Table 1 we report the vertex and edge count, diameter, average distance, and average local clustering coefficient computed on each of the same-voltage subgraphs found in the Eastern, Texas, and Polish networks. There are two major points to make in looking at this table. First, the average distance and diameter values are typically much closer to being on the order of \( \sqrt{n} \) rather than something more like \( \log(n) \), for example. This distinction between \( \sqrt{n} \) and \( \log(n) \) will become important when we talk about generative models in Section 4.1 since most random graph models produce graphs with diameter and average distance that are roughly logarithmic in the number of vertices. Second, notice that the clustering coefficient is always very small. Except in the case of Poland’s 400 kV subgraph, the smallest graph in the group, the average local clustering coefficient is less than 0.10, and typically much less. This means that for most vertices in the same-voltage subgraphs, the neighborhoods contain very few triangles. This does not mean that cycles are rare, only that the cycles must be longer than three edges. In fact, since power grids must be highly resilient to failure of single edges (transmission lines), there are many cycles. But because each transmission line costs money and time to build, the extra redundancy given by a triangle is often outweighed by the cost of commissioning.

In Figure 2 we show the degree distributions, on a log-log scale, for all five same-voltage subgraphs of the Eastern Interconnection. Degree distributions from each of the other networks show similar trends but are not pictured here. The approximate linear shape of these plots indicate a heavy-tailed degree distribution. Heavy-tailed degree distributions are ubiquitous in real-world graphs, including social networks and the connectivity structure of the internet [5]. However, a key difference between many such real-world networks and the power grid is that the power grid networks represent physical infrastructure. Transmission lines are physical connections between generators, buses, loads; in contrast, in a social network or the internet the link may be symbolic and cost may not be strongly influenced by a physical distance.

---

2 In fact, \( \delta(G)/\sqrt{n} \) ranges between 0.5 and 1.5 whereas \( \delta(G)/\log(n) \) \( \in [1, 14.5] \). Similarly, \( \text{avgDist}(G)/\sqrt{n} \) \( \in [1.5, 4] \) and \( \text{avgDist}(G)/\log(n) \) \( \in [3.5, 41] \).
Table 2: Vertex and edge count, diameter, average distance, and local clustering coefficient for Eastern, Texas, and Polish transmission networks.

| Voltage Level | |V| | |E| Diameter | Av. Distance | Local CC |
|---------------|----------|----------|----------|----------|----------|----------|----------|
| Eastern: 138 kV | 12997 | 15752 | 386 | 135.117 | 0.049 |
| Eastern: 230 kV | 5691 | 6831 | 211 | 73.503 | 0.033 |
| Eastern: 345 kV | 1066 | 1360 | 78 | 25.038 | 0.073 |
| Eastern: 500 kV | 370 | 377 | 48 | 18.945 | 0.092 |
| Eastern: 765 kV | 79 | 112 | 16 | 6.204 | 0.037 |
| Texas: 69 kV | 1169 | 1257 | 134 | 47.978 | 0.007 |
| Texas: 138 kV | 2768 | 3272 | 84 | 32.466 | 0.019 |
| Texas: 345 kV | 208 | 290 | 22 | 8.320 | 0.062 |
| Poland: 110 kV | 2024 | 2302 | 92 | 37.661 | 0.008 |
| Poland: 220 kV | 135 | 174 | 20 | 7.899 | 0.032 |
| Poland: 400 kV | 50 | 58 | 17 | 6.484 | 0.141 |

Figure 2: Degree distribution, displayed on a log-log scale, for all five same-voltage subgraphs of the U.S. Eastern Interconnection.

3.2 Transformer edges graph

Recall that an edge is considered part of the transformer subgraph if its endpoints have different voltage levels. Upon studying the real transformer subgraphs for the Eastern, Texas, and Polish networks, we noticed that they tend to have many small connected components, almost all of which are star graphs. Recall that a star graph consists of a single vertex of degree $k$ connected to $k$ vertices of degree 1. In Table 3a we present the number of connected components alongside the number of non-star components for the voltage pairwise transformer subgraphs of the U.S. Eastern Interconnection. For example, the first row indicates that the transformer subgraph consisting only of edges between the 138 kV vertices and the 230 kV vertices of the U.S. Eastern Interconnection consists of 970 connected components, only five of which are not stars. Additionally, Table 3b contains information about component sizes and counts for the entire transformer subgraph of the U.S. Eastern Interconnection. The information for the other networks we studied is similar, but omitted for the sake of brevity. The main conclusion to draw from these tables is that a vast majority of the connected components in the transformer subgraph are $k$-stars. Comparing Table 3a with Table 3b we see that the majority of the (already few) non-stars are created when we consider all of the transformer edges together, rather than looking at only the pairwise voltage components. Out of a total of 1,772 connected components in the transformer subgraph only 21 are not stars, and in all of the pairwise voltage level graphs there are only 6 non-star components. This observation will be used to build our generative model of the transformer subgraph in Section 4.4. Rather than a generic random graph model, we will utilize a more specialized algorithm to produce random stars in order to more closely match the real graphs we wish to mimic.

The occurrence of star components might be explained by the nature of the edges themselves. Transforming voltage from one level to another occurs at an electrical substation. These substations can be large or small; when they are large, they may send transformed power out in multiple directions, yielding the star pattern in the graph. Additionally, though not evident in this table, we observed that the non-star components in the transformer subgraph are almost never cycles. This also makes sense from an engineering perspective: when transforming from voltage $X$ to voltage $Y$, it is typically more economically prudent to push power through a relatively long distance after stepping up in voltage.
The number of non-star components of different sizes in the transformer graph between each pair of voltage levels (3a) and the aggregate transformer graph (3b) in the U.S. Eastern Interconnection.

4 A generative model

In the previous section we identified a number of characteristics that consistently appear in same-voltage subgraphs, as well as features that are common to the transformer edges subgraph. The same-voltage subgraphs, which account for the majority of edges in a power grid graph, have heavy-tailed degree distributions, large diameter and average distance relative to the number of vertices, and low clustering coefficients. The transformer edges graph also has a distinctive, albeit much less rich, structure: in our data, we found that $G[T]$ consists almost entirely of disjoint, small-degree stars. We now turn our attention to random graph models. First we describe distance properties of well-known random graph models, which precludes their use in the power grid setting. Then we introduce our new model which has a favorable distance property for the power grid application.

4.1 Distances in random graph models

Small diameter and average distance are often cited as a ubiquitous property of real-world networks. Indeed, in addition to high clustering coefficients, a key defining property of small-world networks is that the distance between two randomly selected vertices is proportional to the logarithm of the number of the number of vertices $|V|$. The study of diameter and average distance in random graph models has a rich and extensive history. In particular, depending on the random model one considers and assumptions one places on its inputs, there are a plethora of results about the asymptotic behavior of diameter and average distance. In both regards, the same-voltage subgraphs of power-grid graphs stand in stark contrast. In Table 4, using standard asymptotic notation, we list some of these for well-known random graph models under various mild assumptions. In this table, $n$ denotes the number of vertices in the graph.

Recall our observation in the prior section in which both the diameter and average distance are much closer to $\sqrt{n}$ than to $\log(n)$ in the real power grid same-voltage subgraphs. Consequently, a direct application of these well-known models will not suffice for the purposes of generating graphs with large diameter and average distance, unless one introduces further modifications. Our proposed model for the same-voltage subgraphs of the power grid does precisely this. Motivated by the Chung-Lu model’s effectiveness in matching heavy-tailed degree distributions, its documented tendency to produce graphs with low-clustering coefficients, its simplicity, and its efficient implementations, we design a heavily-adapted version of the Chung-Lu model, which we call the CLC model. As we will soon explain in detail, whereas the Chung-Lu model takes a desired degree sequence as its sole input, our CLC model additionally takes desired diameter as a second input.

Our generative model for transformer edges aims to replicate the disjoint star structure observed in power grid graphs while also matching desired transformer degrees between each pair of voltages. Accordingly, our model for $G[T]$ is based on random star generation. While both of these models function independently, they may be used together to output the entire power-grid graph $G$. Before describing each phase formally, we first give a brief overview of the model.

