Generalization of core percolation on complex networks

N. Azimi-Tafreshi
Physics Department, Institute for Advanced Studies in Basic Sciences, 45195-1159 Zanjan, Iran

S. Osat
Quantum Complexity Science Initiative, Skolkovo Institute of Science and Technology, Skoltech Building 3, Moscow, 143026, Russia

S. N. Dorogovtsev
Departamento de Física da Universidade de Aveiro and I3N, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal and A. F. Ioffe Physico-Technical Institute, 194021 St. Petersburg, Russia

(Received 1 August 2018; revised manuscript received 9 December 2018; published 22 February 2019)

We introduce a k-leaf removal algorithm as a generalization of the so-called leaf removal algorithm. In this pruning algorithm, vertices of degree smaller than k, together with their first nearest neighbors and all incident edges, are progressively removed from a random network. As the result of this pruning the network is reduced to a subgraph which we call the Generalized k-core (Gk-core). Performing this pruning for the sequence of natural numbers k, we decompose the network into a hierarchy of progressively nested Gk-cores. We present an analytical framework for description of Gk-core percolation for undirected uncorrelated networks with arbitrary degree distributions (configuration model). To confirm our results, we also derive rate equations for the k-leaf removal algorithm which enable us to obtain the structural characteristics of the Gk-cores in another way. Also we apply our algorithm to a number of real-world networks and perform the Gk-core decomposition for them.

DOI: 10.1103/PhysRevE.99.022312

I. INTRODUCTION

Structural decomposition of complex networks providing classification of the vertices into different subsets is one of the effective approaches for studying the structural properties of networks. As a primary and well-known example, one can indicate k-core decomposition, which is an efficient technique for uncovering structural properties of large networks [1,2]. The k-core of a network is defined as the largest subgraph whose vertices have degree at least k [3]. There is a pruning algorithm enabling one to obtain k-core subgraphs for a given network: at each step, a vertex of degree less than k is randomly chosen and removed. The pruning is continued until no further removal is possible. As the result of this pruning the network is decomposed to a set of enclosed k-cores. The vertices belonging to higher (more central) cores are more strongly connected. It was also shown that the vertices of the inner core are more influential spreaders in epidemic processes [4]. A giant k-core emerges above a percolation threshold [5]. The most remarkable result is that for $k \geqslant 3$ the giant k-core shows a discontinues hybrid phase transition combining discontinuity and a critical singularity [5,6]. Furthermore, generalized models for k-core percolation have been studied on interdependent and multiplex networks, which reveal more features than the ordinary k-core percolation problem on single networks [7,8].

Another key subgraph of a random network is simply called its core. These subgraphs significantly differ from the k-cores. A core of an undirected network is obtained only by a pruning algorithm in contrast to the k-core, which is, in addition, defined by a specific constraint on the connectivity of its vertices. The pruning algorithm producing a core is called the leaf removal algorithm and was introduced by Karp and Sipser [9]. In this pruning algorithm, a vertex of degree one (a leaf) is randomly chosen and removed together with its neighbor and all incident edges. The algorithm is continued until no leaves remain. The resulting subgraph is formed by some isolated subgraphs and the giant one, which is called the core. For the Erdős-Rényi (ER) random graphs, Bauer and Golinelli showed that the core percolation threshold is located at the mean degree $\langle q \rangle = e = 2.718\ldots$, so that above this point the network contains the giant core, while below the threshold the size of the giant core is zero [10]. The core structure and the phase transition at $\langle q \rangle = e$ is related to a number of phenomena in physics such as conductor-insulator transitions [11] and replica symmetry breaking in the minimal vertex covers [12]. Moreover it was shown that the formation of the core is related to controllability robustness [13,14] and some combinatorial optimization problems such as the maximum matching and minimum vertex cover [9,12,15]. Also a generalized leaf removal process, which is applicable in the minimum dominating set problem, has been introduced in Ref. [16]. Using a time-dependent analysis, people have studied the core percolation related to this generalized leaf-removal algorithm.

