Generating Connected, Simple, and Realistic Cyber Graphs for Smart Grids

Osman Boyaci
Electrical Engineering
Texas A&M University
College Station, TX, 77843
osman.boyaci@tamu.edu

M. Rasoul Narimani
College of Engineering
Arkansas State University
Jonesboro, AR, 72404
mnarimani@astate.edu

Katherine Davis
Electrical Engineering
Texas A&M University
College Station, TX, 77843
katedavis@tamu.edu

Erchin Serpedin
Electrical Engineering
Texas A&M University
College Station, TX, 77843
eserpedin@tamu.edu

***Abstract*—Smart grids integrate communication systems with power networks to enable power grids operation and command through real-time data collection and control signals. Designing, analyzing, and simulating smart grid infrastructures as well as predicting the impact of power network failures strongly depend on the topologies of the underlying power network and communication system. Despite the substantial impact that the communication systems bring to smart grid operation, the topology of communication systems employed in smart grids was less studied. The power community lacks realistic generative communication system models that can be calibrated to match real-world data. To address this issue, this paper proposes a framework to generate the underlying topological graphs for the communication systems deployed in smart grids by mimicking the topology of real-world smart grids. In this regard, we have updated the Chung-Lu algorithm to guarantee the communication network connectivity and to match the degree distribution of a real-world smart grid rather than following an expected degree distribution. In addition, key characteristics of communication systems such as diameter, average shortest paths, clustering coefficients, assortativity, and spectral gap were taken into consideration to generate the most similar real-world communication network for smart grid studies. We believe that the proposed algorithm to generate realistic cyber graphs for smart grid studies will benefit the power community.**

NOMENCLATURE

$\mathcal{G}, \mathcal{V}, \mathcal{E}$	Graph, set of nodes, set of edges
$n,\ m \in \mathbb{Z}$	$\lvert\mathcal{V}\rvert, \lvert\mathcal{E}\rvert$
$u.d \in \mathbb{R}$	Degree of a node u
$S \in \mathbb{Z}^n$	Degree sequence
$\overline{d} \in \mathbb{Z}$	Max. degree
$\widehat{(\cdot)}$	Normalization operation s.t. $\sum_1^n \widehat{(\cdot)} = 1$
$D \in \mathbb{Z}^{\overline{d}}$	Degree vector $D = [1, \ldots, \overline{d}]$
$\widehat{K} \in \mathbb{R}^{\overline{d}}$	Normalized frequency vector
$\mathcal{G}.g \in \{0, 1\}$	$\mathcal{G}$ is graphical (no self loops and parallel edges)
$\mathcal{G}.c \in \{0, 1\}$	$\mathcal{G}$ is connected
$\rho \in \mathbb{R}$	Density as m/n
$\varnothing \in \mathbb{R}$	Diameter as the longest shortest path
$\widehat{sp} \in \mathbb{R}$	Average shortest path
$cc \in \mathbb{R}$	Clustering coefficient
$a \in \mathbb{R}$	Assortativity as the Pearson correlation coefficient of degrees between pairs of linked nodes
$\lambda \in \mathbb{R}$	Spectral gap as the minimum non-zero eigenvalue of the normalized Laplacian

This work was supported by NSF under Award Number 1808064.

I. INTRODUCTION

Communication systems play a major role in the deployment of smart grids empowering them to be more resistant, secure, reliable and manageable and ensuring connectivity of the grid components. The backbone of communication systems in smart grids is represented by the information and communication technologies that allow two-way communication and automated control. Communication systems improve the efficiency and reliability of smart grids by gathering and transmitting a wide variety of data for grid control and decision-making purposes. The integration of cyber communications and control systems into the power distribution infrastructure has a profound impact on the operation, reliability, and efficiency of the power grid. The power and communication systems in modern power grids are highly intertwined. Analyzing, simulating, designing, and predicting the impact of network failures strongly rely on the knowledge of a communication network topology [1]. Thus, studying the underlying communication network topology is essential for the smart grid operation and control [2].

