Range-limited centrality measures in complex networks

Mária Ercsey-Ravasz,[1,2,*] Ryan N. Lichtenwalter,[2,3] Nitesh V. Chawla,[2,3] and Zoltán Toroczkai[2,3,4,†]
[1]*Faculty of Physics, Babeş-Bolyai University, Kogalniceanu street 1, RO-400084 Cluj-Napoca, Romania*
[2]*Interdisciplinary Center for Network Science and Applications (iCeNSA), University of Notre Dame, Notre Dame, Indiana 46556, USA*
[3]*Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, Indiana 46556, USA*
[4]*Department of Physics, University of Notre Dame, Notre Dame, Indiana 46556, USA*
(Received 22 November 2011; published 6 June 2012)

Here we present a range-limited approach to centrality measures in both nonweighted and weighted directed complex networks. We introduce an efficient method that generates for every node and every edge its betweenness centrality based on shortest paths of lengths not longer than $\ell = 1, \ldots, L$ in the case of nonweighted networks, and for weighted networks the corresponding quantities based on minimum weight paths with path weights not larger than $w_\ell = \ell\Delta$, $\ell = 1,2\ldots, L = R/\Delta$. These measures provide a systematic description on the positioning importance of a node (edge) with respect to its network neighborhoods one step out, two steps out, etc., up to and including the whole network. They are more informative than traditional centrality measures, as network transport typically happens on all length scales, from transport to nearest neighbors to the farthest reaches of the network. We show that range-limited centralities obey universal scaling laws for large nonweighted networks. As the computation of traditional centrality measures is costly, this scaling behavior can be exploited to efficiently estimate centralities of nodes and edges for all ranges, including the traditional ones. The scaling behavior can also be exploited to show that the ranking top list of nodes (edges) based on their range-limited centralities quickly freezes as a function of the range, and hence the diameter-range top list can be efficiently predicted. We also show how to estimate the typical largest node-to-node distance for a network of N nodes, exploiting the afore-mentioned scaling behavior. These observations were made on model networks and on a large social network inferred from cell-phone trace logs ($\sim 5.5 \times 10^6$ nodes and $\sim 2.7 \times 10^7$ edges). Finally, we apply these concepts to efficiently detect the vulnerability backbone of a network (defined as the smallest percolating cluster of the highest betweenness nodes and edges) and illustrate the importance of weight-based centrality measures in weighted networks in detecting such backbones.

DOI: 10.1103/PhysRevE.85.066103 PACS number(s): 89.75.Hc, 89.65.−s, 02.10.Ox

I. INTRODUCTION

Network research [1–5] has experienced an explosive growth in the last two decades, as it has proven itself to be an informative and useful methodology to study complex systems, ranging from social sciences through biology to communication infrastructures. Both the natural and man made world is abundant with networked structures that transport various entities, such as information, forces, energy, material goods, etc. As many of these networks are the result of evolutionary processes, it is important to understand how the graph structure of these systems determines their transport performance, structural stability, and behavior as a whole. A rather useful concept in addressing such questions is the notion of centrality, which describes the positioning "importance" of a structure of interest such as a node, edge, or subgraph with respect to the whole network. Although the notion of centrality in graph theory dates back to the mathematician Camille Jordan (1869), centrality measures were expanded, refined, and applied to a great extent for the first time in social sciences [6–10], and today they play a fundamental role in studies involving a large variety of complex networks across many fields. Probably the most frequently used centrality measure is betweenness centrality (BC) [10–16], introduced by Anthonisse [11] and Freeman [12] defined as the fraction of all network geodesics (shortest paths) passing through a node (edge or subgraph). Since transport tends to minimize the cost or time of the route from source to destination, it expectedly happens along geodesics, and therefore centrality measures are typically defined as a function of these, however generalizations to arbitrary distributions of transport paths have also been introduced and studied [17,18]. Geodesics are important for structural connectivity as well: removing nodes (edges) with high BC, one obtains a rapid increase in diameter, and eventually the structural breakup of the graph.

In general, centrality measures are defined in the context of the assumptions (sometimes made implicitly) regarding the type of network flow [16]. These are assumptions regarding the *nature of the paths* such as being shortest, or arbitrary length paths, weighted or valued paths, walks (repeated nodes and edges) [19], etc., and the *nature of the flow*, such as transport of indivisible units (packets), or spreading or broadcasting processes (infection, information). Besides betweenness centrality, many other centrality measures have been introduced [10], depending on the context in which network flows are considered; for a partial compilation see the paper by Brandes [14]; here we only review a limited list. In particular, *stress* centrality [20–22], simply counts the number of all-pair shortest paths passing through a node (edge) without taking into account the degeneracy of the geodesics (there can be several geodesics running between the same pair of nodes). *Closeness* centrality [8,13,23] and its variants are simple functions of the mean geodesic distance

*mercseyr@nd.edu
†toro@nd.edu

(hop count) of a node from all other nodes. *Load* centrality [14,24,25] is generated by the total amount of load passing through a node when unit commodities are passed between all source-destination pairs using an algorithm in which the commodity packet is equally divided among the neighbors of a node that are at the same geodesic distance from the destination. *Group* betweenness centrality [26,27] computes the betweenness associated with a set of nodes restricted to all-pair geodesics that traverse at least one of the nodes in the group. *Ego* network betweenness [28] is a local betweenness measure computed only from the immediate neighborhood of a node (ego). *Eigenvector* centrality [29,30] represents a positive score associated to a node, proportional to the sum of the scores of the node's neighbors, solved consistently across the graph. The corresponding score vector is the eigenvector associated with the largest eigenvalue of the adjacency matrix. *Random walk* centrality [31,32] is a measure of the accessibility of a node via random walks in the network. Other centrality-type measures include *information* centrality of Stephenson and Zelen [33], the *induced endogenous and exogenous* centrality by Everett and Borgatti [34], and the notion of *accessibility* pioneered by Costa *et al.* [35–37].

Bounded-distance betweenness was introduced by Borgatti and Everett [10] as betweenness centrality resulting from all-pair shortest paths not longer than a given length (hop count). It is this measure that we expand and investigate in detail in the present paper. A condensed version for unweighted networks has been presented in Ref. [38]. Since we are also generalizing the measure and the corresponding algorithm to weighted (valued) networks, we are referring to it as *range-limited* centrality. Note that range limitation can be imposed on all centrality measures that depend on paths, and therefore the analysis and algorithm presented here can be extended to all these centrality measures.

Centrality measures have received numerous applications in several areas. In social sciences they have been extensively used to quantify the position of individuals with respect to the rest of the network in various social network data sets [6,16]. In physics and computer science they have seen widespread applications related to routing algorithms in packet switched communication networks and transport problems in general [17,24,39–44]. The connection of generalized betweenness centrality based on arbitrary path distributions (not just shortest) to routing that minimizes congestion has been investigated by Sreenivasan *et al.* [17] using minimum sparsity vertex separators. This makes a direct connection to max-flow min-cut theorems of multicommodity flows, extensively studied in the computer science literature [45,46]. Other works that use essentially edge betweenness type quantities to quantify congestion in Internet-like graphs include Refs. [47,48]. Dall'Asta *et al.* connect node and edge detection probabilities in a trace-route-based sampling of networks to their betweenness centrality values [49,50]. Other applications include detection of network vulnerabilities in the face of attacks [51], cascading failures [52–54], or epidemics [55], all involving betweenness-related calculations.

An important extension of centrality is to weighted, or valued, networks [25,56–60]. In this case the edges (and also the nodes) carry an associated weight, which may represent a measure of social relationship in social networks [61], channel capacity in the case of communication networks, transport capacity (e.g., nr of lanes) in roadway networks or seats on flights [62].

From a theory point of view, there have been fewer results, as producing analytic expressions for centralities in networks is difficult in general. However, for scale-free trees, Szabó *et al.* [63] developed a mean-field approach for computing node betweenness, which later was made rigorous by Bollobás and Riordan [64]. Fekete *et al.* provide a calculation of the distribution of edge betweenness on scale-free trees conditional on node in-degrees [65], and Kitsak *et al.* [66] have derived scaling results on betweenness centrality for fractal and nonfractal scale-free networks.

Unfortunately, computation of betweenness can be costly [$\mathcal{O}(NM)$, where N is the number of nodes and M is the number of edges, thus $\mathcal{O}(N^3)$ is the worst case] [14,25,57,67–69], especially for large networks with millions of nodes, hence approximation methods are needed. Existing approximations [32,70,71], however, are sampling based, and not well controlled. Additionally, transport in real networks does not occur with uniform probability between arbitrary pairs of nodes, as transport incurs a cost, and therefore shorter-range transport is expectedly more frequent than long-range transport. Accordingly, the *usage* of network paths is nonuniform, which should be taken into account if we want to connect centrality properties with real transport. In order to address some of the limitations of existing centrality measures, we recently focused on range-limited centrality [38]. We have shown that when geodesics are restricted to a maximum length L, the corresponding range-limited L-betweenness for large graphs assumes a characteristic scaling form as a function of L. This scaling can then be used to predict the betweenness distribution in the (difficult to attain) diameter limit, and with good approximation, to predict the ranking of nodes/edges by betweenness, saving considerable computational costs. Additionally, the range-limited method generates l-betweenness values *for all* nodes and edges and *for all* $1 \leqslant l \leqslant L$, providing systematic information on geodesics on all length scales.

In this paper we give a detailed derivation of the algorithm and the analytical approximations presented in [38] and we demonstrate the efficiency of the method on a social network (SocNet) inferred from mobile phone trace logs [72]. This network has a giant cluster with $N = 5\,568\,785$ nodes and $M = 26\,822\,764$ directed edges. The diameter of the underlying undirected network is approximately $D \simeq 26$ and the calculation of the traditional (diameter-range based) BC values (using Brandes' algorithm) on this network took 5 days on 562 computers.

In addition, we present the derivations for an algorithm that efficiently computes range-limited centralities on *weighted* networks. We then apply these concepts and algorithms to the network vulnerability backbone detection problem, and show the differences between the backbones obtained with both hop-count based centralities and weighted centralities.

The paper is organized as follows. Section II introduces the notations and provides the algorithm for unweighted graphs; Sec. III gives an analytical treatment that derives the existence of a scaling behavior for centrality measures

in large graphs; it gives a method on how to estimate the largest typical node-to-node distance (a lower bound to the diameter); discusses the complexity of the algorithm and the fast freezing phenomenon of ranking by betweenness of nodes and edges. Section IV illustrates the power of the range-limited approach (by showing how well can one predict betweenness centralities and ranking of individual nodes and edges) using the social-network data described above. Section V describes the algorithm for weighted graphs and Sec. VI uses the range-limited betweenness measure to define a vulnerability backbone for networks and illustrates the differences in identification of the backbone obtained with and without weights on the links.