In this paper, we generalize definition of the leaf to the "k-leaf," defined as a vertex of degree less than k. In this algorithm we remove recursively a k-leaf together with all its first neighbors and their incident edges. Following this pruning algorithm, the network is decomposed to a hierarchy of nested cores, similarly to the ordinary k-core decomposition. We call this structure the Generalized k-core (Gk-core). In this

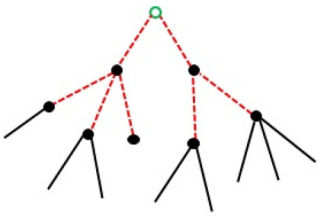

FIG. 1. The open green dot shows a k-leaf. Once a k-leaf is selected, the k-leaf together with the dashed red edges are removed.

notation, the ordinary core is represented by the $G2$-core. The vertices belonging to inner Gk-cores and their first neighbors are of high degree and well connected. Analytical calculation is possible only for the networks with a locally treelike structure. For these kind of networks, and using the generating function technique, we study the structural transitions and emergence points of the Gk-core subgraphs.

The k-leaf removal algorithm can be also considered as the inducing effect, introduced by Zhao *et al.* [17]. In the inducing process, a collapsed vertex i will induce its remaining neighbors, to be collapsed if vertex i has fewer than k remaining neighbors. In Ref. [17] the inducing effect together with the spontaneous collapsing process leads to the emergence of other subgraphs, called protected cores.

The leaf removal algorithm is a Markovian process. We describe evolution of the network structure during the pruning process by applying rate equations, which have been derived for the ordinary leaf removal algorithm on undirected and directed graphs [15,18,19]. This approach provided the size and the emergence point of the ordinary core. In this paper, we also derive rate equations for the degree distribution of a network during the execution of the k-leaf algorithm, which enables us to obtain the structure of the Gk-cores in an alternative way.

The paper is organized as follows. In Sec. II we present an analytical framework to study Gk-core percolation for random networks with arbitrary degree distributions. We apply our general results to the ER and scale-free networks. We compare our results with numerical simulations. In Sec. III we derive the rate equations for the k-leaf removal algorithm, and using these equations we find in another way how Gk-cores are organized. In Sec. IV a set of real-world networks are analyzed in the framework of our approach.

II. ANALYTICAL FRAMEWORK

Let us consider an uncorrelated network with an arbitrary degree distribution $P(k)$. To produce a generalization of the core subgraph, we use the following pruning algorithm: at each step we randomly choose a k-leaf (i.e., a vertex of degree less than k) and remove it together with its neighbors and all incident edges to the neighbors. Figure 1 shows a k-leaf (open green dot) and the k-leaf removal process. As a result of the pruning, the degrees of some vertices change. The procedure is iterated until no vertices of degree less than k remain in the network. The residual network, if it exists, is called the Gk-core.

To find the size of the Gk-core, we classify the vertices into three groups: (1) α-removable: the vertices that can become a leaf; (2) β-removable: the vertices that can become a neighbor of a leaf; (3) the vertices that are neither α-removable nor β-removable and hence belong to Gk-core. Using the assumption that the network has a locally treelike structure, we can write self-consistency equations for probabilities that a random neighbor of a random vertex is α-removable, β-removable, or a nonremovable vertex. We call these probabilities α, β, and $1-\alpha-\beta$, respectively. These probabilities are represented graphically in Fig. 2. Note that the definition of these probabilities is the same as that already defined in Ref. [20]. The difference is in the definition of the leaves.

FIG. 2. Schematic representation of the probabilities α and β.