In spite of the many models proposed for electrical power systems [3], the problem of modeling the underlying communication network in smart grids was less studied. In fact, despite the huge efforts deployed for studying smart grids operation and control, modeling smart grids is still at its infancy. There is not enough realistic and practical information about the topology of the underlying communication network in smart grids. So far, various efforts have focused on developing cyber-physical test models for general use by the power system community [4]–[7]. These studies consider different characteristics of communication systems including vulnerabilities of communication devices, attack paths, etc., to design a practical cyber layer for cyber-physical power systems. However, taking all these characteristics into consideration makes these approaches computationally intractable for larger cyber graphs as the number of attack paths increases exponentially with the number of nodes.

To analyze the impact of cyber-graphs on power networks operation, e.g., cascading failure analysis, we first need a fast and reliable framework to generate realistic cyber graphs for power test cases irrespective of their size. A few efforts were conducted for generating realistic cyber graphs for power test cases. A graph generator based on the characteristics of Luxembourg smart grid, which is a power-line communication (PLC) system [8], was presented in [1] to create random

but realistic smart grid communication topologies. Different characteristics of power grid including nodal degree distribution, graph spectrum and connectivity scaling property, etc., were analyzed in [9] for designing efficient communication schemes for power test cases. Heuristic algorithms were employed in [10] to improve the communication reliability for smart grids at the transmission level. However, many generic graph generation algorithms such as configuration model [11], Havel-Hakimi algorithm [12], and Chung-Lu algorithm do not guarantee the graphical and connected graphs properties, and thus are not fit for designing communication systems for smart grids. Horvát-Modes model, on the contrary, yields both connected and graphical outputs. Due to its edge connection mechanism it produces graphs with large diameter and low assortativity, which are not realistic for communication systems. Thus, it is important to propose generative graph algorithms that take into account the real-world characteristics of communication systems in the design process.

Power system statistics were leveraged to design optimal communication systems for smart grids in [9]. Similarly, the communication system statistics can be leveraged for designing a realistic communication system for smart grids which is the centerpiece of this paper. Along this parallel, we first derive the statistical metrics of a real-world smart grid's communication system and then propose a graph generator based on the statistical information of the smart grid's communication system. Different graph attributes of a real-world smart grid communication graph such as diameter, assortativity, etc., are taken into consideration in designing the communication system for power test cases. Moreover, we adapt the Chung-Lu algorithm to preserve the connectivity of the graph since the connectivity is a key characteristic of communication systems. In addition, edge switching operation is employed to prevent self loops and parallel paths in the communication graph.

The contributions of this work are outlined as follows: (1) To generate connected, simple, and realistic cyber graphs for smart grids, we propose a simple and elegant framework by updating the Chung-Lu algorithm. (2) To satisfy the required degree distribution, we propose an adaptive remaining degree approach instead of the fixed expected degrees. (3) To minimize the length of cross edges between power and cyber graphs, we employ the Hungarian algorithm for optimal matching between cyber and power nodes. (4) To compare the proposed method with the currently available approaches, we implement other graph generation methods in the literature such as configuration model, Havel-Hakimi, Horvát-Modes, and Chung-Lu algorithms using the same degree sequence and analyze the global characteristics of the output graphs.

The remainder of this paper is divided into four sections. Section II proposes the cyber graph generation framework. Section III presents the generated graphs and discusses their global characteristics. Finally, Section IV concludes the paper.

II. Cyber-Graph Generation

A. Analyzing a real-world communication system

To be able to generate realistic cyber graphs, we first analyze a real communication system of a smart grid given in [2, Table (3.2)]. Its degree count vector K^* is extracted as $K^* = [162, 101, 30, 25, 11, 4, 3, 4, 2, 1]$ and its global graph characteristics are tabulated in Table I. It is clear from the

Table I
Global graph characteristics of the reference system [2, Table (3.1)]

$\|V\|$	$\|E\|$	ρ	$\varnothing$	$\widetilde{sp}$	cc	a	λ
343	357	1.04	28	11.47	0.05	-0.22	$1.16e^{-3}$

distribution of K^* that the reference graph shows scale free topology since K^* values tend to diminish for larger degrees ($K^*_1 = 362 \gg K^*_{10} = 1$). In addition, its global characteristics indicate that it is a sparse ($\rho = 1.04$) and tree like graph (relatively high $\varnothing$ and $\widetilde{sp}$). Moreover, it has only a few cyclic structures ($cc = 0.05$), tends to connect higher degree nodes with a lower degree ones ($a = -0.22$) and presents some bottleneck edges ($\lambda = 1.16e^{-3}$) that removal of them may split the graph into different components [2].