II. RANGE-LIMITED CENTRALITY FOR NONWEIGHTED GRAPHS

A. Definitions and notations

Let us consider a directed simple graph $\boldsymbol{G}(V,\boldsymbol{E})$, which consists of a set V of vertices (or nodes) and a set $E \subseteq V \times V$ of directed edges (or links). We will denote by $(v_i,v_j) \in E$ an edge directed from node $v_i \in V$ to node $v_j \in V$. The graph has N nodes and $M \leqslant N(N-1)$ edges. The algorithm below can easily be modified for undirected graphs, we will not treat that case separately. A directed path ω_{mn} from some node m to a node n is defined as an ordered sequence of nodes and links $\omega_{mn} = \{m,(m,v_1),v_1,(v_1,v_2),v_2,\ldots v_l,(v_l,n),n\}$ without repeated nodes. The "distance" $d(m,n)$ is the length of the shortest *directed* path going from node m to node n. We give a definition of distance (path weight) for weighted networks in Sec. V. In nonweighted networks the directed path length is simply the number of edges ("hop count") along the directed path from m to n. There can be multiple shortest paths (same length), and we will denote by σ_{mn} the total number of shortest directed paths from node m to n. $\sigma_{mn}(i)$ will represent the number of shortest paths from node m to node n *going through* node i. As convention we set

$$\sigma_{mn}(m) = \sigma_{mn}(n) = \sigma_{mn}, \quad \sigma_{mm}(i) = \delta_{i,m}. \tag{1}$$

The total number of all-pair shortest paths running through a node i is called the stress centrality (SC) of node i, $S(i) = \sum_{m,n\in V} \sigma_{mn}(i)$. Betweenness centrality (BC) [10–12,14,57] normalizes the number of paths through a node by the total number of paths (σ_{mn}) for a given source-destination pair (m,n):

$$B(i) = \sum_{m,n\in V} \frac{\sigma_{mn}(i)}{\sigma_{mn}}. \tag{2}$$

Similar quantities can be defined for an edge $(j,k) \in \boldsymbol{E}$:

$$B(j,k) = \sum_{m,n\in V} \frac{\sigma_{mn}(j,k)}{\sigma_{mn}}. \tag{3}$$

In order to define range-limited betweenness centralities, let $b_l(j)$ denote the BC of a node j for all-pair shortest directed paths of *fixed, exact* length l. Then

$$B_L(j) = \sum_{l=1}^{L} b_l(j) \tag{4}$$

represents the betweenness centrality obtained from paths *not longer* than L. For edges, we introduce $b_l(j,k)$ and $B_L(j,k)$ using the same definitions. For simplicity, here we include the start and end points of the paths in the centrality measures, however, our algorithm can easily be changed to exclude them, as described later.

Similar to other algorithms, our method first calculates these BCs for a node j [or edge (j,k)] from shortest directed paths all emanating from a "root" node i, then it sums the obtained values for all $i \in V$ to get the final centralities for node j [or edge (j,k)]. This can be done because the set of *all* shortest paths can be uniquely decomposed into subsets of shortest paths distinguished by their starting node. Thus it makes sense to perform a shell decomposition of the graph around a root node i [73–77]. Let us denote by $\boldsymbol{C}_L(i)$ the L-range subgraph of node i containing all nodes which can be reached in at most L steps from i [Fig. 1(a)]. Only links which are part of the shortest paths starting from the root i to these nodes are included in $\boldsymbol{C}_L$. We decompose $\boldsymbol{C}_L$ into shells $G_l(i)$ containing all the nodes at shortest path distance l from the root, and all incoming edges from shell $l-1$ [Fig. 1(b)]. The root i itself is considered to be shell 0 [$G_0(i) = \{i\}$]. Let

$$b_l^r(i|k) = \sum_{n\in G_l(i)} \frac{\sigma_{in}(k)}{\sigma_{in}}, \quad b_l^r(i|j,k) = \sum_{n\in G_l(i)} \frac{\sigma_{in}(j,k)}{\sigma_{in}} \tag{5}$$

denote the fixed-l-betweenness centrality of node k, and edge (j,k), respectively, based only on shortest paths all starting from the root i. Here r is *not* an independent variable: given i and k [or (j,k)], r is the radius of shell $G_r(i)$ containing k [or (j,k)], that is $k \in G_r(i)$ and $(j,k) \in G_r(i)$. Note that $\sigma_{in}(k) = 0$ [or $\sigma_{in}(j,k) = 0$] if k [or (j,k)] do not belong to at least one shortest path from i to n, and thus there is no contribution from those points n from the lth shell. The condition for k [or (j,k)] to belong to at least one shortest path from i to n can alternatively be written in the case of (5) as $d(k,n) = l - r$, a notation which we will use later.

For simplicity of writing, we refer to the fixed-l-betweenness centralities (the b_ls) as "l-BCs" and to the cumulative betweenness centralities [the B_Ls obtained from summing the l-BCs; see (4)] as [L]-BCs.

B. Range-limited betweenness centrality algorithm

While the basics of our algorithm are similar to Brandes' [14,57], we derive recursions that simultaneously compute the [l]-BCs for *all* nodes and *all* edges and for all values $l = 1,\ldots,L$. The algorithm thus generates detailed and systematic information (an L-component *vector* for every node and every edge) about shortest paths on all length scales and thus provides a tool for multiscale network analysis.

First we give the algorithm, then we derive the specific recursions used in it. For the root node i we set the initial condition: $\sigma_{ii} = 1$. For other nodes, $k \neq i$, we set $\sigma_{ik} = 0$. The following steps are repeated for every $l = 1,\ldots,L$:

(1) Build $G_l(i)$, using breadth-first search.

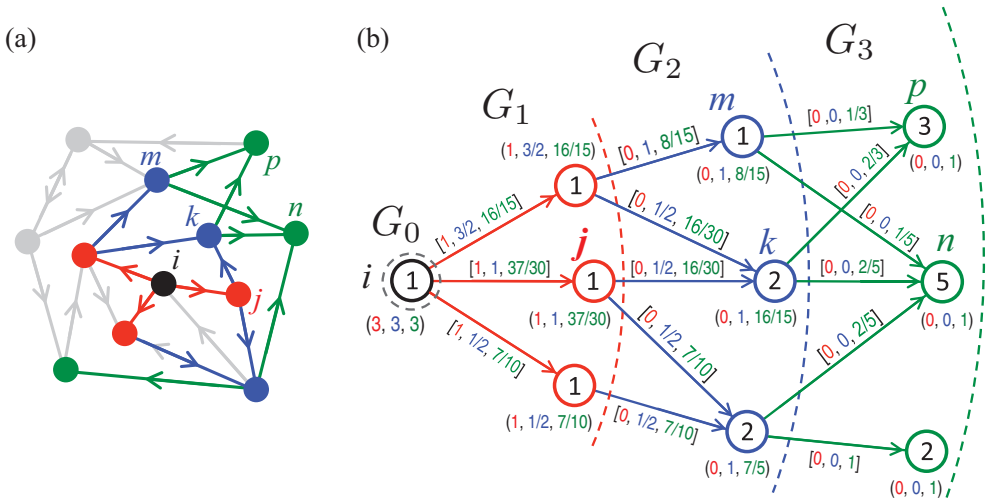


FIG. 1. (Color) (a) Consecutive shells of the $\boldsymbol{C}_3$ subgraph of node i (black) are colored red, blue, green. Grey elements are not part of the subgraph. (b) The (x,y,z) near a node j are the $b_\ell(i|j)$ values for $\ell=1$, $\ell=2$, and $\ell=3$. The $[x,y,z]$ on an edge (j,k) give the $b_\ell(i|j,k)$ values for $\ell=1$, $\ell=2$, and $\ell=3$. Given a node j, the number inside its circle is the total number of shortest paths σ_{ij} to j from i. Colors indicate quantities based on $\ell=1$ (red), $\ell=2$ (blue), and $\ell=3$ (green).

(2) Calculate σ_{ik} for all nodes $k\in G_l(i)$, using

$$\sigma_{ik}=\sum_{\substack{j\in G_{l-1}(i)\\(j,k)\in G_l(i)}}\sigma_{ij}, \tag{6}$$

and set

$$b_l^l(i|k)=1. \tag{7}$$

(3) Proceeding *backward*, through $r=l-1,\ldots,1,0$,

(a) Calculate the l-BCs of links $(j,k)\in G_{r+1}(i)$ [thus $j\in G_r(i)$, $k\in G_{r+1}(i)$] recursively:

$$b_l^{r+1}(i|j,k)=b_l^{r+1}(i|k)\frac{\sigma_{ij}}{\sigma_{ik}}, \tag{8}$$

(b) and of nodes $j\in G_r(i)$ using (8) and

$$b_l^r(i|j)=\sum_{\substack{k\in G_{r+1}(i)\\(j,k)\in G_{r+1}(i)}}b_l^{r+1}(i|j,k). \tag{9}$$

(4) Finally, return to step 1 until the last shell $G_L(i)$ is reached.

In the end, the cumulative $[l]$-BCs, that is the B_ls, can be calculated using (4). Figure 1 shows a concrete example. The subgraph of node i has three layers. Each layer $G_l(i)$ and the corresponding l-BCs are marked with different colors: $l=1$ (red), $l=2$ (blue), and $l=3$ (green). As described above, the first step creates the next layer $G_l(i)$, then in step 2, for every node $k\in G_l(i)$ we calculate the total number of shortest paths σ_{ik} from the root to node k. These are indicated by numbers within the circles representing the nodes in Fig. 1 (e.g., $\sigma_{ij}=1$, $\sigma_{ik}=2$, $\sigma_{in}=5$). As given by (6), σ_{ik} is calculated by summing the number of shortest paths that end in the predecessors of node k located in $G_{l-1}(i)$. For example, node $p\in G_3(i)$ in Fig. 1 is connected to nodes k and m in shell $G_2(i)$, and thus $\sigma_{ip}=\sigma_{ik}+\sigma_{im}=2+1=3$.