At least one of the neighbors of a β-removable vertex must be α-removable. Furthermore, an end vertex of a randomly chosen edge belongs to the Gk-core, if it has at least $k-1$ neighbors which belong to the Gk-core and none of its neighbors are of type α. Taking into account these facts, we write the following two self-consistent equations:

$$1-\alpha-\beta=\sum_q \frac{qP(q)}{\langle q\rangle} \times \sum_{s=k-1}^{q-1}\binom{q-1}{s}(1-\alpha-\beta)^s\beta^{q-1-s},$$
$$\beta=1-\sum_q \frac{qP(q)}{\langle q\rangle}(1-\alpha)^{q-1}. \quad (1)$$

The first equation represents the probability that an end vertex of a randomly chosen edge belongs to the Gk-core. $qP(q)/\langle q\rangle$ is the probability that the end vertex of a uniformly randomly chosen edge has degree q, and the combinatorial multiplier $\binom{m}{n}$ gives the number of ways one can choose n edges from a sample of m edges. At least $k-1$ edges of $q-1$ edges (other edges than the starting one) must lead to the Gk-core. Equation (1) also shows the probability that an end vertex of a randomly chosen edge is β-removable. At least one of the neighbors of a β-removable vertex must be a leaf, i.e., an α-removable vertex. These two equations are schematically represented in Fig. 3.

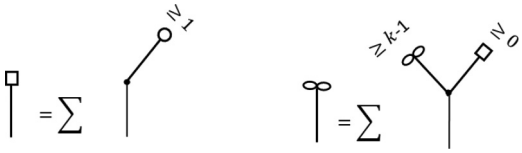

FIG. 3. Graphical representation of the self-consistency equations for the probabilities β and $1-\alpha-\beta$.

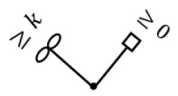

FIG. 4. Schematic representation of the probability that a vertex belongs to the Gk-core, which is the relative size n_{kc} of the Gk-core.

From Eq. (1), one can derive the following self-consistency equation for α:

$$\alpha = \sum_q \frac{qP(q)}{\langle q\rangle} \sum_{s=0}^{k-2} \binom{q-1}{s} (1-\alpha-\beta)^s \beta^{q-1-s}. \quad (2)$$

The probabilities α and β enable us to obtain the probability n_{kc} that a randomly chosen vertex belongs to the Gk-core, which is also the relative size of the Gk-core. Figure 4 shows a schematic representation of this probability. A vertex is in the Gk-core if the vertex has at least k neighbors which belong to the Gk-core. Hence we can write the following equation for the relative size of the Gk-core:

$$n_{kc} = \sum_{q>k} P(q) \sum_{s=k}^{q} \binom{q}{s} (1-\alpha-\beta)^s \beta^{q-s}. \quad (3)$$

To be able to solve Eqs. (1)–(3) analytically, we rewrite these equations using generating functions [21]. For a network with a given degree distribution $P(q)$, the generating function $G(x)$ is defined as

$$G(x) \equiv \sum_q P(q) x^q. \quad (4)$$

Hence, we obtain the following equations for α, β and n_{kc} in terms of the generating function:

$$\begin{aligned} \alpha &= \frac{1}{\langle q\rangle} \sum_{s=0}^{k-2} \frac{(1-\alpha-\beta)^s}{s!} G^{(s+1)}(\beta), \\ \beta &= 1 - \frac{G^{(1)}(1-\alpha)}{\langle q\rangle}, \\ n_{kc} &= G(1-\alpha) - \sum_{s=0}^{k-1} \frac{(1-\alpha-\beta)^s}{s!} G^{(s)}(\beta), \end{aligned} \quad (5)$$

where we used the notation $G^{(s)}(x)$ for the sth derivatives of $G(x)$.

Furthermore, the probability that both end vertices of an edge in the network belong to the Gk-core is $(1-\alpha-\beta)^2$. Hence, the fraction of edges in the $Gk-$core, denoted by l_{kc}, is obtained as

$$l_{kc} = \frac{c}{2}(1-\alpha-\beta)^2. \quad (6)$$

Let us first consider ER networks with Poisson degree distributions, $P(q) = c^q e^{-c}/q!$, where c is the vertex mean degree for the network. For the Poisson distribution, the generating function and its sth derivative are $G(x) = e^{-c(1-x)}$ and $G^s(x) = c^s e^{-c(1-x)}$, respectively. One can easily find the