B. Selecting the degree distribution function

To select an appropriate degree distribution function that can generate the required degree distribution, we consider and optimize the parameters of three main distribution: generalized lognormal distribution [13], power law distribution [14] and zipf distribution [14]. Specifically, we first obtain the frequency vector $\widehat{K^*}$ of the reference system and optimize the parameters θ of each distribution function f by:

$$\min_{\theta} ||\widehat{K^f} - \widehat{K^*}||_2, \tag{1}$$

where $\widehat{K^*}$ is the normalized frequency vector of the reference system and $\widehat{K_f}$ denotes the frequency vector of K^f with $K^f_d = f_\theta(d)$. As can be seen from Fig. 1 while powerlaw and zipf distributions underestimate $\widehat{K^*}$ when $D < 4$ and overestimates it when $D > 4$, lognormal distribution better approximates the reference distribution for all D values. Formulation of the distributions, their optimal parameters, and mean square errors when estimating $\widehat{K^*}$ is given in Table II. Note that since the lognormal distribution provides the best approximation, we use the lognormal distribution $f^*(x) = f_{\alpha=1.371,\beta=1.986}(x)$ to generate the degree sequences for the rest of the paper.

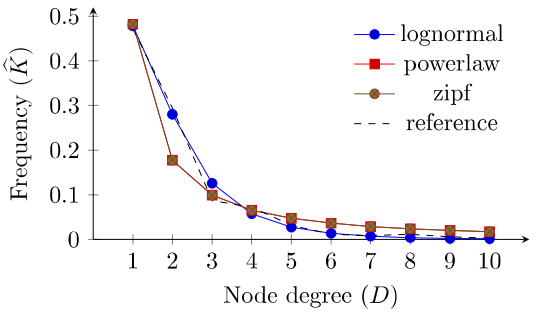

Figure 1. Normalized frequencies of the optimized degree distribution functions to approximate the reference distribution

Table II
OPTIMIZED DEGREE DISTRIBUTION FUNCTIONS

distribution	formula	α	β	MSE
lognormal	$f_{\alpha,\beta}(x) = \exp\left(-\left(\frac{log(x)}{\alpha}\right)^{\beta}\right)$	1.371	1.986	$2e^{-4}$
powerlaw	$g_{\alpha,\beta}(x) = \beta x^{-\alpha}$	1.440	3.745	$1.6e^{-3}$
zipf	$h_{\alpha}(x) = x^{-\alpha}/\zeta(\alpha)$	1.440		$1.6e^{-3}$

C. Generating valid degree sequence

The first step for generating a valid degree sequence for a realistic cyber graph is to specify the number of nodes, n, number of edges, m, and maximum degree for each node, $\overline{d}$, [2]. Then, we can generate the degree sequence S by randomly drawing n samples from the degree vector $D = [1, \ldots, \overline{d}]$ with the corresponding probabilities $P(d = \widehat{F}_d)$ where $F = [f^*(1), \ldots, f^*(\overline{d})]$. Note that not every S is realizable since it should satisfy $\sum_{i=1}^{n} S_i = 2m$ for an undirected graph. In addition, to be able to generate a simple graph without any self-loops or parallel edges, S should be graphical. For instance, $S1 = [1, 1, 4]$ and $S2 = [2, 2]$ are not graphical due to a self-loop and a parallel edge, respectively. To test whether a given degree sequence S is graphical or not, we utilize the well-known Havel Hakimi algorithm [12] given in Algorithm 1. If S is not graphical, a new degree sequence should be generated.

