Designing threshold networks with given structural and dynamical properties

Aric Hagberg and Pieter J. Swart
Mathematical Modeling and Analysis, Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA

Daniel A. Schult
Department of Mathematics, Colgate University, Hamilton, New York 13346, USA

(Received 14 July 2006; published 29 November 2006)

The threshold model can be used to generate random networks of arbitrary size with given local properties such as the degree distribution, clustering, and degree correlation. We summarize the properties of networks created using the threshold model and present an alternative deterministic construction. These networks are threshold graphs and therefore contain a highly compressible layered structure and allow computation of important network properties in linear time. We show how to construct arbitrarily large, sparse, threshold networks with (approximately) any prescribed degree distribution or Laplacian spectrum. Control of the spectrum allows careful study of the synchronization properties of threshold networks including the relationship between heterogeneous degrees and resistance to synchrony.

DOI: 10.1103/PhysRevE.74.056116 PACS number(s): 89.75.Hc, 89.75.Da, 89.75.Fb, 05.45.Xt

I. INTRODUCTION

Discovering and modeling the structure of biological, social, and technological networks has been the subject of intense research in recent years. This activity has been fueled by the increasing availability of large experimental data sets and the discovery that real networks share common topological properties that are quite different from classic random networks. The network structure, encoded in the links (edges) between nodes, is important for many applications including gene transcription regulation and protein interaction [1], strategies for the control of epidemics [2,3], understanding network robustness against failures or deliberate attack [4], and discovering communities in social systems [5]. Many real-world networks display an approximate power-law degree distribution $p(k) \sim k^{-\gamma}$, typically with $2<\gamma<3$ [6], and also tend to have high clustering with low diameter, a structure labeled "the small world effect" [7]. The results on modeling the structure of networks using these two descriptions and others are reviewed in Refs. [8–12].

Global properties, such as the diameter and size of the largest component, have dominated structural analysis of networks. With the increasing availability of a wide array of more detailed experimental data sets and accompanying sophisticated network models, attention has now focused on distributions of local statistics such as degree (number of edges at a node), clustering (number of triangles), and degree-degree correlation (propensity for nodes of degree k to connect to nodes of similar degree).

Most models for the creation of networks with a specific degree distribution, clustering, or other statistical properties are growth models where nodes and edges are added to create the desired features in the limit of a large number of nodes [6,12]. These models are in contrast to "static models" such as the classic Erdős-Rényi random graph [13] or the configuration model [8] and generalization to networks with a given expected degree sequence [14], where a fixed number of nodes, N, is specified and edges are connected between them randomly. More recently, an interesting class of static models has been developed that can generate networks with statistical properties similar to growing network models [15–19]. These models assign a real variable x_i to each node i which is termed the "node weight," "intrinsic fitness," "hidden variable," or "propensity to form edges." In general form, such hidden variable models assign node weights randomly from a specified probability density function $\rho(x)$ and then assign edges between nodes i and j with another probability given by a symmetric function $f(x_i,x_j)$.

A preferential attachment method for building networks with the addition of a node fitness parameter was presented in Ref. [12]. A static model using node intrinsic fitness was then used to study data packet transport through scale-free networks [15] and the size of connected components in a model with a discrete set of node fitness values [18]. Further studies showed that scale-free networks could be constructed from node fitness models even if the distribution of node weights was not scale free [16,17].

In the general setting of random networks constructed from hidden variables, one focus has been the derivation of statistical properties of the network in terms of the distribution of node fitness and the probability of connected nodes [19,20]. Note that static models can always be put into a network growth framework if the node weights are assigned during a growth process.

In this paper, we further analyze the threshold model studied in Refs. [16,19,21]. The threshold model creates random networks through specification of a density function $\rho(x)$ for independently assigned weights on nodes x_i, $i=1,\ldots,N$, and a threshold value θ. Once node weights x_i have been assigned, each possible edge (x_i,x_j) $(i\neq j)$ is created if $x_i+x_j>\theta$. Here we show that the threshold model creates networks that exactly meet the definition of a threshold graph in the graph theory context [22,23] and we call the resulting networks "threshold networks." These threshold networks display many novel properties.

In Sec. II, we discuss the basic structure of threshold networks. We summarize the predictions of distributions of network structure measures for the threshold model in the limit of large network size. We exploit their highly compressible layered structure to derive compact representations. We then show how to construct arbitrarily large, sparse, threshold networks with (approximately) any prescribed degree distribution. In Sec. III, we describe the fast computation of many relevant measures of network structure, including the Laplacian spectrum and eigenvectors. This allows us to construct arbitrarily large, sparse, threshold networks with (approximately) any prescribed Laplacian spectrum. We then use these tools to provide insights into the dynamics on threshold networks, in particular when dynamic coupling is of diffusive type so that the Laplacian spectrum is relevant. Control of the spectrum allows careful study of diffusive dynamics and synchronization properties, including the relationship between heterogeneous degrees and resistance to synchrony. Section IV provides a summary and conclusions.

II. THRESHOLD NETWORKS

In this section we review some properties of threshold networks and some previous results from studies of the threshold model. We present a compact description of threshold networks and a deterministic algorithm for creating threshold networks with a chosen degree distribution.

A. Properties of threshold networks

Threshold networks have been studied extensively in the graph-theoretic literature [24,25] with reviews in Refs. [23,26]. They belong to the chordal, cochordal, comparability, cocomparability, interval, split, and permutation graph classes ([27], p. 23). In threshold networks, the neighborhoods of nodes are nested (i.e., include the neighbors of every lower-degree node), and the graph can therefore be partitioned into groups of identical nodes according to this nesting. We use this partition to provide a compact notation, which we call the creation sequence, for storing and manipulating these networks.

The structure of a threshold network is uniquely determined (modulo relabeling of nodes with the same degree) by its degree sequence [23,28]. It is worth noting that threshold networks are not the only networks with this property. Also note that the degree sequence restricts the number of nodes, while the degree distribution would not.

Threshold networks include many common networks such as the star (one hub connected to many points) and complete graph. They also include networks with a wide range of properties. For example, the density, or fraction of possible edges present, for a connected threshold network ranges over all possible values from $2/N$ (stars) to 1 (complete graphs). As we will show, for large N we can create a threshold network with any approximate degree distribution. In addition, threshold networks have the interesting property that the Laplacian spectrum consists solely of integers [28]. The primary restriction in terms of network properties seems to be the extremely small diameter (when connected), which is at most 2. In addition, the threshold nature results in a nested neighborhood structure. That is, the set of neighbors of each node is a subset of the neighbors of every higher degree node. This rigid structure may seem unrealistic as a model for real networks. But they certainly occur as important subnetworks containing hubs of real networks: the prototype hub, the star, is a threshold network. Furthermore, networks that are similar but not exactly threshold in nature may be approximated in many situations by the threshold networks they mimic. For example, models where the probability of attachment between nodes depends in a smooth way on the sum or product of node weights can produce networks very similar in structure to threshold networks. Finally, network properties other than diameter can be engineered quite flexibly and quickly. This allows us to determine the impact of these network properties.