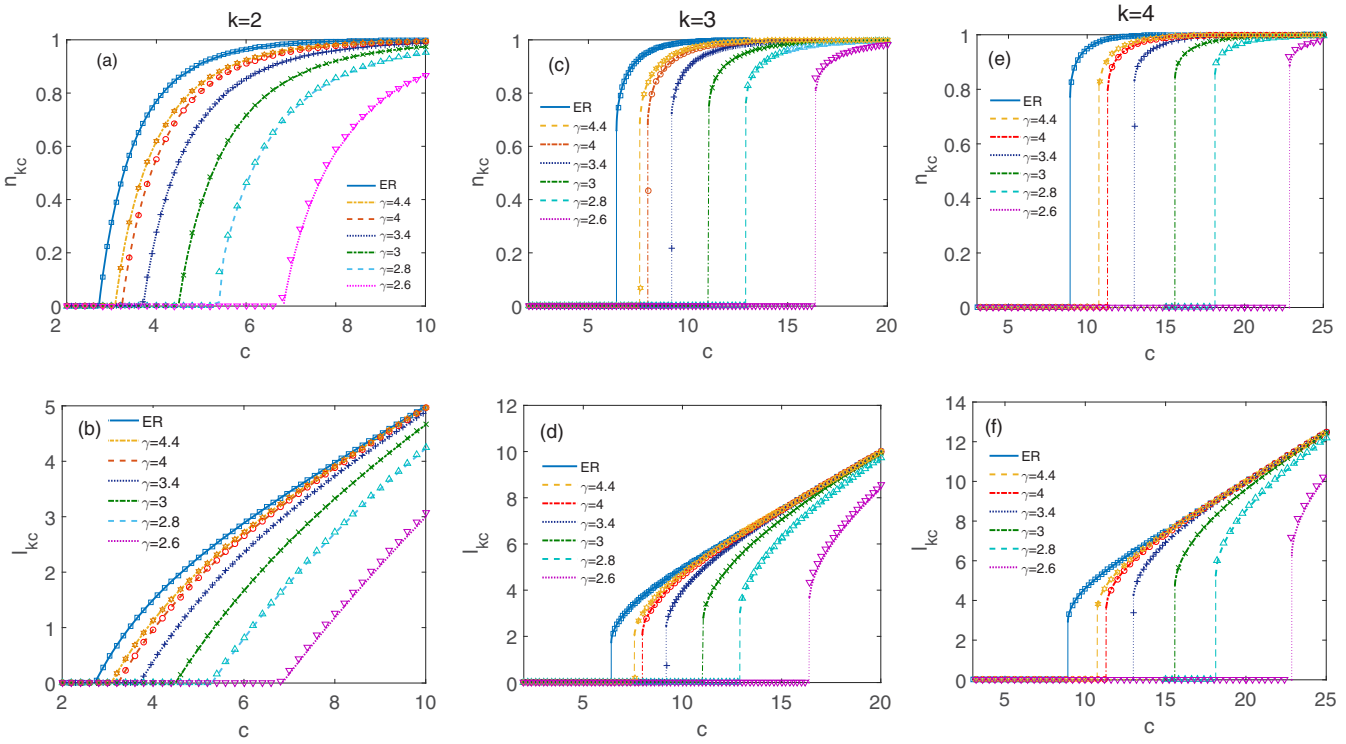

FIG. 5. The relative sizes and the normalized number of edges of the Gk-core for $k = 2, 3, 4$. The points are the results of numerical simulation for the ER and asymptotically scale-free networks of size $N = 10^6$, averaged over 10 realizations. The lines are analytical results obtained from Eqs. (5) and (6). As γ approaches 2, finite size effects become more important, and a deviation between theoretical results and simulations is observed.