Algorithm 1: Check if the degree sequence is graphical

```
1  function is_simple_graph(S)
2      while True do
3          sort S ascending
4          n ← |S|
5          if S_1 = 0 and S_n = 0 then return True
6          v ← popleft(S)
7          if v > n − 1 then return False
8          for i ← 1 to v do
9              decrement S_i if S_i < 0 then return
                False
10         end
11     end
```

D. Generating simple connected cyber graph

The main algorithm outlined in Algorithm 2, for generating a simple graph $\mathcal{G}$ from a degree sequence S, presents five basic steps. The first phase in Algorithm 2 is the initialization phase (lines 1 to 9) in which n and m are determined, nodes are created with their required degrees, visited and unvisited sets I and J are created, and $\mathcal{G}$ which includes the maximum degree node is created. The second phase is the tree generation, and third phase is adding the remaining degrees. The fourth and fifth phases are removing the sell loops and parallel edges in order to make $\mathcal{G}$ simple.

Dynamic adaptive sampling outlined in Algorithm 3 is the backbone of the proposed algorithm to satisfy the required degree sequence. It draws a sample v from the input set I by a probability proportional to its nodes' remaining degrees and decrements the degree of v for further samplings.

Algorithm 4 outlines the tree generation algorithm in which at each iteration a random node pair sampled from the visited set I and the unvisited set J are connected. Sampling random nodes from I and J guarantees the connectivity of the output graph $\mathcal{G}$ in which any two nodes are connected by exactly one path, which brings the tree property.

Algorithm 2: Generate a simple connected graph (main)

```
1  function main(S)
2      n, m ← |S|, sum(S)/2
3      create nodes v_1, v_2, ..., v_n s.t. v_i.d ← S[i]
4      I, J ← { }, {v_1, v_2, ..., v_n}
5      create G s.t. G.V = { }, G.E = [ ]
6      v_max ← max. degree node
7      add v_max to G.V as the first node
8      remove v_max from J and add it to I
9      generate_tree(G, I, J, n)
10     E_s, E_p ← add_remaining_edges(G, I, n, m)
11     remove_self_loops(G, E_s)
12     remove_parallel_edges(G, E_p)
13     return G
```

Algorithm 3: Draw a sample from a set

```
1  function sample(I)
2      draw random v from set I w.r.t. node degrees
3      decrement v.d
4      return v
```

Algorithm 4: Generate the tree.

```
1  function generate_tree(G, I, J, n)
2      for i ← 1 to n − 1 do
3          u ← sample(I)
4          v ← sample(J)
5          remove v from J and add it to I
6          add v to G.V
7          add (u, v) to G.E
8      end
```

If there is any remaining edge to be added to $\mathcal{G}$, in other words, if there is any positive degree node pairs in $\mathcal{G}$, it is simply added between these two randomly sampled nodes as summarized in Algorithm 5. Note that self loops and parallel edges are saved in E_s and E_p for later removal.

Algorithm 5: Add remaining edges.

```
1  function add_remaining_edges(G, I, n, m)
2      E_s, E_p ← [ ], [ ]
3      for i ← n to m do
4          u ← sample(I)
5          v ← sample(I)
6          if u = v then append u to E_s
7          else if (u, v) ∈ G.E then append (u, v) to E_p
8          add (u, v) to G.E
9      end
10     return E_s, E_p
```

To remove the self-loops, we implement the edge switching strategy in Algorithm 6. Assume that node u has a self loop and other nodes s and t present an edge. We first remove the edges (u, u) and (s, t) and then add edges (u, s) and (u, t) to remove the self loop at u. Note that remaining node degrees do not change after this operation as indicated by the numbers

beneath each node in Fig. 2 which show the number of edges that need to be added to each node.