Equation (7) states that the l-BC of nodes located in $G_l(i)$ is always 1. This follows from Eq. (5) for $r=l$ and using the convention $\sigma_{ik}(k)=\sigma_{ik}$. Knowing these values, we proceed backward (step 3) and calculate the l-BCs of all edges and nodes in all the previous layers. Recursion (8) is obtained from a well known recursion for shortest paths. If k [or (j,k)] belongs to at least one shortest path going from i to n, then $\sigma_{in}(k)=\sigma_{ik}\sigma_{kn}$ and $\sigma_{in}(j,k)=\sigma_{ij}\sigma_{kn}$. Inserting these in Eq. (5) for $r\mapsto r+1$ we obtain

$$b_l^{r+1}(i|k)=\sigma_{ik}\sum_{\substack{n\in G_l(i)\\d(k,n)=l-r-1}}\frac{\sigma_{kn}}{\sigma_{in}}, \tag{10}$$

$$b_l^{r+1}(i|j,k)=\sigma_{ij}\sum_{\substack{n\in G_l(i)\\d(k,n)=l-r-1}}\frac{\sigma_{kn}}{\sigma_{in}}, \tag{11}$$

where $d(k,n)=l-r-1$ expresses the condition that the sum is restricted to those n from $G_l(i)$, which have at least one shortest path (from i), going through k or (j,k). Dividing these equations we obtain (8). For, e.g., in Fig. 1, $b_3^3(i|k,n)=b_3^3(i|n)\sigma_{ik}/\sigma_{in}=1\times(2/5)=2/5$.

Having determined the l-BCs of all edges in layer $G_{r+1}(i)$, we can now compute the l-BC of a given node in $G_r(i)$ by summing the l-BCs of its outgoing links, that is using (9) [e.g., in Fig 1, $b_3^2(i|k)=b_3^3(i|k,p)+b_3^3(i|k,n)=(2/3)+(2/5)=16/15$].

This algorithm can be easily modified to compute other centrality measures. For example, to compute all the range-limited stress centralities, we have to replace Eq. (7) with $s_l^l(i|j)=\sigma_{ij}$. All other recursions will have exactly the same form; we just need to replace the l-BCs $[b_l^r(i|j),\ b_l^r(i|j,k)]$ with the l-SCs $[s_l^r(i|j),\ s_l^r(i|j,k)]$.

If we want to exclude start and end points when computing BCs or SCs, we first let the above algorithm finish, then we do the following steps: (a) set the l-BC of the root node i to 0, $b_l^0(i|i)=0$ for all $l=1,\ldots,L$, and (b) for every node $k\in G_l(i)$ reset $b_l^l(i|k)=0$, for all $l=1,\ldots,L$ [for, e.g., in Fig. 1 k is in the second shell, $G_2(i)$, so its 2-BC will become 0 instead of 1]. Then via (4), the $[l]$-BCs and the corresponding $[l]$-SCs are easily obtained.

III. CENTRALITY SCALING—ANALYTICAL APPROXIMATIONS

In [38] we have shown that the $[l]$-BC obeys a scaling behavior as a function of l. This was found to hold for all sufficiently large random networks that we studied [Erdős-Rényi (ER), Barabási-Albert (BA) scale-free, random geometric graphs (RGGs), etc.] including the social network inferred from mobile phone trace-log data (SocNet) [78]. Here we detail the analytical arguments, that indeed show that the existence of this scaling behavior for large networks is a general property, by exploiting the scaling of shell sizes. The scaling of shell sizes was already studied previously, for, e.g., in random graphs with arbitrary degree distributions [79,80]. For simplicity of the notation, we only show the derivations for undirected graphs.

A. Betweenness of individual nodes

Let us define $\langle\cdot\rangle$ as an average over all root nodes i in the graph, and denote by $z_l(i)$ the number of nodes on shell $G_l(i)$. We define the branching factor as

$$\alpha_l = \langle z_{l+1}\rangle/\langle z_l\rangle, \tag{12}$$

and model the growth of shell sizes as a branching process [79,81],

$$z_{l+1}(i) = z_l(i)\alpha_l[1+\epsilon_l(i)]. \tag{13}$$

Here $\epsilon_l(i)$ is a *per-node*, shell occupancy noise term, encoding the relative deviations, or fluctuations from the (i-independent) functional form of α_l. Typically, $|\epsilon_l| \ll 1$, it obeys $\langle\epsilon_l(i)\rangle = 0$ and $\langle\epsilon_l(i)\epsilon_m(j)\rangle = 2A_l\delta_{l,m}\delta_{i,j}$, with A_l decreasing with l. In undirected graphs if $i \in G_m(j)$ then it implies that $j \in G_m(i)$, and vice versa. Hence, in this case,

$$b_{l+1}(j) = \frac{1}{2}\sum_{i\in V} b_{l+1}(i|j) = \frac{1}{2}\sum_{m=0}^{l+1}\sum_{i\in G_m(j)} b^m_{l+1}(i|j). \tag{14}$$

The $1/2$ factor comes from the fact that any given path will be included twice in the sum (once in both directions). In the case of $m = 0$ the only node in $G_0(j)$ is j itself, and the inner sum is equal with $b^0_{l+1}(j|j)$. Due to convention (1) $\sigma_{jn}(j) = \sigma_{jn}$ and hence from (5) we obtain $b^0_{l+1}(j|j) = \sum_{n\in G_{l+1}(j)}\sigma_{jn}(j)/\sigma_{jn} = z_{l+1}(j)$. For $m = l+1$, $b^{l+1}_{l+1}(i|j) = 1$ [see Eq. (7)] and the inner sum is again $z_{l+1}(j)$. Thus we can write

$$\begin{aligned} b_{l+1}(j) &= z_{l+1}(j) + \frac{1}{2}\sum_{m=1}^{l}\sum_{i\in G_m(j)} b^m_{l+1}(i|j) \\ &\equiv z_{l+1}(j) + \frac{1}{2}u_{l+1}(j), \end{aligned} \tag{15}$$

Note that the number of terms in the inner sum $\sum_{i\in G_m(j)} b^m_{l+1}(i|j)$ is $z_m(j)$, which is rapidly increasing with m, and thus is expected to have a weak dependence on j. Accordingly, we make the approximation

$$u_{l+1}(j) \simeq \sum_{m=1}^{l}\sum_{i\in G_m(j)} v^m_{l+1}(i), \tag{16}$$

where we replaced $b^m_{l+1}(i|j)$ by $v^m_{l+1}(i)$, which is an *average* $(l+1)$-BC computed over the shell of radius m, *centered on node* i:

$$v^m_{l+1}(i) = \frac{\sum_{k\in G_m(i)} b^m_{l+1}(i|k)}{z_m(i)}. \tag{17}$$

However, the sum of $(l+1)$-BCs in any $m \leqslant l+1$ layer is equal to the number of nodes in shell G_{l+1}: $\sum_{k\in G_m(i)} b^m_{l+1}(i|k) = z_{l+1}(i)$. We can convince ourselves about this last statement by using (5) and observing that $\sum_{k\in G_m(i)}\sigma_{in}(k) = \sigma_{in}$ as all paths from i to n $[n \in G_{l+1}(i)]$ must "pierce" every shell $m \leqslant l+1$ in between. Figure 1 shows an example: there are three nodes in G_3 and the sum of three-betweenness values (green) in layer G_2 is $(7/5) + (16/15) + (8/15) = 3$. Therefore we may write

$$v^m_{l+1}(i) \simeq \frac{z_{l+1}(i)}{z_m(i)} = \frac{z_l(i)\alpha_l[1+\epsilon_l(i)]}{z_m(i)}, \tag{18}$$

where we used the recursion defined above for $z_{l+1}(i)$ as a branching process (13). Inserting this in (16) we obtain

$$\begin{aligned} u_{l+1}(j) &\simeq \alpha_l\sum_{m=1}^{l}\sum_{i\in G_m(j)}\frac{z_l(i)[1+\epsilon_l(i)]}{z_m(i)} \\ &\simeq \alpha_l\sum_{m=1}^{l}\sum_{i\in G_m(j)}\frac{z_l(i)}{z_m(i)} \\ &\simeq \alpha_l\left[z_l(j) + \sum_{m=1}^{l-1}\sum_{i\in G_m(j)}\frac{z_l(i)}{z_m(i)}\right], \end{aligned} \tag{19}$$

where we neglected the small noise term due to the large number of terms in the inner sum, and we used the fact that for $m = l$ the leading term of the inner sum is just $z_l(j)$. From Eqs. (16) and (18), however, the double sum in (19) equals $u_l(j)$ and we obtain the following recursion:

$$u_{l+1}(j) \simeq \alpha_l[z_l(j) + u_l(j)]. \tag{20}$$

Equations (13), (15), and (20) lead to a recursion for $b_{l+1}(j)$:

$$b_{l+1}(j) \simeq \alpha_l[b_l(j) + z_l(j)/2 + z_l(j)\epsilon_l(j)], \tag{21}$$

which can be iterated down to $l = 1$, where $b_1(j) = z_1(j) = k_j$ is the degree of j:

$$b_l(j) \simeq \beta_l\, k_j\, e^{\xi_l(j)}, \tag{22}$$

with

$$\beta_l = \frac{l+1}{2}\prod_{m=1}^{l-1}\alpha_m = \frac{l+1}{2}\frac{\langle z_l\rangle}{\langle k\rangle}, \tag{23}$$

$$\xi_l(j) = \sum_{n=1}^{l-1}\frac{l+1-n}{l+1}\,\epsilon_n(j). \tag{24}$$

In many networks, the average shell size $\langle z_l\rangle$ grows exponentially with the shell "radius" l (for, e.g., ER, BA, SocNet), implying a constant average branching factor larger than 1:

$$\alpha_l \simeq \alpha = \frac{\langle z_2\rangle}{\langle k\rangle} > 1. \tag{25}$$

The exponential growth holds until l reaches the typical largest shortest path distance L^*, beyond which finite-size effects

appear. Accordingly, $\beta_l \sim \alpha^l$ and b_l grows exponentially with l. In this case, since b_l is rapidly increasing with l, the cumulative $B_L(j) = \sum_{l=1}^{L} b_l(j)$ will be dominated by b_L, and thus B_L obeys the same exponential scaling as b_l, confirmed by numerical simulations [Fig. 3(c) in [38] shows this scaling for SocNet].

However, not all large networks have exponentially growing shell sizes. For example, in spatially embedded networks without shortcuts such as random geometric graphs, roadways, etc., average shell size grows as a *power law* $\langle z_l \rangle \sim l^{d-1}$, where d is the embedding dimension of the metric space. In this case $\beta_l \sim l^d$ and $b_l(j) \sim l^d$ and $B_L \sim L^{d+1}$. Figure 3(d) in [38] shows this scaling for RGG graphs embedded in $d = 2$ dimensions.