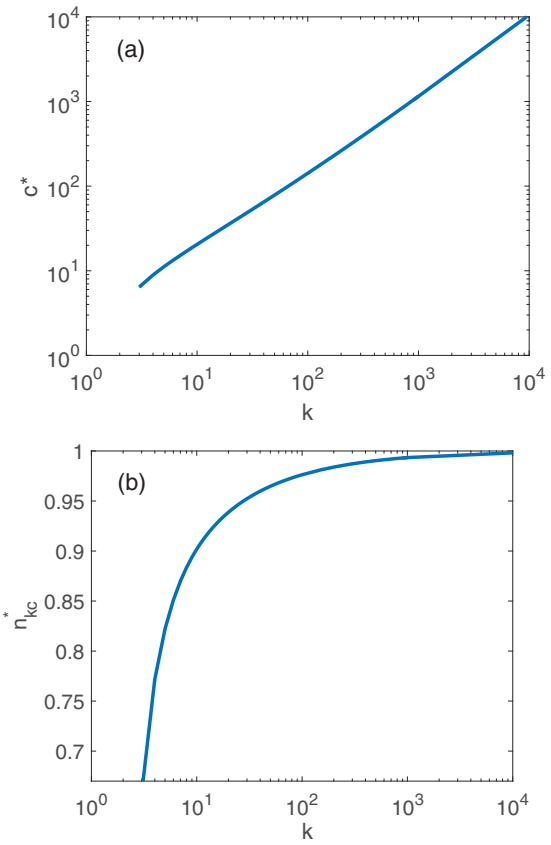

FIG. 6. The behavior of (a) the transition point c^* and (b) the size of the Gk-core at the transition point, n^*_{kc}, vs k for ER random networks.

relation between α and β probabilities as $\beta = 1 - e^{-c\alpha}$, which is independent of the value of k. For ER networks with Poisson degree distributions, one can write a closed form for α and n_{kc} from Eqs. (1) and (2):

$$\alpha = e^{-c\alpha}\frac{\Gamma[k-1, c(e^{-c\alpha}-\alpha)]}{(k-2)!},$$
$$n_{kc} = e^{-c\alpha}\left\{1 - \frac{\Gamma[k, c(e^{-c\alpha}-\alpha)]}{(k-1)!}\right\}, \tag{7}$$

where $\Gamma(s,x)$ is the upper incomplete Γ function.The relative size and the normalized number of edges of the Gk-core for $k = 2, 3$, and 4 are shown in Fig. 5. The analytic results (curves) are compared with numerical simulations (symbols). As we can see in the figure, in contrast to the ordinary core ($k = 2$), for $k \geqslant 3$ a Gk-core emerges discontinuously at the percolation threshold.

Equations (7) enable us to obtain the transition point c^* and the size of the Gk-core at the transition point, n^*_{kc}, for each k. From numerical data, we estimate the asymptotic representations for c^* and n^*_{kc} as the following:

$$c^* \approx k + C\sqrt{k}\ln\ln k,$$
$$n^*_{kc} \approx 1 - \frac{1}{C\sqrt{k}\ln\ln k}, \tag{8}$$
$$C = 2.413\ldots.$$

Figure 6 shows the behavior of c^* and n^*_{kc}, in which the curves asymptotically coincide to Eqs. (8).

Next we consider scale-free networks. It was shown that for the purely power-law scale-free networks the ordinary core does not exist [20]. Hence we consider the asymptotically scale-free, uncorrelated networks generated by the static model with the degree distribution $P(q) = [\frac{c(\gamma-2)}{2(\gamma-1)}]^{\gamma-1}\Gamma[q-\gamma+1, \frac{c(\gamma-2)}{2(\gamma-1)})/\Gamma(q+1) \cong q^{-\gamma}$, where $\Gamma(s)$ is the Γ function [22,23]. For this degree distribution the generating function is $G(x) = (\gamma-1)E_n[(1-x)\frac{c(\gamma-2)}{2(\gamma-1)}]$, where $E_n(x) = \int_1^\infty dy e^{-xy}y^{-n}$ is the exponential integral. Figure 5 shows the relative size and the normalized number of the general 2-, 3-, and 4-cores for different values of γ versus c. With decreasing γ, the emergence point is shifted towards higher values of c. For scale-free networks when $\gamma \to 2$, finite-size effects become more significant. By imposing the proper degree cutoffs, one can eliminate the finite-size effects and the intrinsic degree correlations [24,25]. In Fig. 5 we compare the emergence of cores for asymptotically scale-free and ER networks. As one can see, the dependence of the cores on c for these networks is similar and, as expected, the curves with larger γ approach the result for ER networks.