B. Threshold model

The threshold model generates a threshold network of N nodes each with a weight chosen randomly from a distribution $\rho(x)$. Edges are then formed for any pair of nodes (i,j) for which $x_i + x_j > \theta$ for some threshold value θ. The network structure is completely determined by the choice of $\rho(x)$ and θ. Without loss of generality we can take $\theta=1$, although sometimes it is easier to vary θ rather than scale the weights x.

The degree correlations and clustering of the threshold model in the large-N limit were analyzed in Ref. [19]. Masuda *et al.* [21] demonstrated that many forms of the density function lead to power-law degree distributions with exponent 2. We review these results here briefly.

Let the cumulative distribution function for weights be $R(x)=\int_{-\infty}^{x}\rho(x)dx$. Then the expected degree for a node with weight x is given by

$$E(k|x) = (N-1)[1-R(\theta-x)]. \tag{1}$$

Letting $\tilde{k}=k/(N-1)$, we obtain an approximation for the degree distribution $P(k)$,

$$P(\tilde{k}) = \rho(x)\frac{dx}{d\tilde{k}} = \frac{\rho(x)}{\rho(\theta-x)} = \frac{\rho(\theta-R^{-1}(1-\tilde{k}))}{\rho(R^{-1}(1-\tilde{k}))}. \tag{2}$$

This distribution is computed for a number of forms of the density function. Two results are worth special attention. First, power-law distributions arise for large k (on the order of N) when density functions decay more rapidly than power law. Second, a power-law density with exponent $a+1$ yields a power-law degree distribution with exponent $(a+1)/a$. Thus to obtain a power-law degree distribution with specified exponent γ, one should arrange $\rho(x)$ to be power law with exponent

$$a+1 = \frac{\gamma}{\gamma-1}.$$

Unfortunately, finding the density function which produces a given degree distribution involves solving the integral equation

$$P((N-1)[1-R(\theta-x)])\rho(\theta-x) = \rho(x), \tag{3}$$

with $P(\cdot)$ known and $\rho(\cdot)$ [and thus $R(\cdot)$] unknown. This equation is nonlinear for most interesting distributions. Thus, while we know how to generate certain specific power-law distributions, there is no formulation for general distributions.

The threshold model thus joins the collection of random network generation methods that produce realizations which, in the limit of large networks, approach a desired degree distribution. One distinct advantage of this model is that it provides analytical insight, fast algorithms, and concrete construction without assuming any lattice or hierarchical symmetry.

C. Creation sequence

Threshold networks can be described very compactly by what we call a "creation sequence" S: a binary string that provides a recipe for construction by reading the sequence from the left and adding one node for each digit. There are two types of nodes: dominating nodes, denoted by a 1, and isolated nodes, denoted by a 0.

For any given binary sequence, construct a corresponding threshold network as follows: reading the sequence from left to right, for each digit in the creation sequence a new node is added to the existing network with connectivity determined by its type. A dominating node 1 is connected to all nodes already in the network, and an isolated node is added without connecting it to the existing network. At the end of this construction, each dominating node will be connected to all the nodes that preceded it (to the left) in the creation sequence and to all dominating nodes to its right in the creation sequence. Correspondingly, each isolated node 0 will be connected only to the dominating nodes to its right. This notation is ambiguous for the leftmost node, so by convention the leading bit is assigned 1. It is clear from this construction that the set of all 1 nodes forms the maximal clique, that the set of all 0 nodes is connected only to nodes in the maximal clique, and that neighborhoods are nested from low degree to high degree. For example, the creation sequence 10001 represents the star S_5, 11111 represents the complete graph K_5, and 10101 and 10000101 represent the networks in Figs. 1(c) and 1(d).

The creation sequence description of a threshold network is equivalent to the weight formulation (specifying node weights x_i and a threshold θ). That is, given node weights and a threshold, we can obtain the unique creation sequence which describes the corresponding threshold network. Conversely, given a creation sequence we can find a set of node weights and a threshold which produce the corresponding threshold network. We proceed to a proof of the first statement, postponing the proof of the converse to later in this section.

An algorithm for obtaining the creation sequence from a sorted list of node weights, $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_N$, with a threshold value θ constructs the creation sequence from right to left as follows. First consider the sum $x_1 + x_N$ of the largest and smallest weights in the list. If $x_1 + x_N \leqslant \theta$, then node 1 will not connect to any of the remaining nodes on the list. That means that the node is an isolated node, so prepend a 0 to the creation sequence and discard the lowest weight from the list. If $x_1 + x_N > \theta$, node N will connect to all the remaining nodes. Thus it is a dominating node, so prepend a 1 to the creation sequence and discard the largest weight from the list. Repeat this process with the new largest and smallest weights on the list. When only one weight remains, this node connects to all nodes with larger weight and no nodes with smaller weight. It is both dominating (a member of the clique) and isolated (only attaches to nodes in the clique). By convention for this node we use a leading 1 to complete the creation sequence. The connectivity imposed by the weights is precisely that of the resulting creation sequence as describe above.

