

## Spanning trees of recursive scale-free graphs

C. Tyler Diggans<sup>1,2,3</sup>, Erik M. Bollt<sup>1,4</sup> and Daniel ben-Avraham<sup>1,2</sup><sup>1</sup>Clarkson Center for Complex Systems Science, Clarkson University, Potsdam, New York 13699, USA<sup>2</sup>Department of Physics, Clarkson University, Potsdam, New York 13699, USA<sup>3</sup>Air Force Research Laboratory: Information Directorate, Rome, New York 13441, USA<sup>4</sup>Department of Electrical and Computer Engineering, Clarkson University, Potsdam, New York 13699, USA

(Received 14 December 2021; accepted 9 February 2022; published 22 February 2022)

We present a link-by-link rule-based method for constructing all members of the ensemble of spanning trees for any recursively generated, finitely articulated graph, such as the Dorogovtsev-Goltsev-Mendes (DGM) net. The recursions allow for many large-scale properties of the ensemble of spanning trees to be analytically solved exactly. We show how a judicious application of the prescribed growth rules selects for certain subsets of the spanning trees with particular desired properties (small world, extended diameter, degree distribution, etc.), and thus approximates and/or provides solutions to several optimization problems on undirected and unweighted networks. The analysis of spanning trees enhances the usefulness of recursive graphs as sophisticated models for everyday life complex networks.

DOI: [10.1103/PhysRevE.105.024312](https://doi.org/10.1103/PhysRevE.105.024312)

## I. INTRODUCTION

Recursive, finitely articulated models of complex networks mimic many key properties of real-world networks. At the same time, these models are amenable to *exact* analysis, providing important insights into the nature of everyday life complex networks. In this article, we revisit the subject of spanning trees for recursive scale-free nets, and provide a method for generating all spanning trees, enabling a more complete analysis of this important set of subgraphs.

The link-based method presented here is applicable to any recursive generative network model that produces finitely articulated graphs, but we focus on the well-known Dorogovtsev-Goltsev-Mendes (DGM) net [1] as a representative example of this class. This particular family of networks was introduced in Ref. [1] as a *pseudofractal scale-free web* (and was referred to as the PSW in subsequent literature), but is equivalent to the  $(u, v)$ -flower graph [2] when  $u = 1$  and  $v = 2$ . We use our notation  $F_v(1, 2)$  to denote the  $v$ th generation of this network [2], for which there are two equivalent ways to construct it recursively: the link-by-link approach and the hub-pasting approach. Both recursive growth processes create the same set of small-world complex networks with a hierarchical structure, a scale-free degree distribution with power-law exponent  $\gamma \approx 2.57$ , and a high clustering coefficient of  $C = 4/5$ , matching several characteristic benchmarks of real-life complex nets. The recursive nature and the finite articulation have enabled exact analytical studies of the statistics of cycles [3], diffusion [4], percolation [5], spectral properties [6], and minimum dominating sets [7] for this network, shedding light on the analogous properties in real-life complex nets. Previous work has exploited the hub-pasting approach of constructing this and similar models to enumerate the spanning trees (and calculate the subsequent *tree*

entropy [8] of the ensembles) [9–12], but additional results related to spanning trees of these networks are now obtained by considering a link-by-link approach.

After providing a brief description of both methods of constructing the original  $F_v(1, 2)$  in Sec. II, a few results on the spanning trees of these graphs will be provided from the hub-pasting viewpoint in Sec. III, before turning to the link-based approach. In Sec. IV, we provide an explicit process for efficiently generating the whole ensemble of spanning trees. We use the same recursive link-by-link process that creates the original network, except that the links are assigned an  $R$  or  $G$ , which stand for “real” or “ghost,” respectively. The term ghost indicates that a link is part of the original graph, but not part of the spanning tree, and we represent these links by dashed lines in all the figures.