B. Distribution of l-betweenness centrality

Equation (22) allows us to relate the statistics of fixed-l-betweenness to the statistics of shell occupancies for networks that are uncorrelated, or short-range correlated. Since the noise term (obtained from *per-node* occupancy deviations on a shell) is independent on the root's degree in this case, the distribution of fixed-l-betweenness can be expressed as

$$\rho_l(b) = \langle \delta\left(b_l(j) - b\right)\rangle = \int_{-\infty}^{\infty} d\xi \int_1^{N-1} dk\ \delta\left(\beta_l k e^{\xi} - b\right) P(k)\Phi_l(\xi), \quad (26)$$

where $\delta(x)$ is the Dirac δ function, $P(k)$ is the degree distribution, and $\Phi_l(\xi)$ is the distribution for the noise $\xi_l(j)$, peaked at $\xi = 0$, with fast decaying tails and $\Phi_1(x) = \delta(x)$. Performing the integral over the noise ξ, one obtains the distribution for l-BC, in the form of a convolution:

$$\rho_l(b) = \frac{1}{b}\int_1^{N-1} dk\ P(k)\Phi_l(\ln b - \ln \beta_l - \ln k). \quad (27)$$

From (27) follows that the natural scaling variable for betweenness distribution is $u = \ln b - \ln \beta_l$. The noise distribution Φ_l (for $l > 1$) may introduce an extra l dependence through its width σ_l, which can be accounted for via the rescaling $u \mapsto u/\sigma_l$, $\rho_l \mapsto \rho_l \sigma_l$, thus collapsing the distributions for different l values onto the same functional form, directly supporting our numerical observations presented in Ref. [38]. As Φ_l is typically sharply peaked around 0, the most significant contribution to the integral (27) for a given b comes from degrees $k \simeq b/\beta_l$. Since $k \geqslant 1$, we have a rapid decay of $\rho_l(b)$ in the range $b < \beta_l$, a maximum at $\bar{b} = \beta_l \bar{k}$ where $\bar{k}$ is the degree at which $P(k)$ is maximum, and a sharp decay for $b > (N-1)\beta_l$.

C. Estimating the average node-to-node distance in large networks

The scaling law on its own does not provide information about the typical largest node-to-node distance, which is always a manifestation of the finiteness of the graph. However, knowing the size of the network in terms of the number of nodes N, one can exploit our formulas to find the average largest node-to-node distance as the radius L^* of the *typical largest shell* beyond which finite-size effects become strong, that is where network edge effects appear. This can be estimated as the point where the sum of the average shell sizes reaches N. Hence

$$Z_{L^*} = \sum_{l=1}^{L^*} \langle z_l \rangle = \sum_{l=1}^{L^*} \frac{2}{l+1}\langle k \rangle \beta_l = N, \quad (28)$$

providing an *implicit equation* for L^*. The β_ls are determined numerically for $l = 1,2,3,\ldots$ and a corresponding functional form fitting its scaling with l can be extrapolated for larger l values up to L^*, when the sum in (28) hits N. For our social network data one obtains $L^* \simeq 9.35$ [see Fig. 2(a)]. Here L^* is not necessarily an integer, because it is obtained from the scaling behavior of the average shell sizes, and represents the *typical* radius of the largest shell.

Expression (28) can be easily specialized for the two classes of networks discussed above, namely for those having exponential average shell-size growth $\langle z_l \rangle \sim \langle k \rangle \alpha^{l-1}$ [such as for the ER and BA models, Figs. 2(b) and 2(c)] and for those having a power-law average shell-size growth as $\langle z_l \rangle \sim \langle k \rangle l^{d-1}$. For the exponential growth case we obtain

$$L^* = \frac{1}{\ln \alpha} \ln\left(1 + \frac{\alpha - 1}{\langle k \rangle} N\right), \quad (29)$$

resulting in the $L^* \sim \ln N$ behavior for large N.

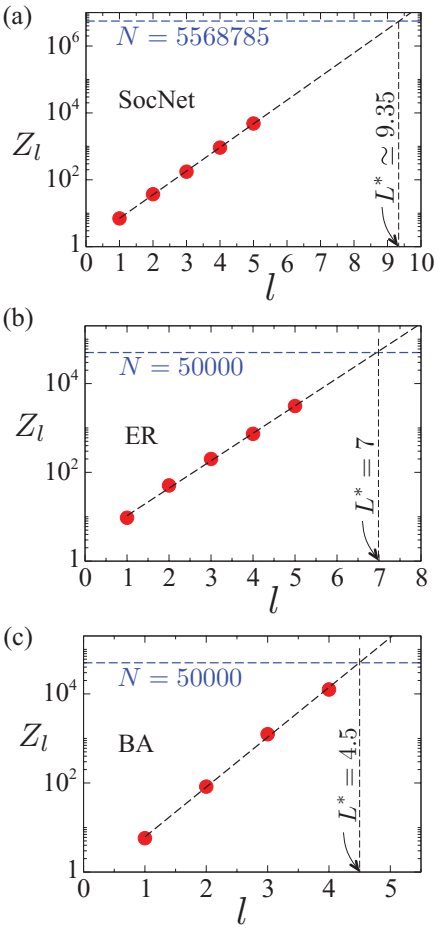


FIG. 2. (Color online) Volume Z_ℓ growth with radius ℓ. (a) Extrapolating the growth for SocNet one can estimate that it reaches the $N = 5\,568\,785$ mark at $L^* \simeq 9.35$. Similarly, there is exponential growth for both ER (b) and BA models (c).

For the power-law growth case there is no easily invertible expression for the sum, however, if we replace the summation with an integral, we find the approximate

$$L^* \simeq \left(1 + \frac{d}{\langle k \rangle} N\right)^{1/d} \tag{30}$$

expression, with the expected asymptotic behavior $L^* \sim N^{1/d}$ as $N \to \infty$.

D. Algorithm complexity

We are now in position to estimate the average-case complexity of the range-limited centrality algorithm. For every root i, we sequentially build its $l = 1,2,\ldots,L$ shells. When going from shell $G_{l-1}(i)$ to building shell $G_l(i)$, we consider all the z_{l-1} nodes on $G_{l-1}(i)$. For every such node j we add all its links that do not connect to already tagged nodes [a tag labels a node that belongs to $G_{l-1}(i)$ or $G_{l-2}(i)$] to $G_l(i)$, and add the corresponding nodes as well. This requires on the order of $\langle k \rangle$ operations for every node j, hence on the order of $\langle k \rangle \langle z_{l-1} \rangle$ operations for creating shell $G_l(i)$. Next is Eq. (6), which involves $\langle e_l \rangle$ steps, where e_l is the number of edges connecting nodes in shell $G_{l-1}(i)$ to nodes in shell $G_l(i)$. Equation (7) involves $\langle z_l \rangle$ steps. Equations (8) and (9) generate a total of $2\sum_{m=1}^{l} \langle e_m \rangle$ operations. Hence for a given l there are a total of $\langle k \rangle \langle z_{l-1} \rangle + \langle e_l \rangle + \langle z_l \rangle + 2\sum_{m=1}^{l} \langle e_m \rangle$ operations on average. Thus the average complexity of the algorithm $\mathcal{C}$ can be estimated as

$$\mathcal{C} \sim N \sum_{l=1}^{L} \left(\langle k \rangle \langle z_{l-1} \rangle + \langle e_l \rangle + \langle z_l \rangle + 2 \sum_{m=1}^{l} \langle e_m \rangle \right). \tag{31}$$

Note that the set of edges in the shells $G_{m-1}(i)$ and $G_m(i)$ are all fanning out from nodes in $G_{m-1}(i)$, and thus we can approximate $\langle e_{m-1} \rangle + \langle e_m \rangle$ with $\langle k \rangle \langle z_{m-1} \rangle$. Thus the estimate becomes

$$\mathcal{C} \sim N \langle k \rangle \sum_{l=1}^{L} \sum_{m=1}^{l} \langle z_m \rangle = N \langle k \rangle \sum_{l=1}^{L} (L - l + 1) \langle z_l \rangle. \tag{32}$$

From (32) it follows that

$$N \langle k \rangle \sum_{l=1}^{L} \langle z_l \rangle < \mathcal{C} < LN \langle k \rangle \sum_{l=1}^{L} \langle z_l \rangle. \tag{33}$$

For *fixed* L, the complexity grows *linearly* with N as $N \to \infty$. For $L = L^*$ we can use (28) to conclude that

$$\mathcal{O}(NM) < \mathcal{C} < \mathcal{O}(L^* NM), \tag{34}$$

where $M = N\langle k \rangle / 2$ denotes the total number of edges in the network. Recall that the Brandes or Newman algorithm has a complexity of $\mathcal{O}(NM)$ for obtaining the traditional betweenness centralities. Specializing the expression (32) to networks with exponentially growing shells one finds the same $\mathcal{O}(NM)$ complexity [that is the upper bound $\mathcal{O}(NM \ln M)$ in (34) is *not* realized]; see Fig. 3; for networks with power-law growth shells, however, we find $\mathcal{O}(N^{1+1/d} M)$, as in the upper bound of (34). The extra computational cost is due to the fact that instead of a single value, our algorithm produces a set of L numbers (the l-BCs), providing multiscale information on betweenness centrality for all nodes and all edges in the network.

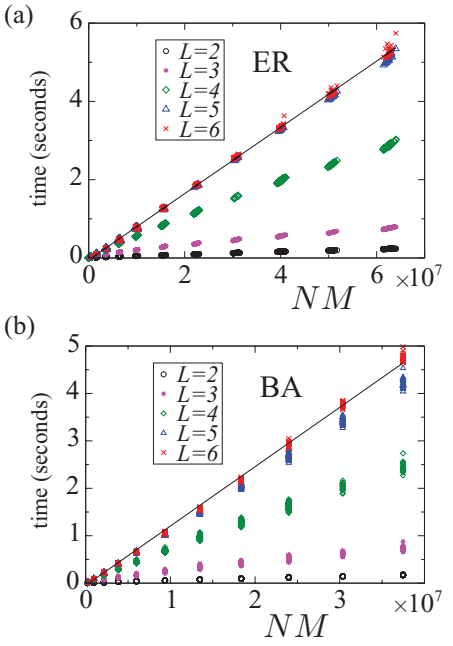


FIG. 3. (Color online) Scaling of range-limited betweenness computation times (in seconds) as a function of NM for ER and BA models, where N is the number of nodes and M is the number of edges of the graphs. For ER the average degree was 5 and the BA model's parameter was $m = 3$. The symbols are actual running times (*not averages*) for 100 networks, measured on an Apple quad core Mac Pro workstation.