---

A function $f(n) = O(g(n))$ if for all sufficiently large values of $n$, there is a positive constant $c$ such that $|f(n)| \leq c \cdot |g(n)|$. We say $f(n) = \Theta(g(n))$ if both $f(n) = O(g(n))$ and $g(n) = O(f(n))$, and say $f(n) = o(g(n))$ if $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$.

---

Table 3: The number of non-star components of different sizes in the transformer graph between each pair of voltage levels (3a) and the aggregate transformer graph (3b) in the U.S. Eastern Interconnection.
4.2 Model overview

Our generative model for the power grid proceeds in two “phases” and is described formally by the pseudocode in Algorithms 1–3 in Appendix A.2. In the first phase we model each same-voltage subgraph, G[X], separately. For each X, this phase takes two inputs: \( \bar{d}^X \), the desired degree sequence, and \( \delta^X \), the desired diameter. These inputs are initially used in a preprocessing stage, described in the function \( \text{SETUP}(\bar{d}, \delta) \) in Algorithm 1 which groups vertices and records information to be used when generating the output graph.

Using the vertex grouping information along with an updated degree sequence, both recorded in preprocessing, the actual graph generation procedure for Phase 1 is described in the function \( \text{CLC}(\bar{d}, \delta, S) \) in Algorithm 2. Here \( \bar{d}, \delta, S \) are the output from \( \text{SETUP}(\bar{d}, \delta) \). Loosely speaking, this model creates a number of smaller Chung–Lu random graphs which are linked together in a special “chain-like” structure to achieve the desired diameter. It is for this reason we call the Phase 1 model the Chung–Lu Chain model.

The second phase of our model creates the transformer edges graph, \( T[X, Y] \), between every pair of voltage levels, X and Y, separately. For each X, Y voltage pair we only need the desired transformer degrees of each voltage level with respect to each other, \( \bar{t}[X, Y] \) and \( \bar{t}[Y, X] \), as inputs. Based on our finding in Section 3.2 that the transformer edges of a power-grid graph consist almost entirely of disjoint stars, the graph generation process is accordingly based on random star generation. The procedure is described in function \( \text{STARS}(\bar{t}[X, Y], \bar{t}[Y, X]) \) in Algorithm 3.

We note that Phase 1 and Phase 2 operate independently in the sense that none of the inputs required for Phase 1 are required for Phase 2, and furthermore, no information about the graphs returned by Phase 1 is necessary to run Phase 2. This allows the user to control how inputs for Phase 1 (the desired degrees for the same-voltage subgraphs) may be correlated with the inputs to Phase 2 (the desired transformer degrees for a pair of same-voltage subgraphs).

In addition to this increased flexibility, using separate phases for the same-voltage and transformer edge subgraphs is a practical necessity, as these graph structures differ radically and thus benefit from individually tailored generation processes. Ultimately, the entire power grid graph \( G \) may be generated by applying Phase 1 to each voltage level, applying Phase 2 to each pair of voltage levels, and taking the union of all returned edge sets. The function \( \text{CLCSTARS} \) in Algorithm 4 describes how to use both phases to generate the entire power grid graph on \( k \) voltage levels.

### 4.3 Phase 1: the Chung-Lu Chain model

Our random graph model for the same-voltage subgraphs consists of two algorithms: (1) a preprocessing stage and (2) a graph generation stage. Given desired vertex degrees and desired graph diameter, the preprocessing is done in Algorithm 1 to randomly partition vertices into boxes which are key to matching the large distances observed in power grid graphs. Among the vertices within each box, one vertex is marked as a diameter path vertex and in some boxes another is marked as a subdiameter path vertex. These selections involve subtle considerations to ensure that, in the graph generation stage of Algorithm 2 the output graph more accurately matches the desired properties specified by the inputs. These are explained in Appendix A.3.

In Algorithm 2 a separate Chung–Lu random graph is generated on the vertices in each of the boxes and a deterministic diameter path is generated on the vertices chosen in Algorithm 3. We note that since the vertices in each box will be modeled as a separate random graph, additional edges between vertices in different boxes are not possible, and hence the diameter of a realization of the CLC model must be at least as large as the length of this diameter path. Furthermore, a deterministic “subdiameter path” is also created on the chosen vertices. The purpose of this path is to facilitate a richer connectivity structure consistent with the data by allowing for the creation of alternate paths and
cycles which connect vertices assigned to different boxes. The existence of such paths is implied both by long cycles observed in visualizations of power grid graphs, as well as by computations of edge connectivity. Figure 3 shows a cartoon illustration of Phase 1.

![Figure 3](image)

**Figure 3:** The output of Algorithms 1 and 2. Algorithm 1 partitions vertices into boxes and identifies a single diameter path vertex in each box and a subdiameter path vertex in some boxes. Algorithm 2 constructs all edges. Each vertex is labeled with its desired degree. The larger vertices are the diameter path (upper) and subdiameter path (lower) vertices.

### 4.4 Phase 2: inserting transformer edges

Algorithm 3 generates the transformer edges between two subgraphs of different voltage levels X and Y. The desired transformer degree of each vertex in the subgraphs of voltage X and Y are provided as input. Following our observation in Section 3.2 that the transformer subgraphs appear to largely consist of star graphs, Algorithm 2 aims to match these desired transformer degrees by generating the transformer edges graph as a collection of random, disjoint star graphs. In order to achieve this, vertices with nonzero transformer degree are partitioned into two sets: vertices with transformer degree at least 2 and those with transformer degree 1. Vertices of degree at least \( k \geq 2 \) are intended to be the centers of a \( k \)-star whereas vertices of degree 1 are intended as the leaf vertices which connect to the center of the star. The stars are generated by randomly selecting a vertex from voltage X with transformer degree \( k \geq 2 \) and connecting this vertex with \( k \) randomly selected vertices from voltage Y that have desired transformer degree 1. The selected vertices are then removed from their respective sets and another vertex is selected until the degree \( k \geq 2 \) vertices from voltage X have been exhausted. The same procedure is then applied for the centers of \( k \)-stars centered at voltage Y vertices. Finally, the remaining degree 1 vertices from each voltage level are randomly matched with each other. An example output of this procedure is illustrated in Figure 4. In this example the voltage X transformer degrees are \( \langle 4, 3, 1 \rangle \) and voltage Y transformer degrees are \( \langle 2, 1, 1, 1, 1 \rangle \). In both voltage X and Y there are vertices which do not participate in the transformer graph (depicted by the smaller blank circles).

Lastly, we note that Algorithm 3 accounts for a special case in which the desired transformer degrees of all vertices cannot be realized as a collection of disjoint stars via the aforementioned process. Namely, if, at any point, there are insufficiently many degree-1 vertices from the remaining pool to generate the current \( k \)-star, the degree \( k \) vertex is

![Figure 4](image)

**Figure 4:** Example transformer graph between the voltage X subgraph and voltage Y subgraph. Small unlabeled circles are vertices within each subgraph with no transformer degree, numbered vertices are labeled with their desired transformer degree. Blue edges are transformer edges, gray are edges formed in Phase 1 within each same-voltage subgraph. Note: the boxes enclosing voltage X and Y subgraphs are different than those in the CLC model of Phase 1.

8
removed from its pool and placed in a “leftover bin”. The final part of the algorithm then generates a random bipartite Chung–Lu graph \(^\text{[11]}\) on the vertices in these leftover bins. In Appendix \(^\text{A.3}\) we describe a sufficient condition on the desired transformer degree sequences of two voltage levels which guarantees this special case will not occur. We note this condition is met by every pair of voltage levels in all our power grid graph data. Furthermore, satisfying this condition also guarantees that the actual transformer degrees of vertices in the output graph of Algorithm\(^\text{3}\) will match the input desired degrees exactly.

5 Comparison of model output to real data

After discussing observations within the real data (Section \(^\text{3}\)), and presenting our Chung-Lu Chain (CLC) and star generation algorithms (Section \(^\text{4}\)), we are now ready to summarize the results of the algorithm in comparison to the real world data. In particular, we first present results for Phase 1, the same-voltage subgraphs, then for the transformer edges, and finally for the aggregate graph, taking the union of the same-voltage subgraphs and the transformer edges. Recall that the measures which we are focusing on matching are: number of vertices and edges in the largest component, diameter, average distance, average local clustering coefficient, and degree distribution. We will discuss the measures separately in the following subsections.