Using the link-based approach, we first verify previous results for the enumeration of the labeled spanning trees of  $F_v(1, 2)$  in a simplified way and calculate the tree entropy of the general  $(u, v)$  flower for comparison with previous results. We then show that this approach finds added utility in solving optimization problems on undirected and unweighted networks of this type. While the well-known minimum spanning tree (MST) problem on weighted networks has garnered a lot of attention [13], because of its relevance to the famed *traveling salesman* problem [14,15] and its useful applications in data science [16–22], the MST is not clearly defined for unweighted networks. Common generalizations do exist, however, for the undirected and unweighted case including the dense, sparse, and minimum routing cost spanning tree problems, as well as many other lesser studied variations [23–34]. The solutions to these problems are relevant to an ever widening diversity of applications from optical network design [35,36] to networked oscillator synchronization [37]. We show that while these combinatorial problems

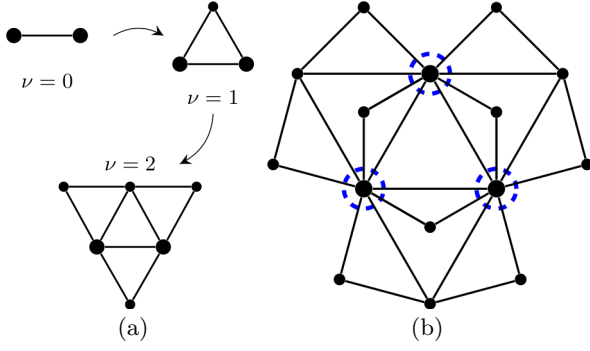


FIG. 1. Two equivalent methods of constructing  $F_v(1, 2)$  from  $F_{v-1}(1, 2)$ : (a) The first three generations of  $F_v(1, 2)$  are obtained by adding a path of length 2 in parallel to each link from the previous generation and (b) the construction of  $F_3(1, 2)$  is obtained by identifying the hub vertices of  $u + v = 3$  copies of  $F_2(1, 2)$ ; the three nodes resulting from the hub identifications (the new hubs) are indicated by dashed circles for clarity.

are often  $NP$ -hard in general [23,24,35], for recursive, finitely articulated networks, such optimal spanning trees (and indeed the whole ensemble of spanning trees) can now be created alongside the original network with minimal overhead. We conclude this article by describing a process for selecting such optimal spanning trees through imposing restrictions on the application of the link-based rules, and provide an interesting open problem that we hope this work sheds light on.

## II. CONSTRUCTING $F_v(u, v)$

Starting from an initial seed graph of the complete graph on two vertices, i.e.,  $K_2$ , in each generation of the growth process, we replace each link from the previous generation with a pair of paths of length  $u$  and  $v$ , in parallel, between the nodes of the original link. For the case of  $u = 1$  and  $v = 2$ , this is equivalent to just adding a new path of length two in parallel to the existing link as shown in Fig. 1(a), which illustrates the first three generations of  $F_v(1, 2)$ .

For this case, it follows that for each link from the previous generation, one new node is added to the network, while  $w = u + v = 3$  links replace each old link in the new generation. Thus, the size (total number of links),  $M_v$ , and order (total number of nodes),  $N_v$ , of the  $v$ th generation are given by  $M_v = w^v = 3^v$  and

$$N_v = \frac{w-2}{w-1}w^v + \frac{w}{w-1} = \frac{1}{2}3^v + \frac{3}{2}, \quad (1)$$

where detailed derivations through recursion relations are provided elsewhere [1,2,38].

Alternately, one can construct  $F_v(1, 2)$  using a hub-pasting approach by taking  $w = u + v = 3$  copies of  $F_{v-1}(1, 2)$  and identifying one hub vertex (those of highest degree) on each copy with another hub vertex on each other copy resulting in  $F_v(1, 2)$ . This process is shown in Fig. 1(b) for the creation of  $F_3(1, 2)$  by hub-pasting three copies of  $F_2(1, 2)$ , where the identified hub nodes are indicated by dashed circles.

Prior to exploring the main contribution of this work within the context of the link-by-link approach, we first mention