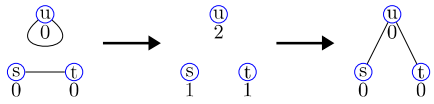

Figure 2. Edge switching operations to remove a self loop

Algorithm 6: Remove self-loop by edge switching

```
function remove_self_loops(G, E_s)
    while |E_s| > 0 do
        draw sample u ∈ E_s
        draw sample (s, t) ∈ G
        if |{u, s, t}| = 3 then
            if (u, s) ∉ G.E and (u, t) ∉ G.E then
                remove u from E_s
                remove (u, u) and (s, t) from G.E
                add (u, s) and (u, t) to G.E
            end
        end
    end
```

Parallel edges are removed similarly as shown in Algorithm 7. Assume $\mathcal{G}$ has parallel edges (u, v) and other nodes s and t present an edge. To remove one of the parallel edges between u and v, we first remove one of them, i.e., (u, v) and (s, t). Next, we add edges (u, s) and (v, t) to the graph. The process of eliminating parallel edges is illustrated in Fig. 2 where the required node degrees are shown by the numbers beneath each node.

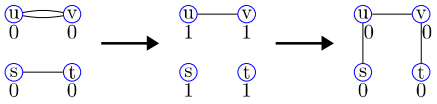

Figure 3. Edge switching operations to remove a parallel edge

Algorithm 7: Remove parallel edges by edge switching

```
function remove_parallel_edges(G, E_p)
    while |E_p| > 0 do
        sample (x, y) ∈ G
        if |{u, v, x, y}| = 4 then
            if (u, x) ∉ G.E and (v, y) ∉ G.E then
                remove (u, v) from E_p
                remove (u, v) and (x, y) from G.E
                add (u, x) and (v, y) to G.E
            end
            else if (u, y) ∉ G.E and (v, x) ∉ G.E
             then
                remove (u, v) from E_p
                remove (u, v) and (x, y) from G.E
                add (u, y) and (v, x) to G.E
            end
        end
    end
```

E. Relabeling the output graph's nodes

Since the proposed algorithm randomly generates the cyber graph, the nodes' position can be imperfect to match their corresponding nodes in the power graph. For instance, a cyber node can be placed far away from the power node that it controls as demonstrated by the cyber node 6 given in Fig.4a and the power node 6 given in Fig.4b. To address this issue,

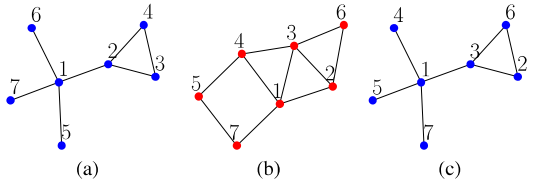

Figure 4. Relabeling the generated cyber graph to minimize the cross edge distances. (a) Generated cyber graph. (b) Given power graph. (c) The cyber graph after relabeling. After relabelling, the distance between a power node in (b) and its corresponding cyber node in (c) with the same label is minimized.

we keep the node positions fixed and relabel the cyber graph labels to minimize the cross links' distances. Specifically, we formalize this as an optimization problem (2) based on the Hungarian algorithm [15] to find the optimal matching between power and cyber nodes:

$$\min_{L,R}, \mathrm{Tr}(LCR) \tag{2}$$

where L and $R \in \{0, 1\}^{n\times n}$ are the permutation matrices and $C \in R^{n\times n}$ is the cost matrix whose elements are defined by Euclidean distances $C_{u,v} = ||u - v||_2$ for each u, v pairs from power and cyber graphs, respectively. As shown in Figure 4 for a 7-bus test case, the node "4" in the generated cyber graph matches with power node "6" which is not in its vicinity before applying the relabeling process. After applying the relabeling process, the node "6" in the generated cyber graph matches with node "6" in power graph that minimizes crossing edge distance between two graphs.

III. Results and Discussion

For a fair comparison, we implemented the existing approaches in the literature such as configuration model (CM), Havel-Hakimi (HH), Chung-Lu (CL), and Horvát-Modes (HM) algorithms. Then, we generate random graphs for 30-, 118-, and 300-buses IEEE test systems using the same degree sequence. Generated graphs are visualized in Fig. 5 where nodes' color and sizes indicate their degrees. In addition, the global characteristics of these graphs are tabulated in Table III.