As an aside, we found our CLC model also performed more favorably than Chung-Lu for a number of other metrics we tested, including degree correlation statistics like the assortativity coefficient, as well as geometric notions of connectivity like the spectral gap of the normalized Laplacian matrix (which is intimately related to “algebraic connectivity”). For brevity, we limit our presentation here to the aforementioned metrics due to their relevance to the aims of our model; nonetheless, we’ve included data on assortativity and the spectral gap in Appendix \(^\text{A.4}\) for the interested reader.

5.1 Same-voltage subgraphs

The bar charts in Figures 5a–5e show the relevant measures on same-voltage subgraphs compared across the real data (blue), our new CLC model (orange), and the traditional Chung-Lu model (gray) for comparison. Each same-voltage subgraph is treated separately and they are listed along the bottom of the chart. Both of the random models were run 100 times with the necessary input data being measured from the corresponding real subgraph, and the measures were collected for each run. For both Chung-Lu and CLC, the bar height is the average of those 100 trials, and error bars indicate the minimum and maximum values that were observed.

We must point out that the real same-voltage subgraphs may not be connected, but tend to contain a single large connected component and many very small connected components. Moreover, Chung-Lu and CLC are not guaranteed to return connected graphs. However, in [12] Chung and Lu proved that if expected average degree is strictly greater than 1 (a mild condition certainly met by our data), the Chung-Lu model will almost surely output a graph with a giant component. Experimentally, this is confirmed for both the Chung-Lu and CLC models. Therefore, for all cases including the real data we only report data corresponding to the largest connected component of each graph.

**Number of vertices and edges** In Figures 5a and 5b we show the comparison of the number of vertices and edges in the largest component across all test cases and the two generative models. It is clear from both of these plots that Chung-Lu and CLC are very similar to each other in vertex and edge count, and additionally tend to match the real data well. The worst matching appears to be the number of edges for the largest same-voltage subgraph, the Eastern 138 kV subgraph. But even in this case both are within 20% on the average with CLC performing slightly better than Chung-Lu. It is worth emphasizing that only (non-isolated) vertices in the largest component contribute to reported total vertex counts in both the original graph data and the random graph model data. The Chung-Lu model is never guaranteed to match largest component size nor total number of non-isolated vertices in the entire graph since any vertex may become isolated with nonzero probability. In order to ensure that that CLC better matches non-isolated vertex counts and largest component sizes in the original data, we use prior theoretical results [12] on the Chung-Lu model to adjust the inputted degree sequence of the CLC model (see Appendix \(^\text{A.3}\) for details).

**Diameter and average distance** These two measures, with comparisons in Figures 5c and 5d, begin to show the value of the CLC model as compared to the traditional Chung-Lu model. In all test cases and trials the CLC model generates a graph which matches the real diameter and average distance significantly more accurately than Chung-Lu. This is because of the use of the diameter and subdiameter paths in the CLC model. As was discussed in Section 4.1, the Chung-Lu model outputs a graph that (under mild assumptions – see admissible degree sequence in [12]) has diameter and average distance roughly \(\log(|V|)\) with high probability, whereas we observed that the diameter and average distance in the real data are on the order of \(\sqrt{|V|}\). Our CLC model is able to more closely approximate this large diameter and average distance. Moreover, because diameter is an input parameter to our CLC model we are able
Figure 5: Bar charts for comparison of measures in single-voltage subgraphs.

Average local clustering coefficient  The comparison plot for average local clustering coefficient is in Figure 5e. Here we plot the local clustering coefficient on a $[0, 1]$ y-axis since clustering coefficient values are always within that range. It is oft-noted that the Chung-Lu model tends to produce graphs of low clustering coefficients, whereas many real-world networks exhibit much larger clustering coefficients \cite{12, 19}. However, as we have observed, power grid graphs stand apart in this regard: their clustering coefficients appear to be small, typically below 0.1 in our data with only one exception (Polish 400 kV). Therefore, what is sometimes cited as a unrealistic property of the Chung-Lu model is in fact quite appropriate for power grid graphs. Since our model is just a set of Chung-Lu graphs connected via diameter and subdiameter paths it is expected, and experimentally confirmed, that CLC will also provide a small clustering coefficient as shown in this figure.

Further research has addressed how clustering coefficients tend to vary as the size of the network increases. Both theoretical and experimental research \cite{8, 24, 29} shows that, under a variety of assumptions, the local clustering coefficient of scale-free power law graphs tends towards zero or some positive constant as the network size approaches infinity. Accordingly, it is not surprising that when modeling networks with heavy-tailed degree distributions, networks of smaller size may have relatively larger local clustering coefficients compared to those of larger networks. Since the Chung-Lu Chain model “stitches together” smaller Chung-Lu random graphs, one might expect both that (1) the local clustering coefficient of the Chung-Lu Chain model exceeds that of the Chung-Lu model run on the same degree distribution; and (2) the local clustering coefficient is larger for Chung-Lu Chain graphs in which the expected number of vertices per box is smaller when compared to Chung-Lu Chain graphs in which the expected number of
vertices per box is larger. Such expectations are consistent with the data in Figure 5 which shows: (1) the Chung-Lu Chain model generating graphs with larger clustering coefficients than the corresponding Chung-Lu model on all datasets, and (2) the Chung-Lu Chain model has larger clustering coefficients on smaller graphs, like the Eastern 765 kV or Texas 345 kV, than on larger graphs like the Eastern 138 kV. We note that while local clustering coefficient is not a directly tunable parameter of the Chung-Lu Chain model, our data shows the model produces “small” clustering coefficients, roughly on the same order of magnitude as those of the original data.

**Degree distribution** The degree distribution is critical to match because of its fundamental nature and distinctive, heavy-tailed shape. These types of distributions are ubiquitous in real-world graphs and therefore important to capture in the model. Both in theory and practice, the Chung-Lu model (which matches desired degrees in expectation) is already well-known to match heavy-tailed degree distributions well. Thus, we skip this data in the comparison. However, as the Chung-Lu Chain model is heavily adapted, and contains deterministic diameter and subdiameter paths, it is not immediately clear to what extent Chung-Lu’s properties in matching degree distribution extend to the CLC model – particularly at different scales. But, in Figures 6–7 we see that visually the degree distribution in the resulting CLC model (blue) closely matches the real degree distribution (red). Beyond a visual comparison, selecting a quantitative metric with which to compare the distance between two general degree distribution is a complex problem. Some popular metrics, like the Kolmogorov-Smirnov statistic, have been criticized (see [28]) for producing counter-intuitive results for graphs whose degree distributions one might consider similar (e.g. the $n$-vertex complete graph vs the $n−1$-vertex complete graph) or different (e.g. a $n$-vertex matching vs a $n$-vertex star). One promising direction worthy of further research is the recently proposed Relative Hausdorff measure introduced in [29]. However, such an investigation is outside the scope and aims of the current work.

![Figure 6: Degree distribution, displayed on a log-log scale, for all five same-voltage subgraphs of the Eastern Interconnect.](image)

**Visualizations** Finally, we offer visual comparisons of selected same-voltage subgraphs to show that in addition to all of the measures that we capture, the visual appearance of our randomly generated graphs is similar to that of the real graphs. This strong similarity in the visualization is consistent with the similar structural properties we measured in Section 5.1. We also include visualizations of corresponding Chung-Lu graphs, which suggest the visualization similarity between our model and the original data is not merely a consequence of having similar degree distributions. In all cases, the visualization layouts were done in a similar fashion, using the Yifan Hu layout [18], by repelling vertices that are not connected and attracting those that are. In Figures 8a–8c we show the Eastern Interconnect 345 kV original, CLC, and CL graphs. Since these graphs are quite large it’s difficult to tell exactly how similar they are. Next, Figures 8d–8f compares smaller systems, the Texas Interconnect 345 kV original, CLC, and CL graphs. Again the graphs are fairly large, but notice the similarity in the presence of a few longer cycles, some shorter cycles,
Figure 7: Degree distribution, displayed on a log-log scale, for all three same-voltage subgraphs of the Texas Interconnect (top row) and Polish (bottom row).

and chains hanging off of the edges. Here, the smaller diameter and average distance of the CL graph relative to the original and CLC graphs are apparent in the visualization. Finally, we show a comparison between three small graphs, the Polish 400 kV original, CL, and CL graphs in Figures 8g-8i. In this case, they are small enough to easily compare visually. Notice the few small cycles on the left of both graphs as well as the two long chains going off to the right. In both the original and CLC graphs, these long chains have a few small branches.