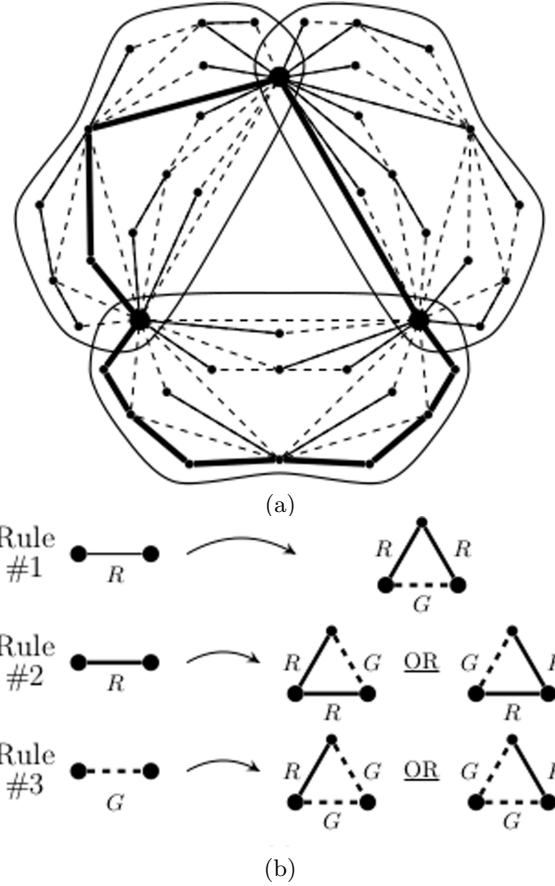


FIG. 2. As with the regular flower graph, there are two methods of constructing spanning trees of  $(1,2)$ -flower graphs: (a) The hub-pasting approach (here requiring an additional step) entails choosing  $1 + 2 = 3$  elements from the previous generation of spanning trees and identifying pairs of *hub* nodes as shown; this, however, results in a single cycle (bolded), which must be broken by removing one of its links. (b) The *link-by-link* approach involves applying one of three rules to each link in each generation recursively. Each sequence of valid rule operations results in a unique spanning tree, allowing for the enumeration and generation of all spanning trees in the ensemble.

two points that are best described within the hub-pasting paradigm.

## III. HUB-PASTING CONSTRUCTION AND THE TYPICAL DIAMETER

For generating spanning trees of  $F_v(1, 2)$ , one can begin by selecting (with replacement) three members from the ensemble of trees from generation  $v - 1$  and identifying them at the vertices that would be considered hubs of  $F_{v-1}(1, 2)$  as shown in Fig. 2(a). We will use the italicized version of the word *hub* to indicate these hub nodes of the original network when in reference to a spanning tree, whether those nodes in fact have the largest degree in the tree or not. This pasting process is done explicitly by temporarily adding a link between the *hub* vertices from adjacent spanning trees (chosen from the previous generation), and then contracting the graph along these temporary links to identify the three pairs of *hub* nodes into the *hubs* for the new generation. Due to the connectedness of

the three individual spanning trees, the identification process will form a single cycle through the new *hub* nodes (shown in bold). To obtain a spanning tree of the new generation, remove one of the links from this cycle. The ambiguity of this link removal step complicates the enumeration of spanning trees, which led to a more arduous accounting process used in previous work [9]. However, this simplified description provides insight into the typical diameter of the spanning trees of  $F_v(1, 2)$ .

In Ref. [3], the typical cycle length of  $F_v(1, 2)$  was shown to scale as  $2^v$  in the thermodynamic limit of large graphs. This implies that the typical spanning tree of  $F_v(1, 2)$  contains a path having length on the order of  $2^v$ . Given that the maximum diameter of any tree is also of order  $2^v$  (see Sec. V), then the typical spanning tree must have a diameter of order  $2^v$ . It is interesting that while the (1,2) flower is a small-world network, the diameter of the typical spanning tree of this network scales as  $\sim 2^v \sim N^{\log_3 2}$  [where the order of  $F_v(1, 2)$  from Eq. (1) is  $N = \frac{1}{2}3^v + \frac{3}{2} \sim 3^v$ ] and is therefore *not* small world. However, Sec. V shows how to select a small-world spanning tree from the (typically non-small-world) ensemble.

Additionally, we note that the “copy machine” algorithm outlined in Appendix C of Ref. [4], which is based on the hub-pasting approach, can be adapted to efficiently generate adjacency matrices of spanning trees of  $F_v(1, 2)$  in a manner similar to what was outlined for ensembles in Ref. [38].

#### IV. LINK-BY-LINK CONSTRUCTION AND TREE ENTROPY

The *link-by-link* approach to spanning tree construction will mirror the general link-by-link construction process of the recursive net, while keeping track of two different kinds of links: “real” links ( $R$ ) that belong to the spanning tree, and “ghost” links ( $G$ ) that belong to the original network, but not the spanning tree.