E. Freezing of ranking by range-limited betweenness

In Ref. [38] we have provided numerical evidence that the ranking of the nodes (same holds for edges) by their $[L]$-BC values freezes at relatively small values of L. Here we show how this freezing phenomenon emerges. Consider two arbitrary nodes i and j, with degrees k_i and k_j. Using Eq. (22) we can write

$$\ln \frac{b_l(j)}{b_l(i)} = \ln \frac{k_j}{k_i} + \xi_l(j) - \xi_l(i) = \ln \frac{k_j}{k_i} + \Delta_l. \tag{35}$$

Based on (24),

$$\Delta_l = \xi_l(j) - \xi_l(i) = \sum_{n=1}^{l-1} \frac{l + 1 - n}{l + 1} X_n, \tag{36}$$

where $X_n = \epsilon_n(j) - \epsilon_n(i)$. By definition, $\epsilon_n(j)$ is the *per-node* variation of shell occupancy from its root-independent value, for the nth shell centered on root node j. Expectedly, for larger shells (larger n), the size of the shells becomes less dependent on the local graph structure surrounding the root node, and for this reason this noise term has a decaying magnitude $|\epsilon_n(j)|$ with n. Thus the X_n can be considered as random variables centered around zero, with a magnitude that is decaying with increasing n. The contributions of the noise terms coming from larger radius shells in the sum (36) is decreasing not only because the corresponding X_ns are decreasing in absolute value, but also because their weight in the sum is decreasing [as

$1/(l+1)$], and therefore when moving from l to $l+1$ in (36) the change (the fluctuation) in Δ_l *decreases* for larger l. This effectively means that the right-hand side of (35) *saturates*, and thus, accordingly, the left-hand side saturates as well, freezing the ordering of betweenness values. If the two nodes have largely different degrees ($\ln k_j/k_i$ is relatively large), the noise term Δ_l will not be able to change the sign on the right-hand side of (35), even for small l values, and thus the ordering between nodes with very different degrees will freeze the fastest, followed by nodes with degrees that are close to each other. Clearly, the freezing of ordering between nodes with identical degrees ($k_j = k_i$) will happen last. The probability for the ordering to flip when increasing the range from l to $l+1$ can be calculated for specific network models, however, it will not be discussed here.

IV. RANGE-LIMITED CENTRALITIES IN A LARGE-SCALE SOCIAL NETWORK

In this section we illustrate the power of the range-limited approach on a real-world social network inferred from cell-phone call logs (SocNet). We show that computing the $[L]$-BCs up to a relatively small limit length can already be used to predict the full, diameter-based betweenness centralities of individual nodes (and edges), their distribution, and the top list of nodes with highest centralities.

This social network was constructed from 708 million anonymized phone calls between 7.2 million callers generated in a period of 65 days. Restricting ourselves to pairs of individuals between which phone calls have been observed in both directions in this period as a definition of an edge, we found that the giant component of this network has about 5.5 million nodes and 27 million edges. The 65 days is long enough to guarantee that individuals with strong social bonds have called each other at least once during this interval, and therefore will be linked by an edge in our graph.

To test and validate our predictions using the range-limited method, we actually performed the computation of the full, diameter-based betweenness centralities of all the nodes in SocNet. To deploy the computation, we used a distributed computing utility called Work Queue, developed in the Cooperative Computing Lab at Notre Dame. The utility consists of a single management server that sends tasks out to a collection of heterogeneous workers and processors. Specifically, our workers consisted of 250 Sun Grid Engine cores, 300 Condor cores, and 12 local workstation cores, for a total of 562 cores. This allowed us to finish thousands of days of computation in the course of 5 days. Each worker received a request to compute the contribution of shortest paths starting from 50 vertices to the betweenness centrality of every vertex in the network, summed the 50 results, and sent them back to the management server. Each time the management server received a contribution, it summed the contribution with all the others and provided another 50 vertices for the worker.

We also determined the network diameter from the data using a similar distributed computing method, obtaining $D = 26$. At first sight this value seems to be at odds with the famous six degrees of separation phenomenon, which implies a much smaller diameter. However, there are two observations that one can make here. (1) The social network has a dense core with protruding branches ("tentacles"), which mathematically speaking, can generate a large diameter. However, the experimentally determined six degrees of separation *does not probe* all the branches, it actually relies on the denser core for information flow. Hence it should be rather similar to the average node-to-node distance, rather than the rigorously defined network diameter. Indeed, the value of $L^* = 9.35$ that we obtained is rather close to the six-degrees observation. (2) The social network constructed based on cell-phone communications gives only *a sample subgraph* of the true social network, where communications happen also face to face and through land-line phone calls. Hence one would likely measure an even smaller L^* were such data available.

A. Predicting betweenness centralities of individual nodes

In large networks, where measuring the full betweenness centralities (i.e., based on all-pair shortest paths) is too costly, we can use the scaling behavior of range-limited BC values to obtain an estimate for the full BC value of a given node. Plotting the $[l]$-BC values measured up to a limit L as a function of l, we can extrapolate to ranges beyond L. In any finite network the $[l]$-BC values will saturate, and thus we expect the appearance of finite-size effects for large enough l, that is in the range $L^* < l \leqslant D$, where L^* is the *typical radius* of the largest shell and can be estimated as described in Sec. III C. In Fig. 4 we plot the $[l]$-BC values $[B_l(i)]$ for $l \leqslant L = 5$ for four nodes of SocNet. The four nodes were chosen to have very different B_l values. Ranking the nodes by their $[l = 5]$-BC values, node i ranked the highest, and nodes j, k, and m ranked 100, 1000, and 10 000, respectively. The horizontal dashed lines represent the full BC values of the nodes obtained from the exact, diameter-length based measurements (as described above). Fitting the five values and extrapolating the range-limited BCs, we can see that for nodes i, j, and k, the curves reach their corresponding full BC at around $l \simeq 9.5$ agreeing well with the typical length $L^* \simeq 9.35$ estimated in Sec. III C. For low ranking nodes (small full BC) finite-size effects should appear at lengths larger than L^*, because they are situated towards the periphery of the graph. Indeed, one can see from Fig. 4 that node m reaches its full BC at $l \simeq 10.3$, still fairly close to the estimated L^*. Figures 4(b) and 4(c) show the same procedure for ER and BA models.

Thus once we determined L^* as described in Sec. III C, then by simply extrapolating the fitting curve to the $[l]$-BCs of a given node up to $l = L^*$, we obtain an estimate and lower bound for its full betweenness centrality.

B. Predicting BC distributions

In SocNet the B_l values have a log-normal distribution [38], thus $Q_l[\ln(B_l)]$ can be well fitted by a Gaussian [Fig. 5(a)]. The parameters of the distribution also show a scaling behavior, and extrapolating up to $L^* = 9.35$ we obtain $\mu^* = 17.28$ for the average [Fig. 5(b)] and $\sigma^* = 2.25$ [Fig. 5(c)] for the standard deviation of the Gaussian. This predicted distribution is shown as a dashed line in Fig. 5(a). Comparing it with the distribution of the full BC values ($l = D$) we can see that while the averages

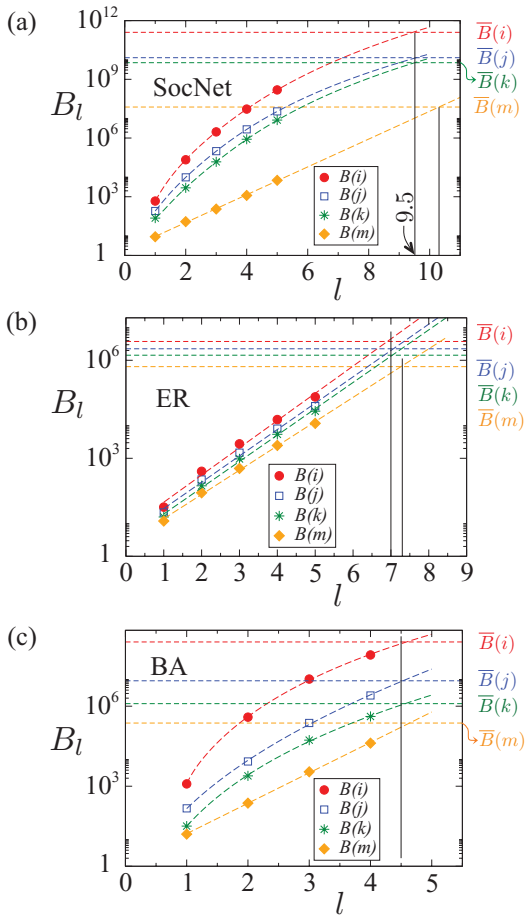


FIG. 4. (Color online) The $[l]$-BC values, B_l, of four individual nodes in the SocNet data (a) in the ER (b) and BA models as function of l. The exact/full BC value of each node is indicated by a horizontal dashed line, and denoted by $\overline{B}(i)$, $\overline{B}(j)$, etc. Extrapolating the range-limited values for larger l, the exact BC values are reached at around L^* in all cases.

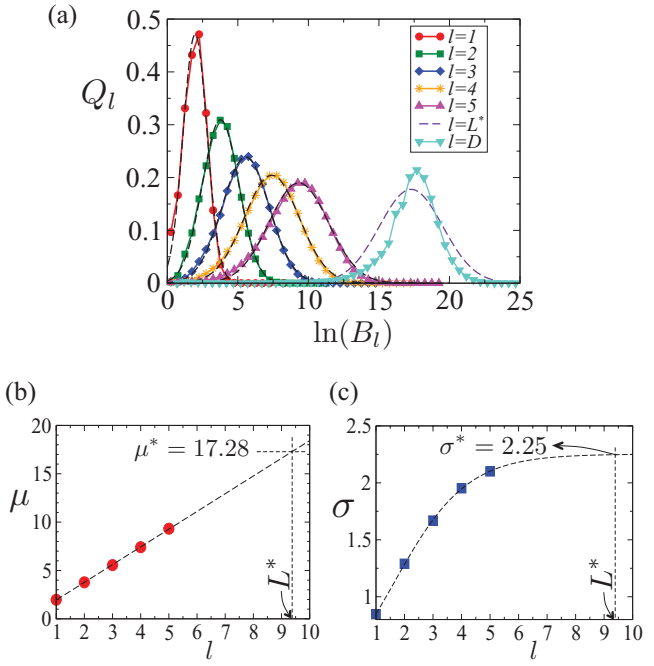


FIG. 5. (Color online) (a) Distribution Q_l of the $\ln(B_l)$ values in the SocNet for $l = 1,2,3,4,5,D$, where $D = 26$ is the diameter, and the predicted distribution for L^*. The distributions can be fitted with a Gaussian. (b) The average μ and (c) standard deviation σ as function of l. Extrapolating to $L^* = 9.35$ we obtain $\mu^* = 17.28$ and $\sigma^* = 2.25$.

agree, the width of the distribution is, however, smaller than the predicted value. This is caused by the fact that the $[l]$-BCs do not saturate at the same l value: for low centrality nodes saturation occurs at larger l, as also shown in Fig. 4.