As can be seen from the Fig. 5 and Table III, although these graphs are generated by the same degree distribution, their configurations are totally different. The graph generated by "Configuration Model" is neither graphical, nor connected. Havel-Hakimi model, in contrast, generates graphical outputs. However, this model does not guarantee connectivity of the graph and also it has a high clustering coefficient ($cc > 0.3$) and assortativity ($a > 0.7$). These features makes Havel-Hakimi model ineligible for generating realistic cyber graphs. Similar to the Havel-Hakimi model, Chung-Lu model generates graphical but not connected graphs. However, since it utilizes the expected degree distribution instead of the exact one, its generated graphs may show different characteristics from the graph that generated by the exact degree distribution.

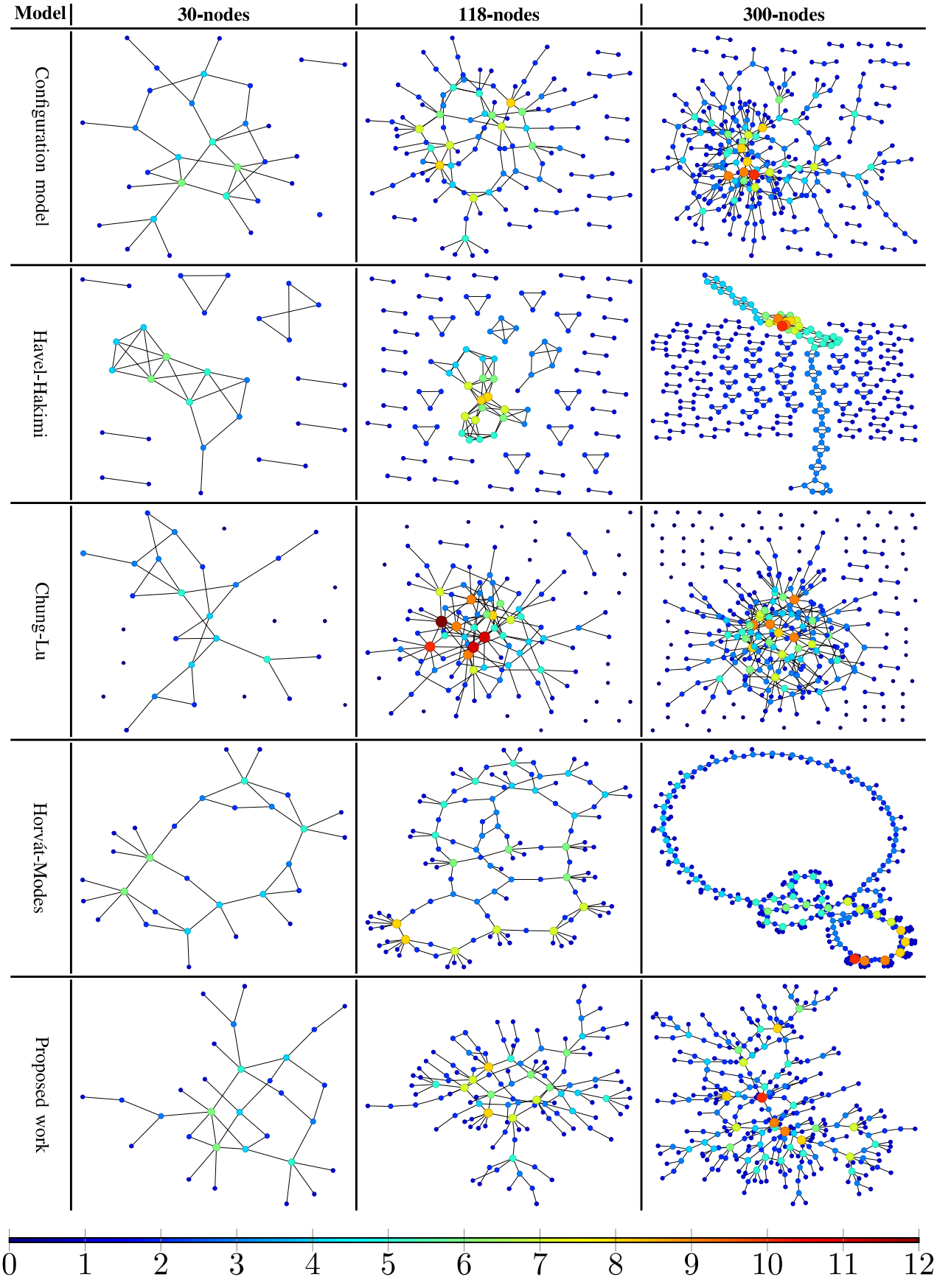

Figure 5. Generated graphs for each model (rows) and system (columns). Node's colors and sizes are given according to their degrees for better visualization