We define pruning time steps in a way that enables us to classify the vertices of the network into a set of layers for a given k. At time step $t' = 1$, we select the vertices of

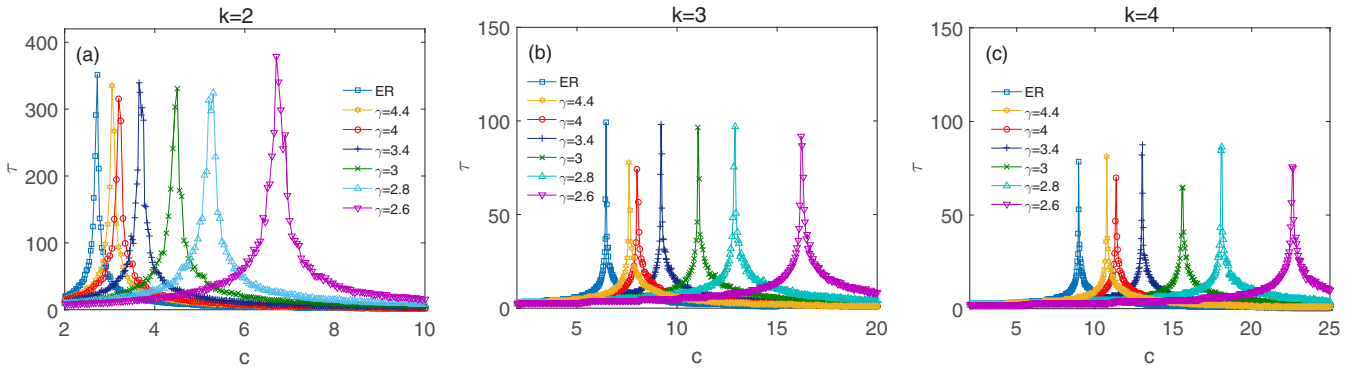

FIG. 7. Total number of pruning steps τ vs mean degree c. The curves shows diverging of $\tau(c)$ at the emergence point of (a) $G2$-core, (b) $G3$-core, and (c) $G4$-core for the ER and asymptotically scale-free networks of size $N = 10^6$, averaged over 10 realizations.

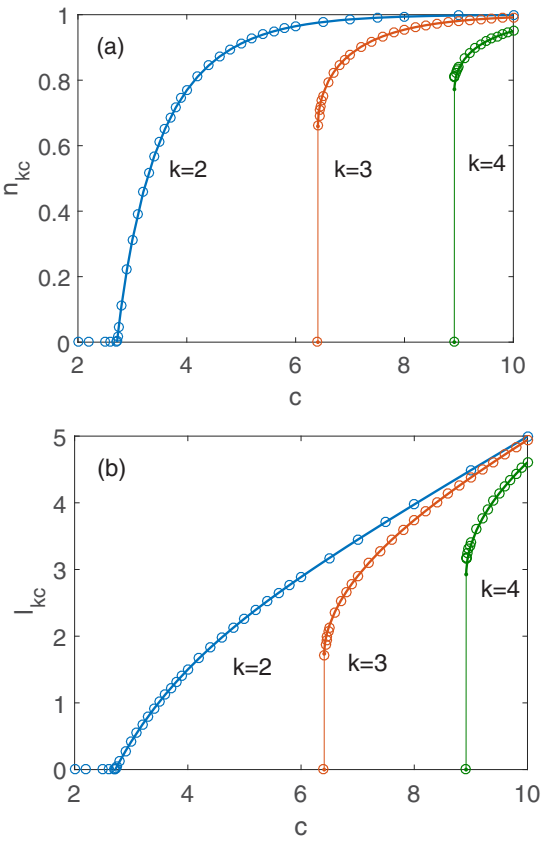

FIG. 8. (a) The relative sizes and (b) normalized number of edges of the Gk-core in ER networks for $k = 2, 3$, and 4, vs the mean degree c of the network. The points show the results obtained by the rate equation approach and lines show the results obtained using the formalism of Sec. II [Eqs. (5) and (6)].