For  $F_v(1, 2)$ , in each generation, an  $R$  link is either replaced by a  $G$  link and two  $R$  links [rule No. 1 in Fig. 2(b)], or the  $R$  link is replaced by an  $R$  link and a path of one  $R$  link and one  $G$  link in either order (rule No. 2). Additionally, in the process going forward, a  $G$  link is always replaced by a  $G$  link and a path of one  $R$  link and one  $G$  link (rule No. 3). Other recursive scale-free network models such as the generic  $(u, v)$ -flower graph [2] or a similar construction described in Ref. [39] may require a larger set of rules with more cases to consider, making the subsequent analysis more tedious, but the process is, in essence, the same.

Denoting by  $R_v$  ( $G_v$ ) the number of real (ghost) links in generation  $v$ , the construction rules in Fig. 2 imply

$$\begin{aligned} R_{v+1} &= 2R_v + G_v, & R_0 &= 1, \\ G_{v+1} &= R_v + 2G_v, & G_0 &= 0, \end{aligned} \quad (2)$$

leading to  $R_v = \frac{1}{2}(3^v + 1)$  and  $G_v = \frac{1}{2}(3^v - 1)$ . This is consistent with the fact that  $R_v = N_v - 1$  and  $G_v + R_v = M_v$ , where  $M_v$  and  $N_v$  are as defined in Eq. (1).

Each sequence of link updatings results in a single unique spanning tree; moreover, each updating rule is uniquely reversible. Finally, the reverse rules applied (in an appropriate manner) to an arbitrary spanning tree of  $F_v(1, 2)$  generate a

spanning tree of  $F_{v-1}(1, 2)$  (i.e., no loops are created, and neither is the tree disconnected by the reverse rule application). It follows that there is a one-to-one correspondence between spanning trees and valid generating sequences of rules, and therefore these rules generate the complete set of labeled spanning trees.

This approach enables an independent verification of the number of labeled spanning trees of generation  $v$  for the DGM net [9]. Denote the number of spanning trees by  $T_v$ , and consider the recursion

$$T_{v+1} = (3^{R_v} \cdot 2^{G_v})T_v, \quad (3)$$

where the term in parentheses accounts for the 3 (2) ways of updating  $R$  ( $G$ ) links. Together with the expressions for  $R_v$  and  $G_v$  and the initial condition of  $T_0 = 1$ , we get

$$T_v = \left(\frac{3}{2}\right)^{\frac{v}{2}} 6^{(3^v - 1)/4}.$$

In terms of the order of the graph  $N$ ,

$$T_v = \left(\sqrt{\frac{3}{2}}\right)^{\log_3(2N-3)} 6^{(\frac{N}{2}-1)} \sim \frac{2^C}{6} N^C \sqrt{6}^N,$$

where  $C = [1 - \log_3(2)]/2$  is a constant. Although a different numbering convention for the generation was used here, this confirms the result found in Ref. [9].

For comparison, the total number of all labeled trees of order  $N$  is  $N^{N-2}$  [40]. Amusingly, this functional dependence of  $T_v$  with  $N$  is closer to that of *unlabeled* trees of order  $N$  [41].

For further comparison, this approach can be adapted to the general  $(u, v)$  flower by letting  $w = u + v$ . The recursion for the enumeration of all labeled spanning trees [analogous to Eq. (3)] is then  $T_{v+1} = w^{R_v} \cdot (uv)^{G_v} T_v$  with  $T_0 = 1$ , since there are  $w$  ways that any  $R$  link can be propagated (the spanning trees of a  $w$  cycle) and  $uv$  ways that any  $G$  link can be propagated (one link in each path having lengths  $u$  and  $v$  must be a ghost link). A recursion similar to Eq. (2) then gives  $R_v = \frac{w-2}{w-1}w^v + \frac{1}{w-1}$  and  $G_v = \frac{1}{w-1}w^v - \frac{1}{w-1}$ , which leads to the closed-form solution

$$T_v = w^{\frac{w-2}{w-1}(\frac{w^v-1}{w-1} + \frac{v}{w-2})} \cdot (uv)^{\frac{1}{w-1}(\frac{w^v-1}{w-1} - v)}.$$