5.2 Transformer subgraphs

As was noted in Section 3.2, the real transformer graphs tend to consist mostly of star graphs, especially when restricted to only one pair of voltage levels. When all of the transformer edges for a particular network are aggregated, the resulting transformer graph still consists almost entirely of star graphs. It was for that reason that we created our model to specifically construct pairwise transformer subgraphs as collections of stars. Because of this choice of model, and the structure of the transformer subgraphs themselves, it does not make sense to compare the degree distributions, diameter, average distance, or clustering coefficient between the real data and the model. Degree distributions will always be matched exactly because of how the stars are formed; in each pairwise transformer subgraph we are guaranteed to have diameter 2 in the model (all stars have diameter 2, the distance from one leaf to another within the same star), average distance is similarly uninteresting; and clustering coefficient is necessarily zero in star graphs as there are no cycles. However, this does not mean that the Phase 2 graph is structurally identical to the original graph, nor does it mean that Phase 2 produces graphs that are structurally identical across different runs of the model. This is because when we consider the aggregate of the union of all transformer edges in Phase 2 the stars can interact in different ways to create larger structures. The only interesting comparison from the aforementioned list, therefore, is that between the number of non-star components produced by the model versus observed in the real data.

In Table 5 we report the total number of components, and the number of non-star components, of each size in the transformer subgraph for each of our three test cases. We give the values that were observed in the real data as well as those generated by the model. In all three example networks the number of components of a given size are matched very closely, always on the same order of magnitude and often within 10% of the original observed data. Similarly the number of non-star components is matched very closely, if not exactly, in almost all cases.

5.3 Entire aggregate graph

We’ve shown our models of the same-voltage subgraphs and transformer edges agree well on the test networks. While our model is explicitly designed to match these vertex and edge-homogenous subgraphs, we will now put these
Figure 8: Visual comparison of graph structure for three different same-voltage subgraphs.
### Table 5: Non-star component counts by size in the transformer subgraphs (Original vs. our simulated Phase 2).

Together to form the aggregate graph and assess our model’s accuracy. As we will see, a number of properties, such as degree distribution, number of vertices and edges in the largest component, and local clustering coefficients are closely matched in the aggregate graph largely by virtue of the local matches for each subgraph. However, while our model is quite accurate in matching diameter and average distance of same-voltage subgraphs, it may be less accurate in matching the *heterogenous diameter* of the aggregate graphs (which is computed from paths which may contain a mixture of transformer and non-transformer edges). As we will explain, this is because transformer edges can shorten distances between vertices that were previously further apart in their same-voltage subgraph. In all cases, we compare our models accuracy against that of the traditional Chung-Lu model applied to the aggregate graph. If the hypothesis is that matching degree distribution is useful then Chung-Lu, a model matching desired degrees in expectation, is a basic null model for comparison. We observe Chung-Lu is less accurate for distance measures and clustering; as was the case for the same-voltage subgraphs, this confirms more relevant structure within the aggregate power grid beyond degree distribution.

In addition to the metrics already discussed, we will consider *resiliency*. Entire power grids must satisfy \(N - 1\) security, meaning that if a single power line (edge) or bus (vertex) is removed the grid must still be operational. With this in mind we consider the effect of removing a single edge of the aggregate graph. In particular, for every non-trivial cut edge (an edge which, if removed, disconnects more than a single vertex from the rest of the graph) we calculate the size of the largest connected component that remains when it is removed. We compare the distribution of these sizes for the real data and our model. We explain the methodology in more detail within that subsection.

### Number of vertices and edges, degree distribution, and local clustering coefficient

In Figure 9a-9b we show a comparison of the vertex and edge count for the Eastern, Texas, and Polish networks. Both the CLC+Stars model, as well as the Chung-Lu model, perform comparably in matching the number of vertices and edges in the largest connected component of the aggregate graph. In Figure 10 we compare degree distributions. Since it is already well-known that the Chung-Lu model will match the degree distribution in the aggregate graph closely, its performance here is unsurprising. Our CLC+Star model also produces degree distributions comparable to the real degree distribution. The reasoning behind this close match globally is likely due to the convergence of our local matches: our same-voltage level CLC model is designed to match the degree within each voltage level and the disjoint star algorithm for the transformer edges matches the transformer degree exactly (under easily met assumptions) or within a very slim margin (otherwise). Lastly, in Figure 9c we compare average local clustering coefficients. As was the case for the same-voltage subgraphs, the average local clustering coefficient for the aggregate graph is extremely low. Figure 9d shows the comparison between the average local clustering coefficient in the real data and the two models for the aggregate graph. Again this is likely because triangles are expensive to make and that cost may outweigh the resilience that the triangle creates. One longer cycle creates similar resilience and is less costly than many triangles covering the same vertices. We again overestimate the average local clustering coefficient using CLC+Stars, but remain within the same order of magnitude, while the Chung-Lu is typically an order of magnitude lower than the real data.

### Diameter and average distance

Comparisons of the diameter and average distance in the real data against the CLC+Stars and Chung-Lu models are shown in Figure 9e-9f. Recall that in the same-voltage subgraphs the CLC
model achieved more accurate diameter and average distance than the Chung-Lu model. Now once we have stitched those graphs together using the transformer edges, modeled as collections of stars, the accuracy of the CLC+Stars method declines. We note that diameter of the aggregate graph is not a tunable parameter of our model. However, the diameter results for our model are still more accurate than for Chung-Lu on its own. For the larger network, the Eastern Interconnect, CLC+Stars has the worst performance. Here, because leaf nodes are chosen randomly in the star-generation process, the insertion of star graphs in Phase 2 created shorter paths between two vertices in a same-voltage subgraph that were previously further apart in graph distance. Nonetheless, the Texas Interconnect and Polish networks are fairly well-matched by our model.

Figure 9: Bar charts showing comparison for measures in aggregate data.

**Resiliency**

Real world power grids are required to be resilient to certain types of failures. For example, the loss of a single line or other component which is often referred to as \( N-1 \) contingency \[20\]. Resilience to this type of failure makes the system \( N-1 \) secure. When considering failures of multiple lines or components, this is called \( N-x \) contingency where \( x \) is the number of failures allowed. In the context of a graph, being resilient to \( N-1 \) contingency means that the removal of any single vertex or edge leaves the graph connected; or, if the graph is disconnected each piece can still function independently (for example, a generator can be found in each component).

We tested resiliency to the loss of single edges in our CLC+Stars model as well as the Chung-Lu model and compared both to the real data. Since our data does not have labels on the vertices to know where the generators are we instead considered the sizes of the smallest connected component remaining when a single edge is removed. We only considered nontrivial cut edges, those edges whose removal disconnects more than one vertex. In all networks, both real and synthetic, there are very few such edges – between 3-5% in the real data and Chung-Lu model, and 5-7% in the CLC+Stars model. In Figures 11a-11c we show a comparison of the sizes of the smallest connected component remaining after removal of single cut edges. These are frequency plots meaning that a point at \((x, y)\) indicates that there were \(y\) cut-edges for which the smallest connected component remaining when the edge is removed had \(x\) vertices. For example, the blue point in Figure 11a at \((2, 1000)\) means that there were 1000 cut edges in the CLC+Stars model of the Eastern interconnect for which the smallest remaining connected component had only 2 vertices.

In the Eastern and Texas networks the CLC+Stars method does a good job of catching the few cut edges that leave a rather large (in comparison) portion of the graph disconnected. This is likely due to our random stars model for the transformer edges. If there is a same-voltage connected component that only has one transformer edge linking it to the rest of the network then it will be disconnected when that transformer is cut. The Polish network seems to have fewer cut edges that disconnect larger portion of the graph and therefore our model does not match its resiliency as well as the other two. Notice also that the Chung-Lu model tends to underestimate the number of cut edges in general.
Entire graph visualizations. Due to the large number of vertices and edges in the entire power grid graphs (e.g. the Eastern Interconnection has 24K vertices and 31K edges), visualizing the entire power grid graph poses challenges. In particular, depicting every vertex and edge would result in a cluttered and potentially uninformative visualization. For this reason, it is advantageous to “reduce” the data while preserving some underlying structure. Here, we present one such compact visualization of the aggregate graph, which we call “interconnection graphs”.