C. Predicting BC ranking

Efficiently identifying high betweenness centrality nodes and edges is rather important in many applications, as these nodes and edges both handle large amounts of traffic (thus they can be bottlenecks or congestion hot spots), and form high-vulnerability subsets (their removal may lead to major failures). Fortunately, due to the freezing phenomenon described in Sec. III E, one does not need to compute accurately the full BCs in order to identify the top ranking nodes and edges. At already modest l values we obtain top lists that have a strong overlap with the ultimate, $[l = D]$-BC top list. Here we illustrate this for the case of SocNet.

Table I lists the $[l]$-BC (for $l = 1,2,3,4,5$ and $l = D = 26$) of the top ten nodes from the $[D]$-BC list in SocNet. The overlap between the top lists at consecutive l values increases with l. Given two lists, we define the overlap between their first (top-ranking) r elements by the percentage of common elements in both r-element lists. Table II shows the overlap between the top list based on [5]-BC and the one based on the ultimate $[D]$-BC values. At $l = 5$ the top four nodes are already exactly in the same order as in the $[D]$-BC list, the overlap is 90% between the lists of the top ten nodes, and even for the top 100 node lists we have an overlap of 75%.

V. RANGE-LIMITED CENTRALITY IN WEIGHTED GRAPHS

In unweighted graphs the length of the shortest path between two nodes is defined as the number of edges included in the shortest path. In weighted networks each edge has a weight or "length": w_{ij}. Depending on the nature of the network this length can be an actual physical distance (e.g., in road networks), or a cost or a resistance value. We define the "shortest" (or lowest-weight) path between nodes i and j as the network path along which the sum of the weights of the edges included is minimal. We will call this sum the "shortest distance" $d(i,j)$ from node i to node j [note that we allow for directed links, which implies that $d(i,j)$ is not necessarily the same as $d(j,i)$].

In order to define a range-limited quantity, let $b_l(j)$ denote the (fixed) l-BC of node j from all-pair shortest directed paths of length $W_{l-1} < d \leqslant W_l$, where $W_1 < W_2 < \ldots < W_L$ are a series of predefined weight values or "distances". The simplest way to define these W_l distances is to take them uniformly $W_l = l\Delta w$, however, depending on the application, these may be redefined in any suitable way. B_L will again denote the

TABLE I. B_l values of the top ten nodes in the $[D]$-BC top list for SocNet, for $l = 1,2,3,4,5,D$, where $D = 26$ is the diameter.

Vertex	B_1	B_2	B_3	B_4	B_5	B_D
1	600	7.76×10^4	2.06×10^6	3.01×10^7	2.87×10^8	1.26×10^{11}
2	715	9.71×10^4	2.22×10^6	3.05×10^7	2.85×10^8	1.25×10^{11}
3	458	5.04×10^4	1.26×10^6	1.87×10^7	1.86×10^8	9.82×10^{10}
4	377	3.11×10^4	8.56×10^5	1.31×10^7	1.29×10^8	5.82×10^{10}
5	337	2.29×10^4	5.04×10^5	7.23×10^6	7.55×10^7	5.34×10^{10}
6	285	1.93×10^4	5.03×10^5	7.56×10^6	7.85×10^7	5.07×10^{10}
7	488	2.82×10^4	5.84×10^5	7.94×10^6	7.96×10^7	4.89×10^{10}
8	299	2.56×10^4	6.91×10^5	1.09×10^7	1.10×10^8	4.88×10^{10}
9	244	1.47×10^4	3.44×10^5	4.87×10^6	4.83×10^7	4.87×10^{10}
10	239	1.64×10^4	4.57×10^5	7.48×10^6	8.06×10^7	4.81×10^{10}

cumulative L-betweenness, which represents centralities from paths not longer than W_L. Note that we are still counting paths when computing centralities, that is $\sigma_{mn}(i)$ still means the number of shortest paths from m to n passing through i, except for the meaning of "shortest," which is now generalized to lowest cost.

The algorithm is similar to the one presented above for unweighted networks. We again build the subgraph of a node i, but now a shell $G_l(i)$ will contain all the nodes k at shortest path distance $W_{l-1} < d(i,k) \leqslant W_l$ from the root node i. An edge $j \to k$ is considered to be part of the layer in which node k is included. In unweighted graphs a connection $j \to k$ can be part of the subgraph only if the two nodes are in two consecutive layers: if $j \in G_r(i)$ then $k \in G_{r+1}(i)$. In weighted networks the situation is different [Fig. 6(a)]. In principle we may have edges connecting nodes which are not in two consecutive layers, but possibly further away from each other [the links $i \to n$, $j \to o$ in Fig. 6(a)], or even in the same layer (the link $m \to n$ in the same figure).

When building the subgraph using breadth-first search, we need to save the exact order in which the nodes and edges are discovered and included in the subgraph [Figs. 6(b) and 6(c)]. Let us denote with $v(p)$ the index of the node which is included at position p in this node's list [Fig. 6(b)]. This means that the following conditions hold: $d[i,v(1)] \leqslant d[i,v(2)] \leqslant d[i,v(3)] \leqslant \cdots$. Similarly we have a list of edges, where $q_x(p) \to q_y(p)$ is the edge in position p of the list, and q_x, q_y denote the indexes of the two nodes connected by the edge [Fig. 6(c)]. This implies the conditions $d[i,q_y(1)] \leqslant d[i,q_y(2)] \leqslant d[i,q_y(3)] \leqslant \cdots$ [note that every edge $q_x(p) \to q_y(p)$ is included in the edge list when node $q_y(p)$ is discovered]. Again, we calculate $b_l^r(i|k)$ for a node k, and $b_l^r(i|j,k)$ for an edge $j \to k$. As defined above, these values take into account only the shortest paths starting from node i, and r denotes the shell containing the corresponding node or edge. One uses the same initial conditions $\sigma_{ii} = 1$, and $\sigma_{ik} = 0$ for all $k \neq i$, as before.

TABLE II. Overlap between the lists of the top r nodes with highest [5]-BC and with the highest $[D]$-BC values.

Top x nodes	Overlap (%)
1	100
2	100
3	100
4	100
10	90
50	72
100	75
500	70.2
1000	67.1

The algorithm has the following main steps, for every $l = 1, \ldots, L$:

(1) We build the next layer $G_l(i)$ using breadth first search. During this search we build the list of indexes v, q_x, q_y as defined above. We denote the total number of nodes included in the list [from all shells $G_1(i)$ up to $G_l(i)$] as N_l and the number of edges included as M_l. During this breadth-first search we also calculate the σ_{ik} of the discovered nodes. Every time a new edge $j \to k$ is added to the list we update σ_{ik} by adding to it σ_{ij} (using algorithmic notation, $\sigma_{ik} := \sigma_{ik} + \sigma_{ij}$). Recall that σ_{ik} denotes the total *number* of shortest paths from i to k. If the edge $j \to k$ is included in the subgraph (meaning that it is part of a shortest path) the number of shortest paths ending in j has to be added to the number of shortest paths ending in k.

(2) The l-betweenness of all nodes included in the new layer is set to $b_l^l(i|k) = 1$, similarly to Eq. (7).

(3) Going backward through the list of edges we calculate the fixed-l-BC of all nodes and edges. For $p = M_l, \ldots, 1$, we perform the following recursions:

(a) for the edge $q_x(p) \to q_y(p)$,

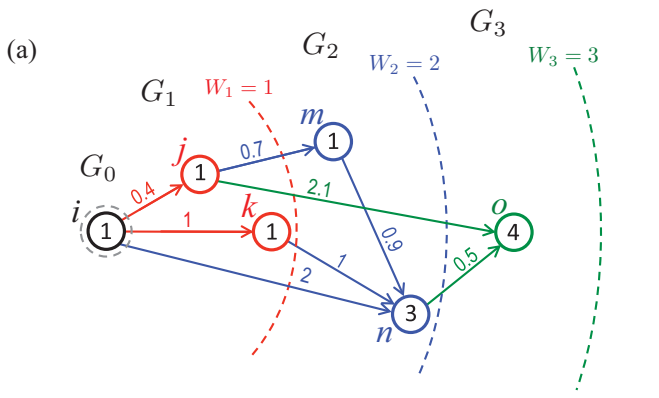
$$b_l^r[i|q_x(p),q_y(p)] = b_l^r[i|q_y(p)]\frac{\sigma_{iq_x(p)}}{\sigma_{iq_y(p)}}, \tag{37}$$

(b) *immediately after* the BC of an edge is calculated, the betweenness of node $q_x(p)$ must also be updated. We have to add to its previous value the l-BC of the edge $q_x(p) \to q_y(p)$:

$$b_l^r[i|q_x(p)] = b_l^r[i|q_x(p)] + b_l^r[i|q_x(p),q_y(p)]. \tag{38}$$

(4) We return to step (1) until the last shell $G_L(i)$ is reached.

As we have seen, the algorithm and the recursions are very similar to the one presented for unweighted graphs. The crucial difference is that the exact order of the discovered nodes and edges has to be saved, because the BC values of edges and nodes in a shell cannot be updated in an arbitrary order. As an example, Fig. 6 shows a small subgraph and the list of



(b)

p	v(p)	b_1	b_2	b_3
1	i	2	2	1
2	j	1	4/3	1/2
3	k	1	1/3	1/4
4	m	0	4/3	1/4
5	n	0	1	3/4
6	o	0	0	1

(c)

p	$q_x(p) \to q_y(p)$	b_1	b_2	b_3
1	$i \to j$	1	4/3	1/2
2	$i \to k$	1	1/3	1/4
3	$j \to m$	0	4/3	1/4
4	$i \to n$	0	1/3	1/4
5	$k \to n$	0	1/3	1/4
6	$m \to n$	0	1/3	1/4
7	$j \to o$	0	0	1/4
8	$n \to o$	0	0	3/4

FIG. 6. (Color) (a) Shells of the $\boldsymbol{C}_3$ subgraph of node i (black) are colored red, blue, green. Distances defining the shells are $W_1 = 1$, $W_2 = 2$, $W_3 = 3$. The weight or length is shown next to each edge. Given a node j, the number inside its circle is the total number of shortest paths coming from the root i: σ_{ij}. (b) The list of nodes $v(p)$ and (c) list of edges $q_x(p) \to q_y(p)$ are shown together with their one-, two-, and three-betweenness values.

nodes and edges together with their one-, two-, and three-betweenness values.