This gives the tree entropy [8] of the  $(u, v)$  flower as

$$\lim_{N \rightarrow \infty} \frac{\ln T_v}{N} = \frac{\ln(w)}{w-1} + \frac{\ln(uv)}{(w-1)(w-2)},$$

So, while the (1,2) flower indeed has a small tree entropy when compared with other networks having an average degree of  $\langle k \rangle = 4$  as described in Ref. [9], it is also true that larger values of  $w$  will result in smaller tree entropies tending toward zero in the limit of large  $w$ , although the average degrees of flower graphs will tend toward  $\langle k \rangle = 2$  as  $w \rightarrow \infty$ . Finally, we note that the tree entropy for the  $w$  cycle is simply  $\frac{\ln(w)}{w}$ , meaning the tree entropy of the flower graphs is increased by an additional term that is quadratic in  $\frac{1}{w}$  due to the overlapping  $w$  cycles that define them.

## V. SPANNING TREE SUBSETS HAVING OPTIMAL CHARACTERISTICS

In many applications of spanning trees, one is not concerned with simply the number of trees, but one seeks an optimal spanning tree for some purpose, e.g., minimal branching vertex spanning tree (MBVST), dense spanning tree (DST), etc. Within the process outlined, due to the finite articulation of the models being addressed, large-scale properties can be defined in terms of recursion relations allowing such optimal trees to be selected from various subsets of the ensemble of spanning trees that share certain properties. This selection is done by restricting how the rules are applied, which alters the recursions that define these properties.

To illustrate, consider the maximum diameter of any spanning tree from the ensemble in generation  $v$ , which we denote by  $s_v^{\max}$ . If rule No. 1 is applied to all  $R$  links, then for each  $R$  link in the path that defined the diameter of the previous generation, we will have two  $R$  links in the path that contribute to the diameter of the new generation. At least for generation  $v > 2$ , there will also be two additional  $G$  links that will enable the diameter to be extended further, meaning we have the recursion  $s_v^{\max} = 2s_{v-1}^{\max} + 2$ , with initial condition  $s_2^{\max} = 5$ , leading to the formula  $s_v^{\max} = 7 \cdot 2^{v-2} - 2$ . So, applying rule No. 1 to all  $R$  links (excluding rule No. 2 entirely) leads to the diameter scaling  $\sim 2^v$ , and in fact, the application of any finite proportion of rule No. 1 to  $R$  links will result in a *non*-small-world spanning tree of these small-world networks.

These maximal diameter spanning trees are solutions to both the minimal branching vertex and the sparse spanning tree problems (MBVST and SST, respectively) [28,33–35]. So by applying rule No. 1 to all  $R$  links and reducing the ambiguity of rule No. 3 by choosing the  $R$  link to extend the diameter whenever possible (e.g., choosing the  $R$  link to be adjacent to nodes with smaller degree), a member of the subset of spanning trees with maximal diameter is obtained [see Fig. 3(a) for one of the subset of eight such spanning trees of  $F_4(1, 2)$  having  $s_4^{\max} = 26$  that are in fact solutions to the MBVST and SST problems].

Contrast this spanning tree with a spanning tree that is created by never applying rule No. 1, which is a small-world spanning tree since the diameter then scales as the logarithm of the order. In the class of trees where only rules No. 2 and No. 3 are applied, if one chooses  $R$  links to be adjacent to the largest degree node (breaking ties randomly), then one obtains a member of the subset of spanning trees with the smallest diameter, i.e., a solution to the dense spanning tree problem, which has a minimal Wiener index [30,33,42] [see Fig. 3(b)]. In particular, we obtain a spanning tree having the minimum diameter, defined by the recursion  $s_{v+1}^{\min} = s_v^{\min} + 1$ , since the ambiguity of rule No. 2 prevents any increase in the span, but there is one ghost link in each generation that must add to the span. The initial condition of  $s_1^{\min} = 2$  leads to  $s_v^{\min} = v + 1$ , meaning in terms of the order of the graph, we have  $s_v^{\min} \sim \log_3(N)$  in the limit of large graphs.