Visualizations of interconnection graphs depict how transformer edges link connected components from different same-voltage subgraphs. In an interconnection graph, each connected component within a same-voltage subgraph is collapsed to a single vertex and two such vertices are connected by an edge if there exists a transformer edge linking any vertex in one component to any vertex in the other component. More formally, the interconnection graph $H$ of the entire power grid graph $G$ is defined as follows. Let $C_1, \ldots, C_k$ denote the connected components of the same-voltage subgraphs of $G$, and $T$ denote the set of transformer edges. Then $H$ has vertex set $V(H) = \{1, \ldots, k\}$ and edge set $E(H) = \{\{i, j\} : \exists\{u, v\} \in T$ with $u \in C_i$, $v \in C_j\}$. The weight of a vertex $i \in H$ is the number of vertices in $C_i$ in $G$ and the weight of a edge $\{i, j\}$ is the number of transformer edges between $C_i$ and $C_j$ in $G$. We note that, with the exception of the edge and vertex weights, an identical idea was proposed in [15].

In Figure 12 we compare visualizations of the interconnection graph derived from the original entire power grid graphs against those derived from our model (i.e. the output of Algorithm 4). The color of a vertex indicates the voltage level of the nodes in its corresponding connected component, while the size of a vertex and thickness of an edge are proportional to their weight as defined above. Given the relatively small nature of the visualizations, a number of qualitative similarities are readily apparent: first, the fact that nearly every edge connects to the largest vertex of each color type reflects that nearly every transformer edge is incident to the largest component of some voltage level. In other words, transformer edges between two components are rare when neither component is the largest within its respective same-voltage subgraph. Furthermore, we also observe that the thickest edges occur between the largest vertices of each color type. This reflects that the multiplicity of transformer edges between components is greatest when those components are the largest connected components.

More quantitative analysis of interconnection graph structure is made difficult by the fact that both vertices and edges are weighted in these networks, precluding many well-known graph metrics and analyses from immediate application to interconnection graphs. Nonetheless, while we believe such analysis would be worthwhile, we defer this to future work and simply present visualizations here.
6 Conclusion and future work

In order to account for the heterogeneity of real-world power grid networks, we designed a generative random graph model that distinguishes between nodes and edges of different type, according to their voltage level. We found that same-voltage subgraphs in real power grid graph data exhibited an unusual combination of structural properties which make modeling them accurately difficult. We proposed a two-phase model which generates both the same-voltage subgraphs as well as the transformer edges that connect them, according to tunable user-specified input. We found

![Comparison of interconnect graphs for each of the three test networks.](image)

**Figure 12**: Comparison of interconnect graphs for each of the three test networks.
that this model either matched or outperformed the Chung-Lu model in nearly all categories tested, particularly with regard to the large diameter and average distance observed in the data. The close match of these structural properties were also seemingly reflected in visualizations that, even at small scales, bore resemblance to the original graph data. Lastly, our model also produced graphs with comparable resiliency to single edge failures.

Rigorously modeling electrical network structure not only provides insight into the behavior of these complex systems, but also may be of use to practitioners. In particular, given that our model only requires desired degrees and diameter, practitioners may easily generate power grid-like graphs artificially, without having to extract the data from real-world power grid graphs. Since power grid graph data is not widely available, this may be particularly advantageous. In this way, practitioners may use this model to generate synthetic power grid graphs, test algorithms at different scales, or serve as a baseline on top of which further information about the power-grid may be incorporated. In particular, it is our hope that the flexibility of the model might allow for future exploration in how graph parameters like diameter and average distance are correlated with operational aspects of the power grid.

Many questions remain for future work. In particular, while there is no dearth of network analysis tools, more work is needed to understand the limitations and interpretation of graph parameters in the context of the real-world power grid. While we believe our model to be a step in the right direction, we are cognizant of the fact that the real-world power grid contains far more critical information than is captured by a power-grid graph, even when accounting for node and edge types as we’ve done. However, at the same time, synthetic models which do not accurately capture graph structure are destined to be ineffective, no matter how much information is appended to the graph. In this sense, we hope our model can serve as an accurate and reliable starting point for future exploration in how graph parameters on top of which one can incorporate further information about the grid. Lastly, another avenue for future work lies in incorporating time evolution into our model. In order to accurately capture these changes, however, careful analysis of snapshot data is needed.

**Funding.** This work was supported in part by the Applied Mathematics Program of the Office of Advanced Scientific Computing Research within the Office of Science of the U.S. Department of Energy (DOE) through the Multifaceted Mathematics for Complex Energy Systems (M2ACS) project.

**Acknowledgement.** The authors would like to thank Josh Tobin and Stephen Young for helpful discussions.

**References**

[1] Aksoy, S. G., Kolda, T. G. & Pinar, A. (2017) Measuring and modeling bipartite graphs with community structure. *Journal of Complex Networks*, 5(4), 581–603.

[2] Albert, R., Albert, I. & Nakarado, G. L. (2004) Structural vulnerability of the North American power grid. *Physical Review E*, 69(2), 025103.

[3] Amraee, T. & Saberi, H. (2017) Controlled islanding using transmission switching and load shedding for enhancing power grid resiliency. *International Journal of Electrical Power & Energy Systems*, 91(Supplement C), 135 – 143.

[4] Anderson, J. E. & Chakrabortty, A. (2012) Graph-theoretic algorithms for PMU placement in power systems under measurement observability constraints. in *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on*, pp. 617–622. IEEE.

[5] Barabási, A. & Bonabeau, E. (2003) Scale-Free Networks. *Scientific American*, 288(60-69).

[6] Birchfield, A. B., Schweitzer, E., Athari, M. H., Xu, T., Overbye, T. J., Scaglione, A. & Wang, Z. (2017) A Metric-Based Validation Process to Assess the Realism of Synthetic Power Grids. *Energies*, 10(8), 1233.

[7] Birchfield, A. B., Xu, T., Gegner, K. M., Shetye, K. S. & Overbye, T. J. (2016) Grid Structural Characteristics as Validation Criteria for Synthetic Networks. *IEEE Transactions on Power Systems*, 32(4), 3258–3265.

[8] Bollobás, B. & Riordan, O. M. (2003) Mathematical results on scale-free random graphs. in *Handbook of Graphs and Networks*, pp. 1–37. Wiley.

[9] Cheverez-Gonzalez, D. & DeMarco, C. L. (2009) Admissible locational marginal prices via Laplacian structure in network constraints. *IEEE Transactions on Power Systems*, 24(1), 125–133.

[10] Chung, F. (1996) *Spectral graph theory*, no. 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society Providence.
C. Hunnam, D. Fiedler, A. Alapanna Var, C. Hines, P. D. H. Atts, W. Inlaw, W. Hines, H. Ille, D. E. E. Ahdian, N. Ittel, S. Vander Stromová, V. A. Paganini, P. Ianari, A. Strogatz, E. Chapman, L. A. K., H. D. Y., F. A., Y. E., F. R. & L. O., S. T. E. H. A. D. H. R. I., C. (2005) Note on the heights of random recursive trees and random graphs with given expected degrees. Proceedings of the National Academy of Sciences, 99(25), 15879–15882.

Chung, F. & Lu, L. (2006) Complex graphs and networks. Complex networks, no. 107 in CBMS Regional Conference Series in Mathematics. American Mathematical Society Providence.

Dommers, S., van der Hofstad, R. & Hooghiemstra, G. (2010) Diameters in Preferential Attachment Models. Journal of Statistical Physics, 139(1), 72–107.

Fiedler, M. (1973) Algebraic connectivity of graphs. Czechoslovak mathematical journal, 23(2), 298–305.

Halappanavar, M., Cotilla-Sanchez, E., Hogan, E., Duncan, D., Zhenyu, Huang & Hines, P. D. H. (2015) A Network-of-Networks Model for Electrical Infrastructure Networks. arXiv:1512.01436.

Hines, P., Blumsack, S., Cotilla-Sanchez, E. & Barrows, C. (2010) The topological and electrical structure of power grids. in System Sciences (HICSS), 2010 43rd Hawaii International Conference on, pp. 1–10. IEEE.