Table III
GLOBAL GRAPH CHARACTERISTICS FOR EACH MODEL AND SYSTEM (CM:CONFIG. MODEL, HH: HAVEL-HAKIMI, CL:CHUNG-LU HM:HORVÁT-MODES, PW:PROPOSED WORK

n	model	$\mathcal{G}.g$	$\mathcal{G}.c$	ρ	$\varnothing$	$\widetilde{sp}$	cc	a	λ
30	CM	✗	✗	1.167	–	–	–	-0.128	$9.62e^{-2}$
	HH	✓	✗	1.167	–	–	0.303	0.776	$2.19e^{-1}$
	CL	✓	✗	0.933	–	–	0.098	-0.132	$9.80e^{-2}$
	HM	✓	✓	1.167	8	3.84	0.050	-0.676	$4.74e^{-2}$
	PW	✓	✓	1.167	7	3.44	0.048	-0.220	$8.62e^{-2}$
118	CM	✗	✗	1.102	–	–	–	-0.010	$3.13e^{-2}$
	HH	✓	✗	1.102	–	–	0.417	0.880	$1.22e^{-1}$
	CL	✓	✗	1.220	–	–	0.058	-0.002	$8.86e^{-2}$
	HM	✓	✓	1.102	17	7.88	0.000	-0.737	$7.68e^{-3}$
	PW	✓	✓	1.102	13	5.64	0.026	-0.220	$1.68e^{-2}$
300	CM	✗	✗	1.040	–	–	–	-0.059	$7.77e^{-3}$
	HH	✓	✗	1.040	–	–	0.365	0.927	$1.79e^{-3}$
	CL	✓	✗	0.943	–	–	0.008	-0.029	$4.50e^{-2}$
	HM	✓	✓	1.040	54	21.22	0.000	-0.606	$7.62e^{-4}$
	PW	✓	✓	1.040	22	9.22	0.007	-0.226	$2.86e^{-3}$

For instance, the graphs generated by the Chung-Lu model for IEEE 30- and 300-bus test systems present less edges ($\rho < 0.95$) than the required edges. Conversely, the graph generated by the Chung-Lu model for the 118-bus test system has more edges ($\rho > 1.2$) compared to the required one. Horvát-Modes model, contrary to the previously mentioned models, generates both graphical and connected outputs. Yet, its highly low assortativity ($a < -0.6$) is not compatible with the real-world cyber graph. In addition, the graph generated for 300-bus test system presents high diameter ($\varnothing = 54$) and high average shortest path ($\widetilde{sp} > 21$) which indicate its "bias" induced to preserve the connectivity. In contrast, our proposed method generates tree-like connected structures that have the exact degree distribution and similar graph attributes to the reference system. Moreover, it generates a graph which has a diameter and an average shortest paths proportional to its node size, an appropriate clustering coefficient, a spectral gap decreasing with its node size, and almost the same assortativity ($a = -0.22$) with the reference system.

Another finding of our experiments is that although the graphs generated by Havel-Hakimi, Horvát-Modes, and the proposed work exhibit the same distribution, they produce totally different topologies, especially for the 300-bus test system in Fig. 5. For instance, Havel-Hakimi produces many 2-vertex or 3-vertex disconnected cliques. Horvát-Modes, on the contrary, tends to create graphs with higher assortativity to preserve the connectivity. In contrast, the proposed model generates tree like structures to better model a real-world cyber graph. Aside from the global graph characteristics given in Table III, this situation corroborates the supremacy of the proposed algorithm over the existing ones.