Letting a parameter  $p$  control the proportion of instances (when applying rules No. 2 and No. 3) where the  $R$  link is attached to the largest degree node, the tree in Fig. 3(b) is a representative of the set created using  $p = 1$ , and the tree

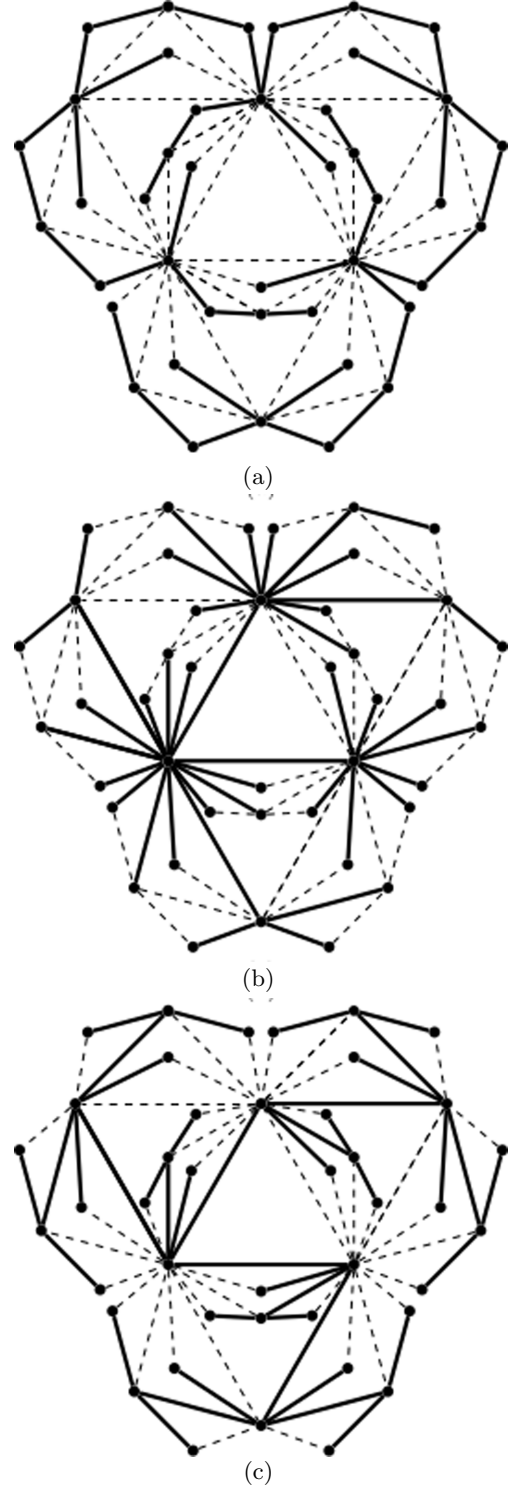


FIG. 3. Representative examples of subsets of spanning trees that can be selected from within the set of spanning trees of  $F_v(1, 2)$ : (a) using rules No. 1 and No. 3 only, where the ambiguity of rule No. 3 is reduced by maximizing the diameter when possible (breaking ties randomly) resulting in a solution to the MBVST problem; (b) using rules No. 2 and No. 3 only with  $p = 1$  resulting in a solution to the DST problem; and (c) using rules No. 2 and No. 3 only with  $p = 0$  leads to a balanced small-world tree with a more homogeneous degree distribution that is a candidate for a solution to the  $k$ -MBVST problem.



in Fig. 3(c) is representative of using  $p = 0$ . The parameter  $p$  then alters the degree distribution while retaining the small-world property, and values of  $p \in (0, 1)$  then define probability distributions over this subset of trees having small Wiener indices, providing a useful tool for approaching the minimal  $k$ -branching vertex problem ( $k$ -MBVST) [32,34].

## VI. CONCLUSION

In summary, a general approach for obtaining spanning trees of recursive, finitely articulated graphs has been presented through the example of the DGM net, and basic properties of the ensemble of all spanning trees, such as the typical or maximum diameter and the number of labeled trees are obtained exactly. Additionally, one can now select a spanning tree from subsets of the ensemble of spanning trees that share desirable properties such as small-world features, cer-

tain degree distributions, and a maximal or minimal diameter, effectively solving many optimal spanning tree problems and approximating solutions to others on such finitely articulated models.