Hines, P., Cotilla-Sanchez, E. & Blumsack, S. (2010) Do topological models provide good information about electricity infrastructure vulnerability?. Chaos: An Interdisciplinary Journal of Nonlinear Science, 20(3), 033122.

Hu, Y. F. (2005) Efficient and high quality force-directed graph drawing. The Mathematica Journal, pp. 37–71.

Kolda, T. G., Pinar, A., Plantenga, T. & Seshadhri, C. (2014) A Scalable Generative Graph Model with Community Structure. SIAM J. Sci. Comput., 36(5), C424–C452.

Kundur, P., Paserba, J., Ajjarapu, V., Andersson, G., Bose, A., van Cutsem, T., Canizares, C., Hatziargyriou, N., Hill, D., Vittal, V., Stankovic, A. & Taylor, C. (2004) Definition and Classification of Power System Stability IEEE/CIGRE Joint Task Force on Stability Terms and Definitions. IEEE Transactions on Power Systems, 19(3), 1387–1401.

Mahdian, M. & Xu, Y. (2010) Stochastic kronecker graphs. Random Struct. Alg., 38(4), 453–466.

Miller, J. & Hagberg, A. (2011) Efficient generation of networks with given expected degrees. Algorithms and models for the web graph, pp. 115–126.

Newman, M. E. (2002) Assortative mixing in networks. Physical review letters, 89(20), 208701.

Ostroumova, L., Ryabchenko, A. & Samovat, E. (2013) Generalized Preferential Attachment: Tunable Power-Law Degree Distribution and Clustering Coefficientpp. 185–202. Springer International Publishing, Cham.

Pagani, G. A. & Aiello, M. (2013) The Power Grid as a complex network: A survey. Physica A: Statistical Mechanics and its Applications, 392(11), 2688 – 2700.

Pinar, A., Seshadhri, C. & Kolda, T. G. (2012) The Similarity between Stochastic Kronecker and Chung-Lu Graph Models. in Proceedings of the 2012 SIAM International Conference on Data Mining, pp. 1071–1082. Society for Industrial & Applied Mathematics (SIAM).

Pittel, B. (1994) Note on the heights of random recursive trees and random m-ary search trees. Random Struct. Alg., 5(2), 337–347.

Simpson, O., Seshadhri, C. & Mcgregor, A. (2015) Catching the head, tail, and everything in between: a streaming algorithm for the degree distribution. 2015 IEEE International Conference on Data Mining (ICDM), pp. 979–984.

Van der Hofstad, R., Janssen, A. J. E. M., van Leeuwaarden, J. S. H. & Stegehuis, C. (2017) Local clustering in scale-free networks with hidden variables. Phys. Rev. E, 95, 022307.

Wang, Z., Scaglione, A. & Thomas, R. J. (2009) On Modeling Random Topology Power Grids for Testing Decentralized Network Control Strategies. IFAC Proceedings Volumes, 42(20), 114 – 119, 1st IFAC Workshop on Estimation and Control of Networked Systems.

Watts, D. J. & Strogatz, S. H. (1998) Collective dynamics of ‘small-world’ networks. Nature, 393(6684), 440–442.

Winlaw, M., DeSterck, H. & Sanders, G. (2015) An In-Depth Analysis of the Chung-Lu Model. Discussion paper, Lawrence Livermore National Laboratory.

Xu, G. (2010) Controlled islanding algorithms and demonstrations on the WECC system. Arizona State University.
A Appendices

A.1 Chung-Lu model description

The Chung-Lu random graph model takes a desired degree sequence \( \mathbf{d} = (d_1, \ldots, d_n) \) as an input, where \( d_i \) is the desired degree of vertex \( i \). Note that this degree sequence also stipulates a desired number of edges, \( m = \frac{1}{2} \sum_i d_i \).

For every pair of vertices \( i, j \), the probability of an edge is \( \Pr(\{i, j\} \text{ is an edge}) = \frac{d_i d_j}{2m} \). In order to ensure that this probability is at most 1, one can further require that the input degree sequence \( \mathbf{d} \) satisfies \( \max_i d_i^2 < \sum_k d_k \). In expectation, each vertex achieves its desired degree \( d_i \), since

\[
E(\text{degree of vertex } i) = \sum_j \Pr(\{i, j\} \text{ is an edge}) = \sum_j \frac{d_i d_j}{2m} = d_i.
\]

For more information on the Chung-Lu model, see the monograph [12] by Chung. Instead of flipping a weighted coin for all \( \binom{n}{2} = O(n^2) \) possible edges, an efficient implementation of the Chung-Lu model favored by the authors of [19], known as “fast Chung-Lu”, instead chooses the endpoints of \( m = \frac{1}{2} \sum_i d_i \) edges by sampling these endpoints proportionally to their desired degree. More precisely, in fast Chung-Lu, \( \Pr(\text{selected}) = \frac{d_i}{2m} \), and thus, the expected degree of a vertex is again its desired degree, since here too

\[
\Pr(\{i, j\} \text{ is an edge}) = 2m \cdot \Pr(\text{selected}) \cdot \Pr(\text{selected}) = \frac{d_i d_j}{2m}.
\]

Since real world graphs are often large and sparse, with a number of edges \( m \) linear in the number of vertices \( n \), the fast Chung-Lu model may be preferred by practitioners; see also [22] for another efficient implementation of Chung-Lu. Note that in fast Chung-Lu, repeated edges or loops are possible since the endpoints of each edge are chosen independently. In practice, these tend to be few and are simply discarded in post-processing. For an in-depth comparison of the properties of the regular, fast, and other related versions of the Chung-Lu model, see [32].

A.2 Algorithms

In this appendix we provide pseudocode for the algorithms described in Section 4.

Algorithm 1 Preprocessing stage for subgraph of voltage \( X \). Input: Desired degrees \( \mathbf{d} = (d_1, \ldots, d_n) \) and desired diameter \( \delta \). Output: Updated degree sequence, \( \mathbf{d}' \), vertex-box sequence \( v = (v_1, \ldots, v_n) \), diameter path vertices \( D \), subdiameter path vertices \( S \).

1: procedure SETUP(\( \mathbf{d}, \delta \))
2: \( \eta \leftarrow |\{d \in \mathbf{d} : d > 0\}| \) \hspace{1cm} ⇒ Desired number of non-isolated vertices
3: \( \delta \leftarrow \text{round}(\delta - 2 \log \frac{n}{\delta+1}) \) \hspace{1cm} ⇒ Diameter adjustment

4: Inflated degree seq. until expected number of non-isolated vertices matches
5: while \( |\mathbf{d}| - \sum_{d_i \in \mathbf{d}} \exp(-d_i) \leq \eta \) do
6: \hspace{1cm} Randomly select nonzero \( d \in \mathbf{d} \) \hspace{1cm} ⇒ Nonisolated vertex to be duplicated
7: \hspace{1cm} \mathbf{d} \leftarrow (\mathbf{d}, d) \\
8: end while

9: Randomly distribute all non-isolated vertices into boxes
10: \( I_D \leftarrow \{i : d_i \geq 1\} \) \hspace{1cm} ⇒ Non-isolated vertex indices
11: \( B \leftarrow \{1, \ldots, \delta + 1\} \) \hspace{1cm} ⇒ Indices of \( \delta + 1 \) boxes

12: To be continued on next page...
In expectation, each box should have sufficiently many vertices so that any vertex can achieve the max desired degree 

\[ \text{if } \frac{n}{\delta + 1} < \max \langle \tilde{d} \rangle \text{ then} \]

Randomly select \( C \subseteq B \) with \( |C| = \frac{n}{\max \langle d \rangle} \). \( \triangleright \) If needed, make some boxes empty (except for diam path vertex)

\[ \text{else} \]

\[ C = B \]

\[ \text{end if} \]

\[ \triangleright \text{See ASSIGNBOXES procedure below} \]

Choose vertices for the diameter path and randomly (re)assign them to distinct boxes

\[ \text{if } \{|i : d_i \geq 3\} \geq \delta + 1 \text{ then} \]

\[ I_P = \{i : d_i \geq 3\} \]

\[ \triangleright \text{Choose these from vertices of degree at least 3, if possible} \]

\[ \text{else} \]

\[ I_P = \{i : d_i \geq 2\} \]

\[ \text{end if} \]