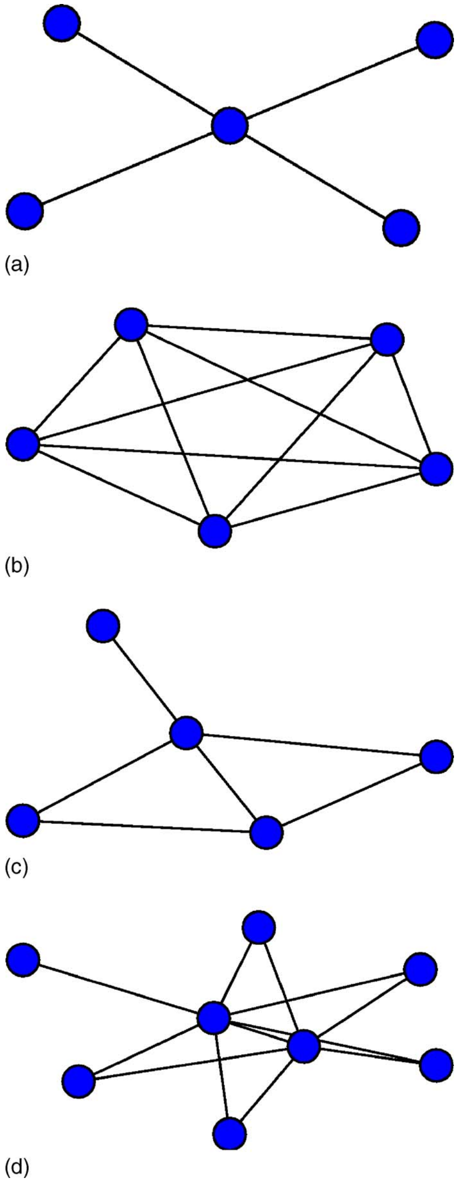

FIG. 1. (Color online) Some threshold networks and corresponding creation sequences. (a) Star with representation $S = 10001$. (b) Complete graph K_5 with representation $S = 11111$. (c) Network with representation $S = 10101$. (d) Network with representation $S = 10000101$. The creation sequence consists of dominating 1 nodes and isolated 0 nodes and is read from left to right. In (a) one dominating node is connected to four isolated nodes, one of which, appearing first in S, is technically both dominating and isolated and by convention denoted 1.

In the absence of additional structure, networks are generally incompressible; i.e., nearly all N-node graphs require $O(N^2)$ bits in any lossless representation. The obvious binary nature of the creation sequence provides storage for threshold networks, which requires at most N bits. We will show that this compact storage also facilitates computation of many network properties. The algorithms presented here work directly with creation sequences, so there is no overhead for retrieval. The transformation from node weights to creation sequence involves sorting the weights and thus $O(N \ln N)$ time. Indeed calculating network properties (such as the degree) also requires sorting the node weights. So it is much more efficient in storage and algorithmic speed to store network connectivity in creation sequence form than via node weights. Our algorithms assume the network is prescribed by a creation sequence.

Adjacent bits in S of the same type represent nodes identical in connectivity. If node labeling is not needed, an even more compact representation C can be obtained by compressing run lengths of similar node types. This is done by counting how many adjacent bits are the same starting from the left. The counts are then listed $C=(D_1,I_1,D_2,I_2,\ldots,D_n,I_n)$ where D_j is the number of dominating nodes in the jth group of 1's and similarly I_j is the number of isolated nodes in the jth group of 0's. Thus the creation sequence $S=11000110$ has compact representation $C=(2,3,2,1)$. Note that the first number always represents dominating nodes. As examples, $C=(1,3,1)$, $C=(5)$, $C=(1,1,1,1,1)$, and $C=(1,4,1,1,1)$ denote the four networks in Fig. 1. For some network properties, the algorithms are actually faster when exploiting a compact creation sequence, allowing the computation of network properties of multimillion node dense or sparse threshold networks on a modern desktop computer.

The high compressibility of threshold networks derives from two facts: (i) groups of adjacent nodes in the creation sequence with the same type are identical (up to node relabeling), and (ii) connectivity is fully determined by a sequence of length $<N$. This observation motivates a visual framework (based on the proof of theorem 1.2.4 in Ref. [23]) that more clearly exposes the underlying network structure.

We call the visual framework a layer-cake description of the network and an example is shown in Fig. 2. The framework, or cake, is made up of multiple layers, each filled with two groups of identical nodes. The right side is filled with dominating nodes while the left is filled with isolated nodes. Moving from left to right in C, we place nodes from bottom to top of the cake with layer j having I_j nodes on the left side and D_j nodes on the right. The top left layer consists of degree zero nodes (which are sometimes not considered as part of the network), and the remainder is a connected network with each dominating group forming a complete subnetwork connected to all nodes below that layer. Nodes in each isolated group connect only to nodes in the dominating groups from layers above. The network shown in Fig. 2 has $C=(2,4,3,6,5,1,1)$. Note that the network visualization in Fig. 2 shows how much clearer the structure becomes when depicted instead by the layer-cake diagram.

We now prove the converse of the connectivity equivalence between node weights and creation sequences stated above. That is, we show that from a creation sequence we can construct a set of node weights and a threshold so that the corresponding threshold networks are the same. Note that this set of node weights is not unique for a given network. Starting from a compact creation sequence C with n layers, assign identical node weights to each node by group starting with $x=1$ for the I_n nodes in the upper left group of the layer cake, moving down the left side and then up the right increasing the weight by 1 for each group. The upper right group has D_n nodes each with node weight $x=2n$ (the total number of groups). Setting the threshold to be $\theta=2n+1$ then ensures that the node weight threshold criterion for edges produces exactly the connectivity of the layer cake (and thus the creation sequence).

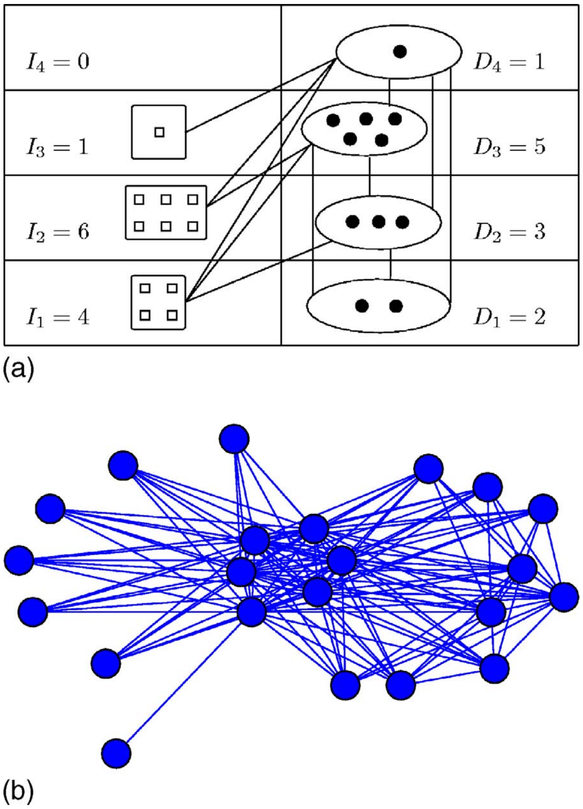

FIG. 2. (Color online) Two representations of a threshold network with the compact creation sequence $C=(2,4,3,6,5,1,1)$. Top: layer-cake depiction of the network showing the structure of the network by different node types. The layers are numbered from bottom to top with the isolated nodes (small squares) on the left and the dominating nodes (small circles) on the right. Ovals represent complete subgraphs for the surrounded nodes while rectangles are groups of isolated nodes that are not connected to each other. Lines between the node group outlines mean that all nodes in each group are connected to all nodes in the other group. Bottom: a two-dimensional layout of the network.