IV. CONCLUSION

In this paper, we propose a simple and elegant framework by modifying the Chung-Lu algorithm to generate connected, simple, and realistic cyber graphs for power systems. For generating graphs which have the exact required node degree distribution, we propose an adaptive remaining degree approach instead of the fixed expected degree in the Chung-Lu method. In addition, we utilize the Hungarian algorithm to minimize the length of cross-links between the power graph and the generated cyber graph. We implement other graph generation methods to compare the suitability of the proposed algorithm with those in literature including configuration model, Havel-Hakimi, Chung-Lu, and Horvát-Modes algorithms and generate cyber graphs using the same degree sequence for each model. Generated cyber graphs for IEEE 30-, 118-, and 300-bus test systems demonstrate that the proposed model yields better results compared to the existing approaches in terms of global characteristics, connectivity, and graphicality to mimic a real-world communication system. The proposed framework can be utilized by power system community to generate realistic cyber graphs for cyber-physical power system studies.

REFERENCES

[1] Z. Wang, A. Scaglione, and R. J. Thomas, "Generating statistically correct random topologies for testing smart grid communication and control networks," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 28–39, 2010.

[2] X. Fan, D. Wang, S. Aksoy, A. Tbaileh, Q. H. andT. Fu, and J. Ogle, "Coordination of transmission, distribution and communication systems for prompt power system recovery after disasters," *PNNL-28598*, 2019.

[3] IEEE PES Task Force on Benchmarks for Validation of Emerging Power System Algorithms, "The Power Grid Library for Benchmarking AC Optimal Power Flow Algorithms," *arXiv:1908.02788*, Aug. 2019. [Online]. Available: https://github.com/power-grid-lib/pglib-opf

[4] K. R. Davis, C. M. Davis, S. A. Zonouz, R. B. Bobba, R. Berthier, L. Garcia, and P. W. Sauer, "A cyber-physical modeling and assessment framework for power grid infrastructures," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2464–2475, 2015.

[5] G. A. Weaver, K. Davis, C. M. Davis, E. J. Rogers, R. B. Bobba, S. Zonouz, R. Berthier, P. W. Sauer, and D. M. Nicol, "Cyber-physical models for power grid security analysis: 8-substation case," in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2016, pp. 140–146.

[6] P. Wlazlo, K. Price, C. Veloz, A. Sahu, H. Huang, A. Goulart, K. Davis, and S. Zounouz, "A cyber topology model for the texas 2000 synthetic electric power grid," in *2019 Principles, Systems and Applications of IP Telecommunications (IPTComm)*, 2019, pp. 1–8.

[7] A. Sahu, H. Huang, K. Davis, and S. Zonouz, "A framework for cyber-physical model creation and evaluation," in *2019 20th International Conference on Intelligent System Application to Power Systems (ISAP)*, 2019, pp. 1–8.

[8] S. Galli, A. Scaglione, and Z. Wang, "For the grid and through the grid: The role of power line communications in the smart grid," *Proceedings of the IEEE*, vol. 99, no. 6, pp. 998–1027, 2011.

[9] Z. Wang, A. Scaglione, and R. J. Thomas, "Generating statistically correct random topologies for testing smart grid communication and control networks," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 28–39, 2010.

[10] K. Velin, L. Martin, T. David, and G. Teresa, "Reliable communication networks for smart grid transmission systems," *Journal of Network and Systems Management*, vol. 24, pp. 629–652, 2016.

[11] M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[12] S. L. Hakimi, "On realizability of a set of integers as degrees of the vertices of a linear graph. i," *Journal of the Society for Industrial and Applied Mathematics*, vol. 10, no. 3, pp. 496–506, 1962.

[13] T. G. Kolda, A. Pinar, T. Plantenga, and C. Seshadhri, "A scalable generative graph model with community structure," *SIAM Journal on Scientific Computing*, vol. 36, no. 5, pp. C424–C452, 2014.

[14] M. Cristelli, M. Batty, and L. Pietronero, "There is more than a power law in zipf," *Scientific reports*, vol. 2, no. 1, pp. 1–7, 2012.

[15] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.