Randomly choose \( D \subseteq I_P \) with \( |D| = \delta + 1 \)

\[ \triangleright \text{Selects diameter path vertices from pool} \]

\[ \text{for each } i \in D \text{ do} \]

Randomly choose \( b \in B \)

\[ v_i \leftarrow b, B \leftarrow B \setminus b \]

\[ \triangleright \text{Assign each diameter path vertex to a distinct box} \]

\[ \text{end for} \]

Choose vertices for the subdiameter path and randomly (re)assign them to distinct boxes. Make subdiameter path as long as possible, up to diameter length

\[ \alpha \leftarrow \min(\delta + 1, |I_P \setminus D|) - 1 \]

\[ \triangleright \text{Length of subdiameter path} \]

\[ \beta \leftarrow \left\lfloor \frac{\delta + 1}{2} \right\rfloor - \left\lfloor \frac{\alpha - 1}{2} \right\rfloor \]

\[ \triangleright \text{Used to center subdiameter path below diameter path} \]

\[ B \leftarrow \{\beta, \ldots, \beta + \alpha\} \]

\[ \triangleright \text{Indices of boxes for the subdiameter path vertices} \]

\[ \text{Randomly choose } S \subseteq I_P \setminus D \text{ with } |S| = \alpha + 1 \]

\[ \triangleright \text{Select subdiameter path vertices from remaining pool} \]

\[ \text{for each } i \in S \text{ do} \]

Randomly choose \( b \in B \)

\[ v_i \leftarrow b, B \leftarrow B \setminus b \]

\[ \triangleright \text{Assign each sub-diameter path vertex to a distinct box} \]

\[ \text{end for} \]

\[ \text{return } (\tilde{d}, v, D, S) \]

\[ \text{end procedure} \]

\[ \text{procedure \ ASSIGNSBOXES}(\text{indices, boxes}) \]

\[ \text{for each } i \in \text{indices} \text{ do} \]

Randomly choose \( b \in \text{boxes} \)

\[ v_i \leftarrow b \]

\[ \text{end for} \]

\[ \text{return } v \]

\[ \text{end procedure} \]
Algorithm 2 Generative model for subgraph of voltage \( X \). \textbf{Input:} Updated degree sequence output from Algorithm 1 \( \tilde{d}' = (d_1, \ldots, d_n) \), vertex-box list \( \mathbf{v} = (v_1, \ldots, v_n) \), diameter path vertices \( D \), subdiameter path vertices \( S \). \textbf{Output:} edge list \( E \)

1: \textbf{procedure} CLC(\( \tilde{d}' \), \( \mathbf{v} \), \( D \), \( S \))

2: \( E \leftarrow \emptyset \)

3: \textbf{Make diameter path}
4: \textbf{for} \( k = 1, \ldots, |D| - 1 \) \textbf{do}
5: \hspace{1em} Find the (unique) vertices \( i, j \in D \) such that \( v_i = k \) and \( v_j = k + 1 \).
6: \hspace{1em} \( E \leftarrow E \cup \{i, j\} \)
7: \textbf{end for}

8: \textbf{Make subdiameter path}
9: \textbf{for} \( k = 1, \ldots, |S| - 1 \) \textbf{do}
10: \hspace{1em} Find the (unique) vertices \( i, j \in S \) such that \( v_i = k \) and \( v_j = k + 1 \).
11: \hspace{1em} \( E \leftarrow E \cup \{i, j\} \)
12: \textbf{end for}

13: \textbf{Create Chung-Lu graph on desired degrees of vertices in each box}
14: \textbf{for} \( k = 1, \ldots, \max (\mathbf{v}) \) \textbf{do}
15: \hspace{1em} \( B_k \leftarrow \{j : v_j = k\} \) \hspace{1em} \( \triangleright \) All vertices in box \( k \)
16: \hspace{1em} \( m_k \leftarrow \text{round} \left( \frac{1}{2} \sum_{i \in B_k} d_i \right) \) \hspace{1em} \( \triangleright \) Desired number of edges in Chung-Lu graph for box \( k \)
17: \hspace{1em} \textbf{for} \( c = 1, \ldots, m_k \) \textbf{do}
18: \hspace{2em} Randomly select \( i \in B_k \) proportional to \( d_i / 2m_k \)
19: \hspace{2em} Randomly select \( j \in B_k \) proportional to \( d_j / 2m_k \)
20: \hspace{2em} \( E \leftarrow E \cup \{i, j\} \) \hspace{1em} \( \triangleright \) Discard any loops or duplicate edges in post-processing
21: \hspace{1em} \textbf{end for}
22: \hspace{1em} \textbf{end for}

23: \textbf{return} \( E \)
24: \textbf{end procedure}

Algorithm 3 Insert transformer edges between subgraphs of voltage \( X \) and \( Y \). \textbf{Input:} Desired transformer degrees \( \tilde{t}[X, Y] = (\{t[X, Y]_1, \ldots, t[X, Y]_n\}, \{t[Y, X]_1, \ldots, t[Y, X]_n\}) \) and \( \tilde{t}[Y, X] = (\{t[Y, X]_1, \ldots, t[Y, X]_n\}). \textbf{Output:} edge list \( E \)

1: \textbf{procedure} STARS(\( \tilde{t}[X, Y] \), \( \tilde{t}[Y, X] \))

2: \( E \leftarrow \emptyset \), \( L^X \leftarrow \emptyset \), \( L^Y \leftarrow \emptyset \) \hspace{1em} \( \triangleright \) Edge list and “leftover bins”
3: \( I^X_0 \leftarrow \{i : t[X, Y]_i = 1\} \), \( I^X_1 \leftarrow \{i : t[Y, X]_i = 1\} \) \hspace{1em} \( \triangleright \) Degree 1 ver. in \( G[X] \), \( G[Y] \)
4: \( I^X_C \leftarrow \{i : t[X, Y]_i \geq 2\} \), \( I^Y_C \leftarrow \{i : t[Y, X]_i \geq 2\} \) \hspace{1em} \( \triangleright \) Degree at least 2 vertices in \( G[X] \) and \( G[Y] \)

5: \textit{To be continued on next page...}
Continued from the previous page...

Define procedure for \( k \)-stars centered at a vertex in \( G[A] \) with \( k \) leaves in \( G[B] \)

\[
\text{procedure } k\text{-STARS}(A, B) \\
\text{while } I^A_C \neq \emptyset \text{ do } \quad \triangleright \text{ If there are enough leaf vertices from } Y \text{ for create a star centered at } i \\
\begin{align*}
&\text{Randomly select } i \in I^A_C \\
&\text{if } |I^B_O| \geq t[A, B], i \text{ then } \\
&\quad \text{Randomly select } j \in I^B_S \\
&\quad E \leftarrow E \cup \{i, j\} \\
&\quad I^B_O \leftarrow I^B_O \setminus j \\
&\end{align*} \\
\text{end for} \\
\quad \triangleright \text{ Remove leaf vertex from pool} \\
I^A_C \leftarrow I^A_C \setminus i \\
\text{else} \\
\quad L^A \leftarrow i \\
\quad I^A_C \leftarrow I^A_C \setminus i \\
\end{align*}
\]
\[
\text{end if} \\
\text{end while}
\]
end procedure

Generate \( k \)-stars centered at vertices in both voltage \( X \) and \( Y \)
\[
k\text{-STARS}(X, Y) \\
k\text{-STARS}(Y, X)
\]

Insert 1-stars (edges) on any remaining degree 1 vertices from \( G[X] \) and \( G[Y] \)

\[
\text{while } I^X_O \neq \emptyset \text{ and } I^Y_O \neq \emptyset \text{ do } \\
\begin{align*}
&\text{Randomly select } i \in I^X_O \text{ and } j \in I^Y_O \\
& E \leftarrow E \cup \{i, j\} \\
& I^X_O \leftarrow I^X_O \setminus i, I^Y_O \leftarrow I^Y_O \setminus j \\
&\end{align*} \\
\text{end while}
\]
\[
L^X \leftarrow (L^X \cup I^X_O), \quad L^Y \leftarrow (L^Y \cup I^Y_O)
\]