Perhaps the most interesting related open question is to find the number of *unlabeled* spanning trees of recursive finitely articulated graphs. While this is, in general, an intractable problem, we hope that the recursive properties of these graphs might provide a convenient foothold to approach such a problem.

## ACKNOWLEDGMENTS

C.T.D. has received funding from the Air Force Office of Scientific Research (20RICOR010). E.M.B. has received funding from the Army Research Office (N68164-EG) and also DARPA.

- 
- [1] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, Pseudofractal scale-free web, *Phys. Rev. E* **65**, 066122 (2002).
  - [2] H. D. Rozenfeld, S. Havlin, and D. ben-Avraham, Fractal and transfractal recursive scale-free nets, *New J. Phys.* **9**, 175 (2006).
  - [3] H. Rozenfeld, J. E. Kirk, E. M. Bollt, and D. ben-Avraham, Statistics of cycles: How loopy is your network? *J. Phys. A: Math. Gen.* **38**, 4595 (2004).
  - [4] E. M. Bollt and D. ben-Avraham, What is special about diffusion on scale-free nets? *New J. Phys.* **7**, 26 (2005).
  - [5] H. D. Rozenfeld and D. ben-Avraham, Percolation in hierarchical scale-free nets, *Phys. Rev. E* **75**, 061102 (2007).
  - [6] P. Xie, Z.-Z. Zhang, and F. Comellas, On the spectrum of the normalized Laplacian of iterated triangulations of graphs, *Appl. Math. Comput.* **273**, 1123 (2016).
  - [7] L. Shan, H. Li, and Z.-Z. Zhang, Domination number and minimum dominating sets in pseudofractal scale-free web and Sierpinski graph, *Theor. Comput. Sci.* **677**, 12 (2017).
  - [8] R. Lyons, Asymptotic enumeration of spanning trees, *Comb. Probab. Comput.* **14**, 491 (2005).
  - [9] Z. Zhang, H. Liu, B. Wu, and S. Zhou, Enumeration of spanning trees in a pseudofractal scale-free web, *Europhys. Lett.* **90**, 68002 (2010).
  - [10] Z. Zhang, H. Liu, B. Wu, and T. Zou, Spanning trees in a fractal scale-free lattice, *Phys. Rev. E* **83**, 016116 (2011).
  - [11] Z. Zhang, B. Wu, and F. Comellas, The number of spanning trees in apollonian networks, *Discrete Appl. Math.* **169**, 206 (2014).
  - [12] W. Sun, S. Wang, and J. Zhang, Counting spanning trees in prism and anti-prism graphs, *J. Appl. Anal. Comput.* **6**, 65 (2016).
  - [13] R. L. Graham and P. Hell, On the history of the minimum spanning tree problem, *IEEE Ann. Hist. Comput.* **7**, 43 (1985).
  - [14] M. Held and R. M. Karp, The traveling-salesman problem and minimum spanning trees, *Oper. Res.* **18**, 1138 (1970).
  - [15] T. L. Magnanti and R. T. Wong, Network design and transportation planning: Models and algorithms, *Transport. Sci.* **18**, 1 (1984).
  - [16] L. S. Shapiro, Clustering using maximum affinity spanning trees, in *Affine Analysis of Image Sequences*, Distinguished Dissertations in Computer Science (Cambridge University Press, Cambridge, UK, 1995), pp. 61–99.
  - [17] Y. Xu, V. Olman, and D. Xu, Clustering gene expression data using a graph-theoretic approach: An application of minimum spanning trees, *Bioinformatics* **18**, 536 (2002).
  - [18] S. Varma and R. Simon, Iterative class discovery and feature selection using minimal spanning trees, *BMC Bioinf.* **5**, 126(2004).
  - [19] Q. He and C.-H. H. Chu, A new image segmentation technique using maximum spanning tree, in *Combinatorial Image Analysis*, edited by V. E. Brimkov, R. P. Barneva, and H. A. Hauptman (Springer, Berlin, 2008), pp. 197–204.
  - [20] L. Galluccio, O. Michel, P. Comon, and A. O. Hero, Graph based  $k$ -means clustering, *Signal Process.* **92**, 1970 (2012).
  - [21] S. E. Maeng, H. W. Choi, and J. W. Lee, Complex networks and minimal spanning trees in international trade network, *Int. J. Mod. Phys. Conf. Ser.* **16**, 51 (2012).
  - [22] J. L. Rainbolt and M. Schmitt, The use of minimal spanning trees in particle physics, *J. Instrum.* **12**, P02009 (2017).
  - [23] G. Salamon and G. Wiener, On finding spanning trees with few leaves, *Inf. Process. Lett.* **105**, 164 (2008).
  - [24] S. Silvestri, G. Laporte, and R. Cerulli, A branch-and-cut algorithm for the minimum branch vertices spanning tree problem, *Comput. Oper. Res.* **81**, 322 (2017).
  - [25] B. Y. Wu and K.-M. Chao, Minimum routing cost spanning trees, in *Discrete Mathematics and its Applications* (CRC Press, Boca Raton, FL, 2004), pp. 41–84.
  - [26] A. A. Dobrynin, R. Entringer, and I. Gutman, Wiener index of trees: Theory and applications, *Acta Appl. Math.* **66**, 211 (2001).
  - [27] D.-H. Kim, J. Dong Noh, and H. Jeong, Scale-free trees: The skeletons of complex networks, *Phys. Rev. E* **70**, 046126 (2004).
  - [28] R. Cerulli, M. Gentili, and A. Iossa, Bounded-degree spanning tree problems: Models and new algorithms, *Comput. Optim. Appl.* **42**, 353 (2009).