D. Custom degree distributions

In addition to randomly created networks based on the threshold model, we now show how to construct arbitrarily large, sparse, threshold networks with (approximately) any prescribed degree distribution or Laplacian spectrum. A generalization of this problem to vertex hidden variable models with scale-free degree distributions was studied in Ref. [17].

Let p_k be a discrete degree distribution that is normalized so that $\Sigma_k p_k = 1$. The goal is to construct a threshold network with approximately N nodes and with node degrees following the degree distribution p_k. The maximum degree k_{max} must be less than N, and the number of nodes with degree k should be $n_k = \lfloor N p_k \rfloor$. The construction strategy involves first creating a specific degree distribution realization $\{n_k\}$ from the discrete distribution. The resulting degree sequence need not be graphical since our algorithm adjusts the number of nodes and edges slightly to be able to build a threshold network.

We build a threshold network with approximately this distribution by using isolated (0) nodes to fill out the degree distribution and dominating (1) nodes to keep the network connected and distinguish between isolated nodes of differing degrees. With this construction, we create N isolated nodes and k_{max} dominating nodes. While the number of nodes is thus larger than N, for large sparse networks it is close to N. Similarly, there will be a small number of nodes (the k_{max} dominating nodes) of very high degree. These will generally affect the connectivity of the network but they do not alter the degree distribution very much for large networks.

As a simple example consider the degree distribution [1,2,3,5] which we will assume is chosen from a given p_k. Starting at the highest degree k, we create n_k isolated nodes followed by a single dominating node. The highest degree in the distribution is 4 of which there are 5 nodes so the sequence starts with 100001 (the first node in the creation sequence is always 1, but it will be treated as an isolated node for this construction process). Descending in degree with the same algorithm we find the creation sequence $S = 100001000100101$ or $C = (1,4,1,3,1,2,1,1)$. The corresponding threshold network has 15 nodes instead of 11, and the degree distribution is [1,2,3,5,0,0,0,1,0,0,1,0,1,1], which has 4 nodes of degree higher than 4 that do not exactly match the original degree distribution. While these few extra nodes certainly influence the connectivity or topology of the network, they do not significantly affect the degree distribution in the limit of large networks.

More complicated threshold network examples can be generated by following the same construction procedure with n_k given by a prescribed distribution. The general compact creation sequence alternates one dominating node with n_k isolated nodes from $k = k_{max}$ to $k = 1$: $C = (1, n_{(k_{max})} - 1, 1, n_{(k_{max}-1)}, 1, n_{(k_{max}-2)}, 1, \ldots, 1, n_1, 1)$. For example, if the n_k's are produced by a Gaussian function, the threshold network has an approximately Gaussian degree distribution as shown in Fig. 3. If the n_k's are from a power-law function, then the threshold network has a power-law degree distribution (Fig. 4).

Networks created with this method have high clustering and low assortativity (negative degree-degree correlations). Every isolated node has a clustering coefficient equal to 1 so the average clustering is more than $N/(N + k_{max})$. Since the lowest-degree nodes are connected only to the highest-degree nodes and this pattern continues through the layer cake, the networks are disassortative. This method can also be used to construct networks with approximately any Laplacian spectrum using the connection between the spectrum and the degree distribution described in Sec. III B.

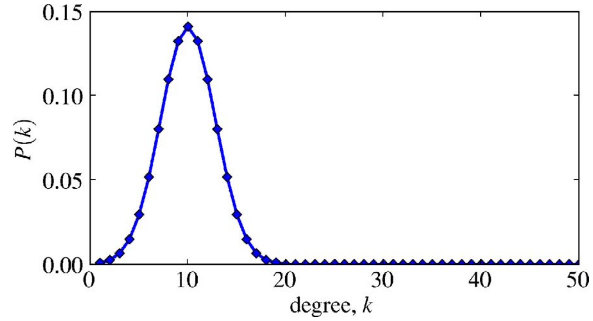

FIG. 3. (Color online) The degree distribution $P(k)$ for a threshold network that is approximately Gaussian. The threshold network was generated as described in the text with the degree distribution given by $n_k = a e^{-[(k-c)/w]^2}$ for $k = 1, \ldots, 20$, $a = 1000$, $c = 10$, and $w = 4$.

We turn now from construction of threshold networks to description of computation of network properties including dynamics where coupling occurs over the threshold network.

III. STRUCTURE, SPECTRUM, AND SYNCHRONIZATION

In this section, we describe algorithms for fast computation of many network properties, including both local structure measures such as clustering and degree and global structure measures such as betweenness centrality and the Laplacian spectrum. We then examine implications for network synchronization with diffusively coupled oscillators.

A. Fast computation

We now describe a series of algorithms for fast computation of network properties. Most algorithms start with the creation sequence, so we note that the algorithm for obtaining the creation sequence from a sorted list of weights described in Sec. II C is linear in the number of nodes. The degree of each node is obtainable in linear time from the

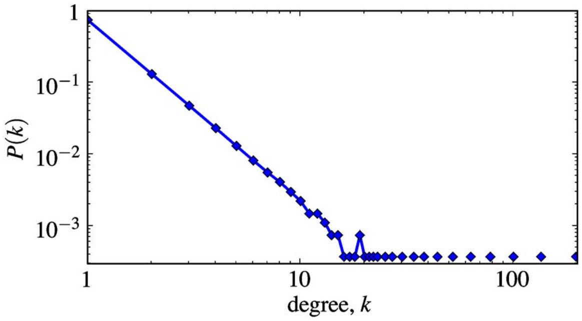

FIG. 4. (Color online) The degree distribution $P(k)$ for a threshold network that is approximately a power law. The threshold network was generated as described in the text with the degree distribution given by $n_k = a k^{-2.5}$ for $k = 1, \ldots, 20$ and $a = 2000$.

creation sequence. In addition, the Laplacian spectrum, clustering, and betweenness centrality can be similarly calculated in time linear in the number of nodes.

Following are some algorithms for computing statistical properties of threshold networks for a given creation sequence. For brevity, we introduce the notation

$$N_D = \sum D_j, \quad D_\ell^+ = \sum_{j>\ell} D_j, \quad I_\ell^- = \sum_{j<\ell} I_j, \quad N_\ell^- = \sum_{j<\ell} (D_j + I_j).$$

(a) Degree. The degree of an isolated node at level ℓ is simply $k = D_\ell^+$. The degree of a dominating node at level ℓ is $k = N_D + I_\ell^- - 1$. As an example, the network in Fig. 1(d) has $C = (D_1, I_1, D_2, I_2, D_3) = (1,4,1,1,1)$, so the (ordered as in C) degree sequence is (2,2,2,2,2,6,1,7).