Create bipartite Chung-Lu graph on any leftover vertices

\[
\text{if } L^X \neq \emptyset \text{ then } \\
\begin{align*}
&\quad m \leftarrow \sum_{i \in L^X} t[X, Y], i \\
&\quad \text{for } k = 1, \ldots, m \text{ do } \\
&\quad \quad \text{Randomly select } i \in L^X \text{ proportional to } t[X, Y], i/m \\
&\quad \quad \text{Randomly select } j \in L^Y \text{ proportional to } t[Y, X], j/m \\
&\quad \quad E \leftarrow E \cup \{i, j\} \\
&\quad \end{align*} \\
\text{end for}
\]
\[
\text{end if}
\]
\[
\text{return } E
\]
end procedure
Algorithm 4 Generative model for entire power grid graph on $k$ voltage levels, $X_1, \ldots, X_k$. Input: Desired same-voltage degrees $\vec{d}^{X_1}, \ldots, \vec{d}^{X_k}$, same-voltage diameters $\vec{\delta}^{X_1}, \ldots, \vec{\delta}^{X_k}$ and transformer degrees $\vec{t}[X_i, X_j]$ for each pair $i, j \in \{1, \ldots, k\}$. Output: edge list $E$

1: procedure CLCSTARS($\{\vec{d}^{X_i}\}, \{\vec{\delta}^{X_i}\}, \{\vec{t}[X_i, X_j]\}$)
2: $E \leftarrow \emptyset$
3: Create each same-voltage subgraph
4: for $i = 1, \ldots, k$ do
5: $(\vec{d}', v, D, S) \leftarrow \text{SETUP}(\vec{d}^{X_i}, \vec{\delta}^{X_i})$
6: $E \leftarrow E \cup \text{CLC}(\vec{d}', v, D, S)$
7: end for
8: Insert transformer edges between each pair of same-voltage subgraphs
9: for $i = 1, \ldots, k$ do
10: for $j = i + 1, \ldots, k$ do
11: $E \leftarrow E \cup \text{STARS}(\vec{t}[X_i, X_j], \vec{t}[X_j, X_i])$
12: end for
13: end for
14: return $E$
15: end procedure

A.3 Remarks on algorithms

Before discussing the algorithms constituting our generative model, we note our model is general in the sense that neither Phase 1 nor 2 require additional assumptions on its inputs except that they are valid and simultaneously realizable by some graph. In the case of Phase 1, this means there must be sufficiently many vertices of degree at least 2 in order to generate the diameter path of desired length. In the case of Phase 2, since the inputs $\vec{t}[X, Y]$ and $\vec{t}[Y, X]$ are in fact desired degree sequences associated with partitions of a bipartite graph, these sequences must sum to the same value. In order to achieve this generality, Phase 1 and 2 contain adjustments that will only be made for extreme cases which appear to be rare or non-existent in our tested power-grid graph data, but nonetheless possible in a graph. Furthermore, the setup of Phase 1 also makes some subtle adjustments to better ensure that the isolated vertex count, diameter, and maximum degree stipulated by the model inputs are more accurately achieved in the output of the CLC model. While none of these adjustments are central to the idea of the model, we explain and justify them below.

Algorithm 1

- Line 4 Adjusting diameter. While the diameter path created by the CLC model will have desired diameter length $\delta$, the diameter of the entire CLC graph may still be slightly larger. This is because the creation of random Chung-Lu graphs within each box are likely to extend the diameter path in the first and last boxes. Thus, we expect that the additional length added to the diameter path won’t exceed twice the expected diameter for each Chung-Lu graph. We estimate the diameter of each Chung-Lu graph by $\log (\mathbb{E} (\# \text{ vertices in each box})).$

- Lines 5-7 Adjusting the expected number of isolated vertices. In a realization of the Chung-Lu model, not only will vertices with desired degree 0 be isolated with probability 1, but vertices with nonzero desired degree can also become isolated with nonzero probability. Thus, the number of isolated vertices in an instance of the Chung-Lu model will exceed the number of vertices desiring degree 0. In order to adjust for this, we slightly inflate the provided degree sequence by randomly duplicating vertices until the expected number of isolated vertices matches the number of vertices desiring degree 0. Here, we use a result from [12], which states that the expected number of isolated vertices for a Chung-Lu graph with degree sequence $d = (d_1, \ldots, d_n)$ is

$$\mathbb{E}(\# \text{ isolated vertices}) = \sum_i \exp(-d_i) + O \left( \frac{\sum_i d_i^2}{\sum_i d_i} \right) \approx \sum_i \exp(-d_i).$$

- Lines 8-10 Allowing maximum degree to be achieved in any box. In the CLC model, the expected number of vertices in each box, and thus in each Chung-Lu random graph, is $\frac{\# \text{ total vertices}}{\# \text{ boxes}}$. If the desired degree
of a vertex exceeds the number of vertices in its box, that vertex will only be able to achieve a strictly smaller degree.\footnote{Additionally, the desired degree sequence within each particular box still may not be achievable (e.g. the sum of the desired degrees within each box may not be even). Nonetheless, the Chung-Lu model will still output a random graph with (if not an exact match) an approximation to this degree sequence; for practical purposes, this has not posed a problem for matching the overall desired degree sequence reasonably well.} We note that this scenario is extremely uncommon in our power grid data. Nonetheless, in such cases, the CLC adjusts by randomly choosing a number of boxes to be empty and allocating vertices in the other boxes, so that the expected number of vertices in each non-empty box is at least the maximum desired degree.

Algorithm 3

- Lines 36-40 **Sufficient condition for perfect transformer degree match.** These lines describe the fast bipartite Chung-Lu model introduced in 1. We apply this model only to any “leftover” vertices which could not be allocated to create stars by the preceding procedure. We note that a sufficient condition for all vertices of voltage \(X\) and \(Y\) to be allocated to stars (in which case they are guaranteed to exactly achieve their desired degree) is that there are sufficiently many vertices of voltage \(Y\) with transformer degree 1 so that every vertex in \(X\) with transformer degree 2 or more can achieve its degree by being linked to these degree-1 vertices from \(Y\), and vice versa. Loosely speaking, this condition is likely to be satisfied when the transformer edges graph consists mainly of disjoint edges. Stated formally, this condition is that the following inequalities hold

\[
\sum_{i : t[X,Y]_i \geq 2} t[X,Y]_i \leq |\{j \in Y : t[Y,X]_j = 1\}|, \quad \sum_{i : t[Y,X]_i \geq 2} t[Y,X]_i \leq |\{j \in X : t[X,Y]_j = 1\}|.
\]

If the above condition is satisfied – which is the case in our power grid graph data – the leftover bins \(L^X\) and \(L^Y\) defined in Algorithm 3 will be empty and consequently lines 36-40 will never be executed by the model.

A.4 Assortativity and spectral gap

Below, we present supplemental results on our models performance with regard to assortative mixing and spectral gap for the Eastern Interconnection. Bar plots comparing the real data to the graphs produced by our model are shown in Figure 13. As before, the bar heights are the average over 100 trials, and the error whiskers represent the minimum and maximum values observed. These figures show that our CLC and CLC+Stars models achieve a closer match than the CL model in all cases. The assortativity coefficient \(\tau\) of a graph is the Pearson correlation coefficient of degree between pairs of adjacent vertices. We note that \(-1 \leq \tau \leq 1\), with small values of \(\tau\) suggesting that high-degree vertices tend to link to low-degree vertices, while large values of \(\tau\) suggest vertices tend to link with vertices of similar degree. The spectral gap of a graph is the second smallest eigenvalue of the Laplacian matrix of the graph. This eigenvalue is a numerical measure of the graphs connectivity, with smaller values suggesting a less-robust connectivity structure. The second Laplacian eigenvalue also corresponds to the Fiedler eigenvector \(\lambda_2\) used in graph partitioning problems. In order to facilitate more meaningful comparisons between networks of varying sizes, we consider the second smallest eigenvalue \(\lambda_1\) of the normalized Laplacian matrix popularized by Chung [10]. This quantity is intimately related to the Cheeger constant (also called isoperimetric number or conductance) of the graph, which measures the graphs “bottleneckedness” or sparsest cut. We note that \(\lambda_1 > 0\) if and only if the graph is connected; thus, we restrict attention to the largest connected components.

\[\text{Figure 13: Bar charts showing comparison for assortativity and spectral gap.}\]