VI. VULNERABILITY BACKBONE

An important problem in network research is identifying the most vulnerable parts of a network. Here we define the vulnerability backbone (VB) of a graph as the smallest fraction of the highest betweenness nodes *forming a percolating cluster* through the network. Removing simultaneously all elements of this backbone will efficiently shatter the network into many disconnected pieces [51,82]. Although the shattering performance can be improved by sequentially removing and *recomputing* the top-ranking nodes [51], here we focus only on the simultaneous removal of the one-time computed VB of a graph, the generalization being straightforward.

Next we illustrate that range-limited BCs can be used to efficiently detect this backbone by performing calculations up to a length much smaller than the diameter. This is of course expected in networks that have a small diameter [$D = \mathcal{O}(\ln N)$ or smaller], however, it is less obvious for networks with large diameter [$D = \mathcal{O}(N^\alpha)$, $\alpha > 0$]. For this reason, in the following we consider random geometric (RG) graphs [83,84] in the plane. The graphs are obtained by sprinkling at random N points into the unit square and connecting all pairs of points that are found within a given distance R of each other. We will use the average degree

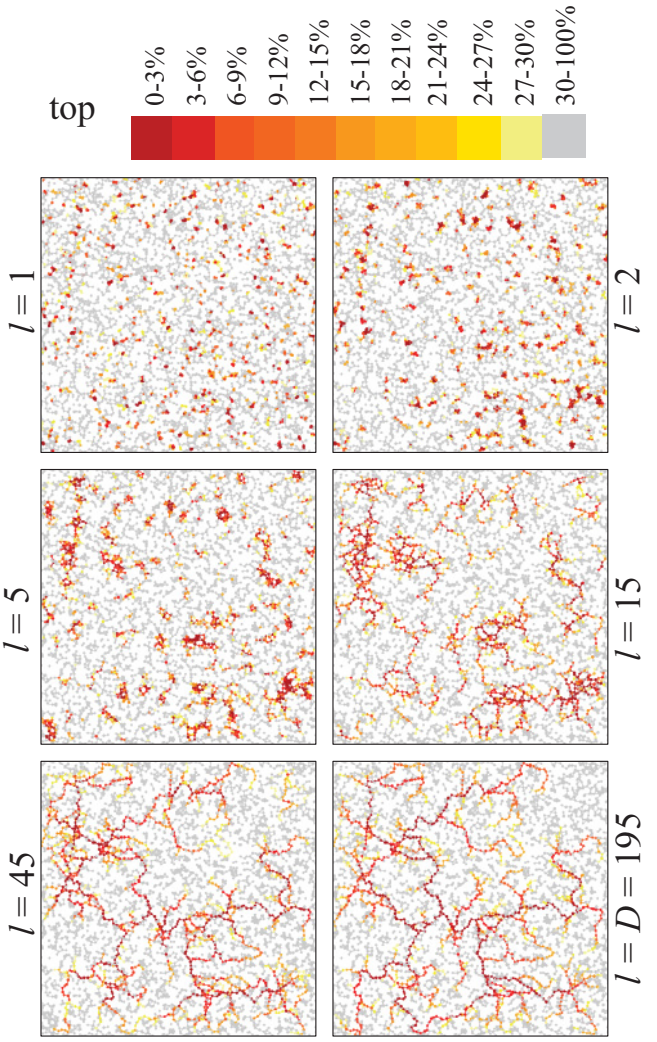


FIG. 7. (Color) The vulnerability backbone VB of a random geometric graph in the unit square with $N = 5000$, $\langle k \rangle = 5$, and $D = 195$. The top 30% of nodes are colored from red to yellow according to their [l]-BC ranking (see color bar). The VB based on the [l]-BC is shown for different values: $l = 1,2,5,15,45,195$.

$\langle k \rangle = N\pi R^2$ [84] instead of R to parametrize the graphs. In Fig. 7 we present measurements on a random geometric graph with $N = 5000$ nodes, average degree $\langle k \rangle = 5$. The hop-count diameter of this graph is $D = 195$. The weights of connections are considered to be the physical (Euclidean) distances. Clearly, since the links of the graph are built based on a rule involving the Euclidean distances, the weight structure and the topology of the graph should be tightly correlated. Thus we do expect strong correlations between the [l]-BC values measured both from the unweighted and the weighted graph. The weight ranges W_l defining the layers during the algorithm were chosen as $W_l = 0.00725l$, $l = 1, \ldots, D$, so that $W_D = 0.007\,25D = 1.413$ is close to the diagonal length of the unit square $\sqrt{2}$. The nodes and connections are colored according to their [l]-BC ranking for different l values (see the color bar in Fig. 7). The backbone is already clearly formed at $l = 45$. Figure 8 compares the VBs of the graphs obtained with and without considering the connection weights (distances).

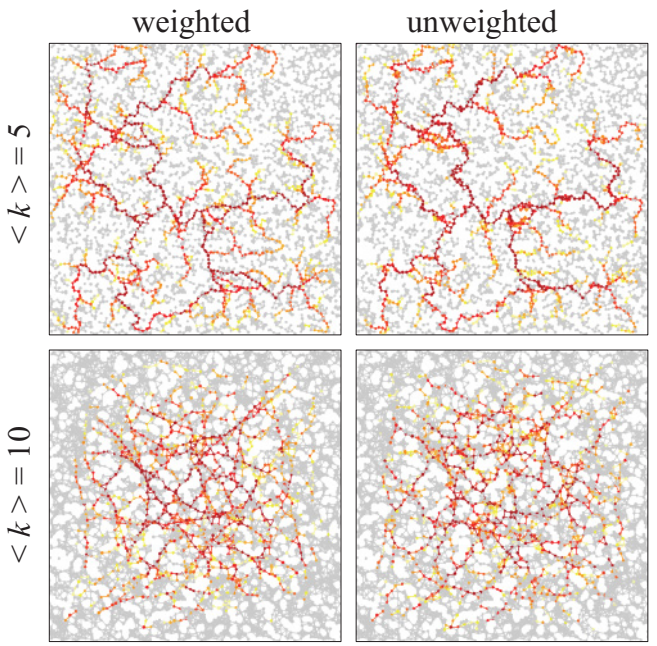


FIG. 8. (Color) Vulnerability backbones based on full BC rankings in two random geometric graphs with $N = 5000$ nodes, and average degrees $\langle k \rangle = 5$ and $\langle k \rangle = 10$, respectively. The rankings were calculated both on the unweighted graph (left column) and weighted one (right column).

Two RGs with densities $\langle k \rangle = 5$ and $\langle k \rangle = 10$ are presented. In the case of the denser graph the backbone is concentrated towards the center of unit square, as periphery effects in this case are stronger (we do not use periodic boundary conditions). Although qualitatively the two VBs are similar, the VB is sharper and clearer in the weighted case. There can be actually significant differences between the two backbones, in spite of the fact that one would expect a strong overlap. In Fig. 9 we show these differences by coloring the nodes of the two graphs from Fig. 8 according to the $\ln(r_{nw}/r_w)$ values, where r_{nw} is the rank of a node obtained using the nonweighted algorithm and r_w is obtained using the weighted graph. The nodes are colored from blue to red, blue corresponding to the case when the unweighted algorithm strongly underestimates the weighted ranking of a node and red is used when it overestimates it. Although it is of no surprise that weighted and unweighted backbones differ in networks where the graph topology and the weights are weakly correlated, the fact that there are considerable differences also for the strongly correlated case of random geometric graphs (the blue and red colored parts in the right panel of Fig. 9) is rather unexpected, underlining the importance of using weight-based centrality measures in weighted networks.

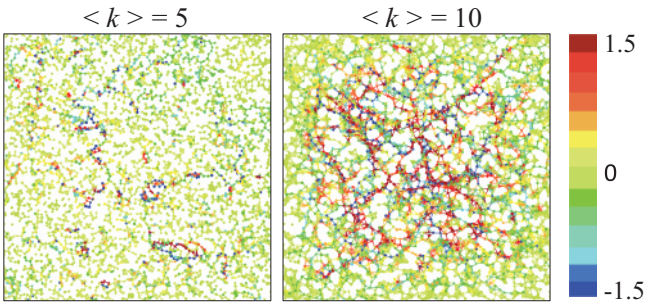


FIG. 9. (Color) Comparison between the rankings obtained with and without considering the weights of connections for the two RG graphs in Fig. 8. Colors indicate the $\ln(r_{nw}/r_w)$ values, where r_{nw} is the rank of a node obtained using the nonweighted algorithm and r_w is obtained with the weighted graph (see the color bar). In denser graphs the differences become more significant.

VII. CONCLUSIONS

In this paper we have introduced a systematic approach to network centrality measures decomposed by graph distances for both unweighted and weighted directed networks. There are several advantages to such range-based decompositions. First, they provide much finer grained information on the positioning importance of a node (or edge) with respect to the network than the traditional (diameter-based) centrality measures. Traditional centrality values are dominated by the large number of long-distance network paths, even though most of these paths might not actually be used frequently by the transport processes occurring on the network. Due to the fast growth of the number of paths with distance in large complex networks, one expects that the distribution of the centrality measures (which incorporate these paths) to obey scaling laws as the range is increased. We have shown both numerically and via analytic arguments (identifying the scaling form) that this is indeed the case, for unweighted networks; for the same reasons, however, we expect the existence of scaling laws for weighted networks as well. We have shown that these scaling laws can be used to predict or estimate efficiently several quantities of interest that are otherwise costly to compute on large networks. In particular, the largest typical node-to-node distance L^*, the traditional individual node and edge centralities (diameter range) and the ranking of nodes and edges by their centrality values. The latter is made possible by the existence of the phenomenon of fast freezing of the rank ordering by distance, which we demonstrated both numerically and via analytic arguments. We have also introduced efficient algorithms for range-limited centrality measures for both unweighted and weighted networks. Although they have been presented for betweenness centrality, they can be modified to obtain all the other centrality measure variants.