(b) Triangles. The number of triangles for an isolated node at level ℓ is $D_\ell^+(D_\ell^+ - 1)/2$. The number of triangles for a dominating node at level ℓ is made up of $(N_D - 1)(N_D - 2)/2$ triangles in the clique and $\Sigma_{j<\ell} I_j (D_j^+ - 1)$ triangles formed with isolated nodes. The example network in Fig. 1(d) has triangle sequence (1,1,1,1,1,5,0,5).

(c) Betweenness centrality. Betweenness centrality is the number (or fraction) of shortest paths which pass through a given node excluding paths for which the node is an endpoint. If more than one path has the same length, the count is split between them. The relevant shortest paths in a threshold network are always length 2, go through a dominating node, and have an isolated node at one end. Thus, isolated nodes have zero betweenness centrality. For dominating nodes, betweenness is the same within each layer. The betweenness $B_{\ell+1}$ for dominating nodes in layer $\ell+1$ is computed from the formula $B_1 = 0$ and

$$B_{\ell+1} = B_\ell + \frac{I_\ell(I_\ell - 1)}{D_\ell^+} + \frac{2I_\ell(N_\ell^- + D_\ell)}{D_\ell^+}. \tag{4}$$

The first term appears because all paths through lower dominating nodes have the same path length as paths through this node. In fact, the only paths through this layer's nodes that are shorter than paths through lower dominating nodes have one end in the previous layer's isolated nodes I_ℓ. The second term represents all paths from nodes in I_ℓ to other nodes in I_ℓ. The third term represents paths from nodes in I_ℓ to each node in other groups below layer $\ell+1$. The second and third terms are divided by D_ℓ^+ to account for shortest paths of the same length that go through dominating nodes above level ℓ. The example network in Fig. 1(d) has betweenness sequence (0,0,0,0,0,10,0,22).

B. Laplacian spectrum

The Laplacian spectrum can be calculated easily from the degree sequence and has some unique properties. If A_{ij} are the elements of the adjacency matrix of a network, then the (combinatorial) Laplacian is defined as $L = D - A$ where D is the diagonal matrix of degrees (the sum of the rows of A). The Laplacian of a network is useful for modeling diffusion processes which move through edges. If a time-varying scalar field $u(i,t)$ is assigned to the nodes of the network and spreads via diffusion, the equation governing this motion might be $u_t = f(u) - Lu$. Notice that the conventional sign for L is opposite to that for the continuous Laplacian operator. The Laplacian spectrum of a network is the matrix spectrum of L.

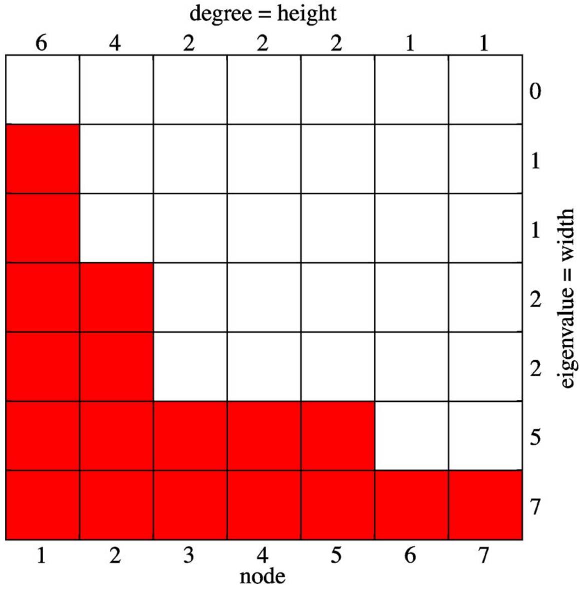

FIG. 5. (Color online) The Ferrer diagram for the degree sequence (6,4,2,2,2,1,1). Each column has a number of squares to match the degree of a node. Because this degree sequence represents a threshold network, the spectrum for the Laplacian can be obtained by counting squares in each row to get (7,5,2,2,1,1,0).

The Laplacian spectrum for a threshold network is entirely made up of integers, and it can be determined immediately from the degree sequence via transposition of the Ferrer diagram [25,29,30]. The Ferrer diagram (Fig. 5) is a visual depiction of the sorted degree sequence where the degree k of each node is represented by a stack of k squares. The eigenvalues are found by transposing this diagram or simply counting the number of squares in each of the N rows. Note that the top row will always be empty ($k_i < N$); hence, zero is always an eigenvalue. It corresponds to the constant eigenvector since the sum of each row of L is zero. In fact there is an eigenvalue $\lambda = 0$ for each connected component of the network and the corresponding eigenvectors are constant on each component.

The eigenvectors are only slightly more difficult to obtain than the eigenvalues. Threshold networks are constructed by combining groups of nodes as shown in the layer cake using the two binary graph operations of union and join. Unions of two networks take the union of the node and edge sets. Joins of two networks take the union and then add edges from each node of one network to each node of the other. Using the layer cake we can form an expression of unions and joins which completely characterizes the network. First notice that each dominating group is a complete subnetwork, while each isolated group is a stable (self-isolated) subnetwork. Using notation from the compact creation sequence, the network can be constructed layer by layer starting at the bottom. At step j join the current network with a complete graph on D_j nodes. Then union the network with I_j isolated nodes. This formulation of the network construction process allows us to build the eigenvectors of the network because the effect of these operations on eigenvectors is understood [30].

Notice that each row of the Laplacian matrix is associated with a node. The elements of the eigenvectors are similarly associated with a node. Assume that G is connected, and denote the constant eigenvector as x_0 (note that it is associated with the single zero eigenvalue). Let $X(G)$ denote the remaining $N-1$ orthogonal eigenvectors. The following observations were attributed in [30] to the Laplacian “folklore” and can be readily checked using matrix manipulation of the Laplacian matrix.

(i) When a network G is joined or unioned with another network, the eigenvectors can be *extended* to the new larger network by assigning zero values at all nodes not in G.

(ii) The spectrum of the union of two networks G_1 and G_2 with respective number of nodes N_1 and N_2 is the union of the spectra of the individual networks. The eigenvectors consist of the vector x_0, extensions of $X(G_1)$ and $X(G_2)$ and an eigenvector $x_{N_1+N_2-1}$ that identifies the networks. This latter eigenvector, which we call the *identifier eigenvector*, is chosen to have entries $-N_2$ for each node in G_1 and N_1 for each node in G_2. Thus it is orthogonal to the other eigenvectors (is constant on the nodes of each of G_1 and G_2) and has associated eigenvalue $\lambda_{N_1+N_2-1}=0$.