- [29] R. Silva, D. Silva, M. Resende, G. Mateus, J. Gonçalves, and P. Festa, An edge-swap heuristic for generating spanning trees with minimum number of branch vertices, *Optim. Lett.* **8**, 1225 (2014).
- [30] L. Székely, S. Wagner, and H. Wang, *Problems Related to Graph Indices in Trees* (Springer, Berlin, 2016), pp. 3–30.
- [31] T. Li, Y. Gao, Q. Dong, and H. Wang, Degree sums and dense spanning trees, *PLoS ONE* **12**, e0184912 (2017).
- [32] M. Merabet, J. Desai, and M. Molnar, A generalization of the minimum branch vertices spanning tree problem, in *Combinatorial Optimization*, edited by J. Lee, G. Rinaldi, and A. R. Mahjoub (Springer, Cham, 2018), pp. 338–351.
- [33] M. Ozen, G. Lesaja, and H. Wang, Globally optimal dense and sparse spanning trees, and their applications, *Stat. Optim. Inf. Comput.* **8**, 328 (2020).
- [34] V. Božović, Ž. K. Vukićević, G. Popivoda, R.-Y. Pan, and X.-D. Zhang, Extreme Wiener indices of trees with given number of vertices of maximum degree, *Discrete Appl. Math.* **304**, 23 (2021).
- [35] L. Gargano, P. Hell, L. Stacho, and U. Vaccaro, Spanning trees with bounded number of branch vertices, in *Automata, Languages and Programming*, edited by P. Widmayer, S. Eidenbenz, F. Triguero, R. Morales, R. Conejo, and M. Hennessy (Springer, Berlin, 2002), pp. 355–365.
- [36] A. Marín, Exact and heuristic solutions for the minimum number of branch vertices spanning tree problem, *Eur. J. Oper. Res.* **245**, 680 (2015).
- [37] C. T. Diggans, J. Fish, Abd Al Rahman R. AlMomani, and E. M. Boltt, The essential synchronization backbone problem, *Chaos* **31**, 113142 (2021).
- [38] C. Tyler Diggans, E. M. Boltt, and D. ben-Avraham, Stochastic and mixed flower graphs, *Phys. Rev. E* **101**, 052315 (2020).
- [39] Z.-Z. Zhang, L.-L. Rong, and S. Zhou, A general geometric growth model for pseudofractal scale-free web, *Physica A* **377**, 329 (2007).
- [40] A. Cayley, A theorem on trees, in *Cambridge Library Collection - Mathematics* (Cambridge University Press, Cambridge, UK, 2009), Vol. 13, pp. 26–28.
- [41] R. Otter, The number of trees, *Ann. Math.* **49**, 583 (1948).
- [42] H. Wiener, Structural determination of paraffin boiling points, *J. Am. Chem. Soc.* **69**, 17 (1947).