Finally, we presented an application of these concepts in identifying the vulnerability backbone of a network, and have shown that it can be identified efficiently using range-limited betweenness centralities. We have also illustrated the importance of taking into account link weights [85] when computing centralities, even in networks where graph topology and weights are strongly correlated.

ACKNOWLEDGMENTS

This project was supported in part by the NSF BCS-0826958, HDTRA 1-09-1-0039, and by the Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053. M.E.R. was partly supported by a grant of the Romanian CNCS-UEFISCDI, Project No. PN-II-RU-TE-2011-3-0121.

[1] R. Cohen and S. Havlin, *Complex Networks: Structure, Robustness and Function* (Cambridge University Press, New York, 2010).

[2] M. Newman, *Networks: An Introduction* (Oxford University Press, New York, 2010).

[3] A. Barrat, M. Barthelemy, and A. Vespignani, *Dynamical Processes on Complex Networks* (Cambridge University Press, New York, 2008).

[4] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, Phys. Rep. **424**, 175 (2006).

[5] *Complex Networks*, Lecture Notes in Physics, edited by E. Ben-Naim, F. Frauenfelder, and Z. Toroczkai, Vol. 650 (Springer-Verlag, Berlin, 2004).

[6] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications* (Cambridge University Press, Cambridge, England, 1994).

[7] J. Scott, *Social Network Analysis: A Handbook* (Sage Publications, London, 1991).

[8] G. Sabidussi, Psychometrika **31**, 581 (1966).

[9] N. E. Friedkin, Amer. J. Soc. **96**, 1478 (1991).

[10] S. P. Borgatti and M. G. Everett, Soc. Netw. **28**, 466 (2006).

[11] J. M. Anthonisse, Technical Report BN 9/71, Stichting Math. Centr., Amsterdam, 1971.

[12] L. C. Freeman, Sociometry **40**, 35 (1977).

[13] L. C. Freeman, Soc. Netw. **1**, 215 (1979).

[14] U. Brandes, Soc. Netw. **30**, 136 (2008).

[15] D. White and S. Borgatti, Soc. Netw. **16**, 335 (1994).

[16] S. Borgatti, Soc. Netw. **27**, 55 (2005).

[17] S. Sreenivasan, R. Cohen, E. Lopez, Z. Toroczkai, and H. E. Stanley, Phys. Rev. E **75**, 036105 (2007).

[18] S. Dolev, Y. Elovici, and R. Puzis, J. ACM **57**, 1 (2010).

[19] B. Bollobás, *Modern Graph Theory*, Graduate Texts in Mathematics, Vol. 184 (Springer-Verlag, New York, 1991).

[20] A. Shimbel, Bull. Math. Biophys. **15**, 501 (1953).

[21] A. Perer and B. Shneiderman, IEEE Trans. Vis. Comput. Graph. **12**, 693 (2006).

[22] S. Lammer, B. Gehlsen, and D. Helbing, Phys. A - Stat. Mech. Appl. **363**, 89 (2006).

[23] D. Eppstein and J. Wang, J. Graph Alg. Appl. **8**, 39 (2004).

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

[25] M. E. J. Newman, Phys. Rev. E **64**, 016132 (2001).

[26] M. Everett and S. Borgatti, J. Math. Soc. **23**, 181 (1999).

[27] R. Puzis, Y. Elovici, and S. Dolev, Phys. Rev. E **76**, 056709 (2007).

[28] M. Everett and S. Borgatti, Soc. Netw. **27**, 31 (2005).

[29] P. Bonacich, J. Math. Soc. **2**, 113 (1972).

[30] P. Bonacich, Soc. Netw. **29**, 555 (2007).

[31] J. D. Noh and H. Rieger, Phys. Rev. Lett. **92**, 118701 (2004).

[32] M. Newman, Soc. Netw. **27**, 39 (2005).

[33] K. Stephenson and M. Zelen, Soc. Netw. **11**, 1 (1989).

[34] M. G. Everett and S. P. Borgatti, Soc. Netw. **32**, 339 (2010).

[35] F. Travencolo and L. D. F. Costa, Phys. Lett. A **373**, 89 (2008).

[36] B. A. N. Travencolo, M. P. Viana, and L. D. F. Costa, New J. Phys. **11**, 063019 (2009).

[37] S. F. N., B. A. N. Travencolo, M. P. Viana, and L. D. F. Costa, J. Phys. A **43**, 325202 (2010).

[38] M. Ercsey-Ravasz and Z. Toroczkai, Phys. Rev. Lett. **105**, 038701 (2010).

[39] A. Arenas, A. Diaz-Guilera, and R. Guimera, Phys. Rev. Lett. **86**, 3196 (2001).

[40] R. Guimera, A. Diaz-Guilera, F. Vega-Redondo, A. Cabrales, and A. Arenas, Phys. Rev. Lett. **89**, 248701 (2002).

[41] G. Yan, T. Zhou, B. Hu, Z.-Q. Fu, and B.-H. Wang, Phys. Rev. E **73**, 046108 (2006).

[42] B. Danila, Y. Yu, S. Earl, J. A. Marsh, Z. Toroczkai, and K. E. Bassler, Phys. Rev. E **74**, 046114 (2006).

[43] B. Danila, Y. Yu, J. A. Marsh, and K. E. Bassler, Phys. Rev. E **74**, 046106 (2006).

[44] B. Danila, Y. Yu, J. A. Marsh, and K. E. Bassler, Chaos **17**, 026102 (2007).

[45] F. T. Leighton and S. Rao, J. ACM **46**, 787 (1999).

[46] V. V. Vazirani, *Approximation Algorithms*, 2nd ed. (Springer, New York, 2003).

[47] C. Gkantsidis, M. Mihail, and A. Saberi, in *Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems* (ACM, San Diego, California, 2003), p. 148.

[48] A. Akella, S. Chawla, A. Kannan, and S. Sheshan, in *Proceedings of the Twenty-Second ACM Symposium on Principles of Distributed Computing (PODC 2003)* (ACM, Boston, MA, 2003).

[49] L. Dall'Asta, I. Alvarez-Hamelin, A. Barrat, A. Vázquez, and A. Vespignani, Theor. Comput. Sci. **355**, 6 (2006).

[50] L. Dall'Asta, I. Alvarez-Hamelin, A. Barrat, A. Vázquez, and A. Vespignani, Phys. Rev. E **71**, 036135 (2005).

[51] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, Phys. Rev. E **65**, 056109 (2002).

[52] A. E. Motter and Y. C. Lai, Phys. Rev. E **66**, 065102 (2002).

[53] A. E. Motter, Phys. Rev. Lett. **93**, 098701 (2004).

[54] A. Vespignani, Science **325**, 425 (2009).

[55] L. Dall'Asta, A. Barrat, M. Barthelemy, and A. Vespignani, J. Stat. Mech. Theor. Exp. (2006) P04006.

[56] L. C. Freeman, S. P. Borgatti, and D. R. White, Soc. Netw. **13**, 141 (1991).

[57] U. Brandes, J. Math. Soc. **25**, 163 (2001).

[58] A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani, Proc. Natl. Acad. Sci. USA **101**, 3747 (2004).

[59] H. Wang, J. M. Hernandez, and P. V. Mieghem, Phys. Rev. E **77**, 046105 (2008).

[60] T. Opsahl, F. Agneessens, and J. Skvoretz, Soc. Netw. **32**, 245 (2010).

[61] M. Granovetter, Am. J. Soc. **78**, 1360 (1973).

[62] V. Colizza, R. Pastor-Satorras, and A. Vespignani, Nat. Phys. **3**, 276 (2007).

[63] G. Szabo, M. Alava, and J. Kertesz, Phys. Rev. E **66**, 026101 (2002).

[64] B. Bollobás and O. Riordan, Phys. Rev. E **69**, 036114 (2004).

[65] A. Fekete, G. Vattay, and L. Kocarev, Phys. Rev. E **73**, 046102 (2006).

[66] M. Kitsak, S. Havlin, G. Paul, M. Riccaboni, F. Pammolli, and H. E. Stanley, Phys. Rev. E **75**, 056115 (2007).

[67] D. B. Johnson, J. ACM **24**, 1 (1977).

[68] R. W. Floyd, Commun. ACM **5**, 345 (1962).

[69] S. Warshall, J. ACM **9**, 11 (1962).

[70] U. Brandes and C. Pich, Int. J. Bifurcation Chaos **17**, 2303 (2007).

[71] R. Geisberger, P. Sanders, and D. Schultes, in *ALENEX* (SIAM, San Francisco, California, 2008), pp. 90–100.

[72] M. C. González, C. A. Hidalgo, and A.-L. Barabási, Nature (London) **453**, 779 (2008).
[73] T. Kalisky, R. Cohen, O. Mokryn, D. Dolev, Y. Shavitt, and S. Havlin, Phys. Rev. E **74**, 066108 (2006).
[74] J. Shao, S. V. Buldyrev, R. Cohen, M. Kitsak, S. Havlin, and H. E. Stanley, Europhys. Lett. **84**, 48004 (2008).
[75] L. D. F. Costa, Phys. Rev. Lett. **93**, 098702 (2004).
[76] L. D. F. Costa and S. F. N., J. Stat. Phys. **125**, 845 (2006).
[77] L. D. F. Costa and R. F. S. Andrade, New J. Phys. **9**, 311 (2007).
[78] J. Onnela, J. Saramaki, J. Hyvonen, G. Szabo, D. Lazer, K. Kaski, J. Kertesz, and A. Barabasi, Proc. Natl. Acad. Sci. USA **104**, 7332 (2007).
[79] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, Phys. Rev. E **64**, 026118 (2001).
[80] J. Shao, S. V. Buldyrev, L. A. Braunstein, S. Havlin, and H. E. Stanley, Phys. Rev. E **80**, 036105 (2009).
[81] T. E. Harris, *The Theory of Branching Processes* (Springer-Verlag, Berlin, 1963).
[82] E. Eubank, H. Guclu, V. S. A. Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang, Nature (London) **429**, 180 (2004).
[83] M. Penrose, *Random Geometric Graphs (Oxford Studies in Probability)* (Oxford University Press, New York, 2003).
[84] J. Dall and M. Christensen, Phys. Rev. E **66**, 016121 (2002).
[85] M. A. Serrano, M. Boguna, and A. Vespignani, Proc. Natl. Acad. Sci. USA **106**, 6483 (2009).