(iii) The join of two networks G_1 and G_2 with number of nodes N_1 and N_2 is only slightly more complicated. The eigenvectors are the same as for a union. The associated eigenvalues are increased by the number of edges added to each node in that network. That is, the eigenvalues associated with $X(G_1)$ increase by N_2 and those of $X(G_2)$ increase by N_1. The identifier eigenvector $x_{N_1+N_2-1}$ has associated eigenvalue N_1+N_2.

These results follow from the definitions of join, union, Laplacian, and eigenvector and can be checked via straightforward calculation.

We now describe the simultaneous construction of the spectrum and eigenvectors. Start with the complete graph K_{D_1}. Take as its eigenvectors x_0 and D_1-1 mutually orthogonal vectors which are also orthogonal to x_0. Standard (and simple) choices are shown in Fig. 6. Other orthogonal choices can clearly be used as well. Note also that the subgraphs on the left side of the layer cake are initially isolated ($K_{I_1}^c$) with the same eigenvector structure as K_{I_1}, but zero eigenvalues.

Each union and join operation involves forming new eigenvectors as described in (ii) and (iii) above and either (unions) retaining the eigenvalues or (joins) adding to each eigenvalue the number of nodes in the other network. As an example, the Laplacian and eigenvector matrix for the network in Fig. 2 appears in Fig. 6.

The resulting spectrum is actually much simpler than the construction process and can be computed quickly as follows. Each isolated node contributes an eigenvalue equal to its degree. The leading node in S contributes the zero eigenvalue. All other dominating nodes contribute an eigenvalue one more than their degree. Thus, to compute the spectrum, we need only compute the degrees.

The degree distribution is thus very similar to the Laplacian spectrum for threshold networks. There is a shift up by 1 for large values. The shift starts at the node with weight x_i just above half the threshold $x_i > \theta/2$. This near correspondence between degree distribution and spectrum is convenient for designing networks with a given spectrum (Sec. II D) and seems to be generally true for networks which are close to, but not quite, threshold in nature.

$$
\begin{array}{c|cccccccccccccccccccccc}
 & 21 & & & & & & & & & & & & & & & & & & & & & 1 \\
 & -1 & 20 & & & & & & & & & & & & & & & & & & & & 1 \\
 & -1 & -1 & -1 & -1 & -1 & -1 & 15 & & & & & & & & & & & & & & & 1 \\
 & -1 & -1 & 1 & -1 & -1 & -1 & 15 & & & & & & & & & & & & & & & 1 \\
 & -1 & -1 & & 2 & -1 & -1 & 15 & & & & & & & & & & & & & & & 1 \\
 & -1 & -1 & & & 3 & -1 & 15 & & & & & & & & & & & & & & & 1 \\
 & -1 & -1 & & & & 4 & 15 & & & & & & & & & & & & & & & 1 \\
 & -1 & -1 & & & & & -5 & -1 & -1 & -1 & -1 & -1 & 9 & & & & & & & & & 1 \\
 & -1 & -1 & & & & & -5 & 1 & -1 & -1 & -1 & -1 & 9 & & & & & & & & & 1 \\
 & -1 & -1 & & & & & -5 & & 2 & -1 & -1 & -1 & 9 & & & & & & & & & 1 \\
 & -1 & -1 & & & & & -5 & & & 3 & -1 & -1 & 9 & & & & & & & & & 1 \\
 & -1 & -1 & & & & & -5 & & & & 4 & -1 & 9 & & & & & & & & & 1 \\
 & -1 & -1 & & & & & -5 & & & & & 5 & 9 & & & & & & & & & 1 \\
 & -1 & -1 & & & & & -5 & & & & & & -6 & -1 & -1 & 6 & & & & & & 1 \\
 & -1 & -1 & & & & & -5 & & & & & & -6 & 1 & -1 & 6 & & & & & & 1 \\
 & -1 & -1 & & & & & -5 & & & & & & -6 & & 2 & 6 & & & & & & 1 \\
 & -1 & -1 & & & & & -5 & & & & & & -6 & & & -3 & -1 & -1 & -1 & 2 & & 1 \\
 & -1 & -1 & & & & & -5 & & & & & & -6 & & & -3 & 1 & -1 & -1 & 2 & & 1 \\
 & -1 & -1 & & & & & -5 & & & & & & -6 & & & -3 & & 2 & -1 & 2 & & 1 \\
 & -1 & -1 & & & & & -5 & & & & & & -6 & & & -3 & & & 3 & 2 & & 1 \\
 & -1 & -1 & & & & & -5 & & & & & & -6 & & & -3 & & & & -4 & -1 & 1 \\
 & -1 & -1 & & & & & -5 & & & & & & -6 & & & -3 & & & & -4 & 1 & 1 \\
\hline
\lambda & 22 & 1 & & & & & 21 & & & & & & 6 & & & 15 & & & & 9 & 11 & 0 \\
\text{deg} & 21 & 1 & & & & & 20 & & & & & & 6 & & & 14 & & & & 9 & 10 & 10 \\
D/I & D & I & & & & & D & & & & & & I & & & D & & & & I & D & D
\end{array}
$$

FIG. 6. The matrix of column eigenvectors of the Laplacian for the network shown in Fig. 2. Blank entries indicate zero value. Boxes have been placed around the entries describing groups of identical nodes. Eigenvalues for each eigenvector appear at the bottom of each column. Below them are the degree and node type for nodes in the group identified by the eigenvector. Notice that dominating nodes have degree one less than the corresponding eigenvalue while for isolated nodes the spectrum and degree are equal.

C. Synchronization

Synchronization of networks of oscillators has been studied in various contexts [31–36] with the basic framework described in Ref. [32]. A common intuition is that networks with small diameter or small average path length should be easier to synchronize [31,32,35,36]. Threshold networks provide an excellent counterexample: we can easily construct networks that are arbitrarily hard to synchronize and have a diameter of 2. Moreover, threshold networks show that resistance to synchrony can vary greatly with fixed average path length and is not related to maximal betweenness centrality. It is related to the heterogeneity of degree through the range of degree values though not the variance of degree.

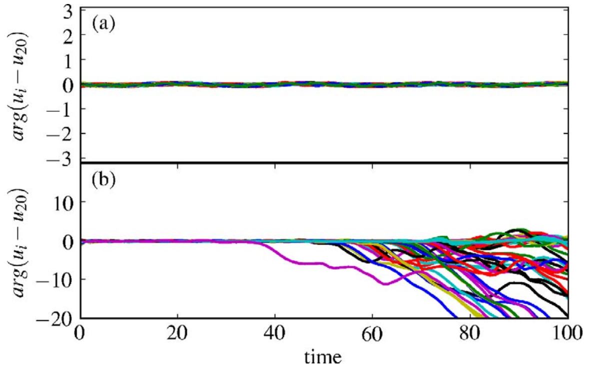

FIG. 7. (Color online) The phase difference of coupled oscillators vs time for a network with the dynamics given by $u_i = -u_i + \sigma \Sigma_{j=1}^{N} L_{ij}(u_j^3 - u_j^5/2)$ and $\sigma = 0.01855$. The system is started from an initially synchronized state with a small random perturbation applied to all nodes. The instantaneous phase is estimated using a Hilbert transform of the time signals [39]. (a) A 100-node threshold model with weights randomly selected from a power-law distribution with exponent $\gamma = -2.5$ and with threshold $\theta = 3$. The eigenvalue ratio is $r = \lambda_N / \lambda_2 = 100/33 \approx 3.0$. (b) The same network as (a) but with three nodes added so that the creation sequence ends in 101. This change makes the synchronized state unstable. The eigenvalue ratio in (b) is $r = \lambda_N / \lambda_2 = 103/1 = 103$.

Consider a system of N identical oscillators with state vector field $u(t)$ where solutions with $u_i(t) = u_j(t)$ for all nodes i and j are defined as synchronized. The system is said to be synchronizable if a synchronized solution is linearly stable to nonuniform perturbations. A standard linear analysis near the synchronized solution shows that for general oscillators with diffusive coupling the stability of the synchronized state is determined by the largest Lyapunov exponent $\Gamma(\lambda)$, also called the master stability function [32,37,38]. If $\Gamma(\lambda_i) < 0$ for each $i \geqslant 2$, the synchronized state is linearly stable. (The eigenvalue $\lambda_1 = 0$ corresponds to spatially uniform perturbations.)

For many oscillatory systems the master equation is negative only in a single interval $[\alpha_1, \alpha_2]$ determined by the type of oscillator and strength of coupling. This implies that the network is synchronizable only when the ratio $r \equiv \lambda_N / \lambda_2 < \alpha_2 / \alpha_1$ [32]. Thus if λ_2 and λ_N are inside this interval—i.e., $r < \alpha_2 / \alpha_1$—then network synchronization is stable. Construction of a synchronizable network is easier for small r. To make a network resistant to synchronization, we design the connectivity so that r, the resistance to synchrony, is large.

The explicit representation of the spectrum of threshold networks allows us to design networks with large r. Any threshold network with N nodes and creation sequence ending in 101 has diameter two and $r = N$. Figure 7 shows the effect of adding 101 to the end of the creation sequence for a random 100-node network created with the threshold model. This change increases the resistance to synchrony. In both cases, the nodes are initially only slightly perturbed from the synchronized state. That state is clearly unstable when the network is made resistant to synchronization. This small change in network structure is enough to push the system over the stability boundary for the synchronized state. Thus our construction produces a network that is arbitrarily hard to synchronize despite a small diameter.

More generally, the resistance to synchrony, r, can be derived for threshold networks explicitly. The largest eigenvalue λ_N for a connected threshold network is the number of nodes, N. The eigenvalue λ_2 is the minimal degree in the network. That is, $\lambda_2 = D_n$ where n is the number of layers so that D_n is the number of dominating nodes in the top layer. Combining, we obtain $r = N / D_n$. Notice that this ratio does not depend on most of the creation sequence.

We can design networks with prescribed resistance to synchrony without limiting many other structure measures such as the average path length, maximal betweenness centrality, or degree variance. For example, consider maximal betweenness centrality and connected threshold networks with three node groups (two layers) and $C = (D_1, I_1, D_2)$. The algorithm described in Sec. III shows that maximal betweenness centrality is attained by dominating nodes in the top layer and is equal to $B_{max} = I_1(2D_1 + I_1 - 1)/D_2$. Fixing the number of nodes, N, and the resistance r forces a fixed value for $D_2 = N/r$, but we are free to shift nodes between the first two groups, changing I_1 and D_1 to customize B_{max}. Thus r and B_{max} are independent for large networks. In practice, we can attain any maximal betweenness centrality with prescribed resistance to synchrony. Conversely, we can prescribe B_{max} and design large networks which attain any value of r.

Similar construction constraints allow r to be independent of the average path length and heterogeneity of degree as measured by the variance. The variance of degree can be designed using primarily the nodes in the lower layers of the layer cake, while r is determined by the top layer (and the total number of nodes).

One measure of degree homogeneity which *is* directly related to the resistance to synchrony for threshold networks is the range of degrees. This is because $k_{max} + 1 = N$ and $k_{min} = D_n$ so that $r = (k_{max} + 1)/k_{min}$. This simple relationship shows that once the interval of allowed degrees is established, r has been determined. We have thus derived a direct relationship between the heterogeneity of degree as measured by the range of degree and resistance to synchrony for threshold networks. One can increase the resistance to synchrony by either increasing the degree of the highest degree node or decreasing the degree of the lowest degree node. A narrow range implies ease of synchrony.

Other applications of spectral design using threshold networks include graph partitioning [40] and generic diffusively coupled systems governed by reaction diffusion equations or wave equations. Variations between states of low-degree nodes and their neighbors are associated with low eigenvalues and so are slowest to decay in diffusive systems and oscillate the slowest in wave settings. This agrees with our intuition since the low-degree nodes have least coupling with other nodes. The fact that the eigenvalues are precisely the degrees for the low-degree nodes confirms this intuition. Less intuitive, perhaps, is the result that eigenvalues are the same for eigenvectors identifying a group and eigenvectors showing variation within that group. Thus, variations within a group decay at the same rate as variations between that group and other neighboring nodes.

IV. SUMMARY

The threshold model for network creation is one of many models used to generate networks of arbitrary size with an approximate local properties such as degree distribution, clustering, degree correlation, or spectrum. We summarize previous work on networks created by the threshold model and present an alternative deterministic model for threshold network creation which approximates a prescribed degree distribution or spectrum. In either case, the created threshold networks are graph-theoretic threshold graphs, a fact that imposes a very specific network structure. We use this structure to develop a methodology for compact storage and fast computation of many network properties. The degree distribution, clustering, betweenness centrality, and Laplacian spectrum can all be computed in linear time. In addition, the Laplacian spectrum and eigenvector structure are completely characterized, allowing these networks to be created with customized spectrum. Algorithms for the generation, storage, and structural analysis presented here are contained in the authors' open-source software package NetworkX [41].

Building on this base, we have described some implications for the study of synchronization of diffusively coupled oscillators. Diffusive spreading occurs most quickly through high degree nodes, with spread around a clique being no faster than spread to those outside the clique [42,43]. Synchronization is described in terms of the spectrum of the network. Threshold networks provide constructive counterexamples to the notion that networks with small diameter are easy to synchronize. We also derived the result for threshold networks that resistance to synchrony is completely determined by the minimum and maximum degrees in the network.

The existence of fast algorithms for structural analysis suggests that threshold networks are good candidates for network deconstruction. That is, rather than analyzing an entire network at once, we might consider the important threshold networks embedded in a larger network and how they are connected. Storing and manipulating this reduced network may be more effective than working with the original network for some tasks. The network motif literature (see, e.g., [1]) deconstructs large networks using subnetworks with small numbers of nodes. By identifying small structures that occur more often than expected they attempt to identify structures with useful features in the network. Using threshold networks as the motif structures may have advantages over small subnetworks because threshold networks are arbitrarily large and yet are still computationally manageable.

Threshold networks are also good candidates for constructing nonthreshold networks with specified structure. The algorithm might consist of creating many threshold networks with desired properties and then connecting them in ways that do not significantly alter those properties. The ability to create networks by connecting subnetworks with given structure could provide great flexibility in network design.

ACKNOWLEDGMENTS

This work was carried out under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396, partially supported by the Laboratory Directed Research and Development Program, and the DOE Office of Science Advanced Scientific Computing Research (ASCR) Program in Applied Mathematics Research.

[1] E. Yeger-Lotem, S. Sattath, N. Kashtan, S. Itzkovitz, R. Milo, R. Y. Pinter, U. Alon, and H. Margalit, Proc. Natl. Acad. Sci. U.S.A. **101**, 5934 (2004).

[2] M. E. J. Newman, Phys. Rev. E **66**, 016128 (2002).

[3] S. Eubank, H. Guclu, V. S. A. Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang, Nature (London) **429**, 180 (2004).

[4] J. C. Doyle, D. L. Anderson, L. Li, S. Low, M. Roughan, S. Shalunov, R. Tanaka, and W. Willinger, Proc. Natl. Acad. Sci. U.S.A. **102**, 14497 (2005).

[5] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **99**, 7821 (2002).

[6] A. L. Barabási and R. Albert, Science **286**, 509 (1999).

[7] D. J. Watts and S. H. Strogatz, Nature (London) **393**, 440 (1998).

[8] M. E. J. Newman, SIAM Rev. **45**, 167 (2003).

[9] S. N. Dorogovtsev and J. F. F. Mendes, Adv. Phys. **51**, 1079 (2002).

[10] R. Albert and A. L. Barabási, Rev. Mod. Phys. **74**, 47 (2002).

[11] *Complex Networks*, edited by E. Ben-Naim, H. Frauenfelder, and Z. Torozckai, Vol. 650 of Lecture Notes in Physics (Springer, Berlin, 2004).

[12] G. Bianconi and A. L. Barabási, Europhys. Lett. **54**, 436 (2001).

[13] B. Bollobás, *Random Graphs*, 2nd ed. (Cambridge University Press, Cambridge, England, 2001).

[14] F. R. K. Chung, L. Lu, and V. Vu, Internet Mathematics **1**, 257 (2004).

[15] K.-I. Goh, B. Kahng, and D. Kim, Phys. Rev. Lett. **87**, 278701 (2001).

[16] G. Caldarelli, A. Capocci, P. De Los Rios, and M. A. Muñoz, Phys. Rev. Lett. **89**, 258702 (2002).

[17] V. D. P. Servedio, G. Caldarelli, and P. Buttà, Phys. Rev. E **70**, 056126 (2004).

[18] B. Söderberg, Phys. Rev. E **66**, 066121 (2002).

[19] M. Boguña and R. Pastor-Satorras, Phys. Rev. E **68**, 036112 (2003).

[20] M. Catanzaro and R. Pastor-Satorras, Eur. Phys. J. B **44**, 241 (2005).

[21] N. Masuda, H. Miwa, and N. Konno, Phys. Rev. E **70**, 036124 (2004).

[22] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs* (Academic Press, New York, 1980).

[23] N. V. R. Mahadev and U. N. Peled, *Threshold Graphs and*

Related Topics, Vol. 56 of Annals of Discrete Mathematics (Elsevier, New York, 1995).

[24] P. L. Hammer, T. Ibaraki, and B. Simeone, SIAM J. Algebraic Discrete Methods **2**, 39 (1981).

[25] R. Merris, Linear Algebr. Appl. **198**, 143 (1994).

[26] R. Merris, Eur. J. Comb. **24**, 413 (2003).

[27] M. Golumbic and A. Trenk, *Tolerance Graphs* (Cambridge University Press, Cambridge, England, 2004).

[28] R. Merris, Linear Algebr. Appl. **199**, 381 (1994).

[29] P. L. Hammer and A. K. Kelmans, Discrete Appl. Math. **65**, 255 (1996).

[30] R. Merris, Linear Algebr. Appl. **278**, 221 (1998).

[31] T. Nishikawa, A. E. Motter, Y. C. Lai, and F. C. Hoppensteadt, Phys. Rev. Lett. **91**, 014101 (2003).

[32] M. Barahona and L. M. Pecora, Phys. Rev. Lett. **89**, 054101 (2002).

[33] H. Hong, B. J. Kim, M. Y. Choi, and H. Park, Phys. Rev. E **69**, 067105 (2004).

[34] D. U. Hwang, M. Chavez, A. Amann, and S. Boccaletti, Phys. Rev. Lett. **94**, 138701 (2005).

[35] F. M. Atay, T. Biyikoğlu, and J. Jost, IEEE Trans. Circuits Syst., I: Fundam. Theory Appl. **53**, 92 (2006).

[36] A. E. Motter, C. Zhou, and J. Kurths, Phys. Rev. E **71**, 016116 (2005).

[37] L. M. Pecora and T. L. Carroll, Phys. Rev. Lett. **80**, 2109 (1998).

[38] K. S. Fink, G. Johnson, T. Carroll, D. Mar, and L. Pecora, Phys. Rev. E **61**, 5080 (2000).

[39] M. G. Rosenblum, A. S. Pikovsky, and J. Kurths, Phys. Rev. Lett. **76**, 1804 (1996).

[40] A. Arenas, A. Díaz-Guilera, and C. J. Pérez-Vicente, Phys. Rev. Lett. **96**, 114102 (2006).

[41] A. Hagberg, D. A. Schult, and P. J. Swart, https://networkx.lanl.gov/.

[42] M. Barthélemy, A. Barrat, R. Pastor-Satorras, and A. Vespignani, Phys. Rev. Lett. **92**, 178701 (2004).

[43] M. Barthélemy, A. Barrat, R. Pastor-Satorras, and A. Vespignani, J. Theor. Biol. **235**, 275 (2005).