degree less than k (k-leaves) and remove these vertices and their neighbors by applying the k-leaf algorithm. Removing the vertices in the first step may produce new k-leaves, which will be removed at $t' = 2$ and so on. The vertices removed at each step t' form a layer of the network. In other words, the network is pruned layer by layer until there is no k-leaf left. We denote the total number of pruning steps as τ so that $t' = \{1, 2, \ldots, \tau\}$. After τ steps, the network consists of finite components or a giant Gk-core. For different networks we obtain $\tau(c)$ using numerical simulation; see Fig. 7. As we can see, the dependencies $\tau(c)$ diverge at the birth points of the cores.

III. RATE EQUATIONS

The structural evolution of the network during pruning processes is described by the so-called rate equations for the degree distribution of the remaining network [15,18]. Here we derive rate equations for the k-leaf removal algorithm. Let us consider a network of N vertices and L edges. For simplicity we remove only the edges during the pruning process. In other words, at each time step t we choose randomly a k-leaf and remove all k edges incident to it, together with all edges incident to its k neighbors. In this way, the number of vertices of the network remain constant. Note that the time steps t differ from t'. The algorithm is iterated until $P(q) = 0$ for all $q < k$. The important point of this approach is that the dynamics is self-averaging in the thermodynamic limit: $N \to \infty$. After a certain number of time steps, almost all random networks have the same degree distribution, which is independent of the (random) order of the removal of leaves [10]. Hence this approach can be used as a confirmation of the results obtained in the previous section.

We introduce the rescaled time $t = \frac{T}{N}$, where T is the total number of steps of the pruning algorithm, so $\Delta t = 1/N$ is the rescaled time of one iteration. Let $N(q, t)$ be the average number of vertices with degree q at time t. Since the total number of vertices is constant, i.e., $N(t) = N$, we have $N(q, t) = NP(q, t)$. We can write the change of $N(q, t + \Delta t) - N(q, t)$ after one iteration. In the large network limit, we can pass from the discrete difference to the time derivative of the degree distribution and obtain the following evolution equation for the degree distribution:

$$
\begin{aligned}
N(q, t + \Delta t) - N(q, t) &= \dot{P}(q, t) \\
&= -\frac{\theta(k-q)P(q,t)}{\sum_q \theta(k-q)P(q,t)} + \delta_{q,0}\left[1 + \frac{\sum_q q\theta(k-q)P(q,t)}{\sum_q \theta(k-q)P(q,t)}\right] - \frac{\sum_q q\theta(k-q)P(q,t)}{\sum_q \theta(k-q)P(q,t)}\frac{qP(q,t)}{\langle q\rangle_t} \\
&\quad + \frac{\sum_q q\theta(k-q)P(q,t)}{\sum_q \theta(k-q)P(q,t)}\frac{\sum_q q(q-1)P(q,t)}{\sum_q qP(q,t)}\left[\frac{(q+1)P(q+1,t) - qP(q,t)}{\langle q\rangle_t}\right].
\end{aligned} \tag{9}
$$

Let us explain different terms on the right-hand side of Eq. (9). First, we choose a random vertex of degree less than k and remove all edges incident to it. The probability that a vertex has degree less than k is $\frac{\theta(k-q)P(q,t)}{\sum_q \theta(k-q)P(q,t)}$, where $\theta(i)$ is defined for integers: $\theta(i \geqslant 0) = 1$ and $\theta(i < 0) = 0$. Thus with this probability, the number of vertices with $q < k$ decreases by 1. This gives the first term. After removing the edges incident to the leaf and all edges incident to its neighbors, the leaf and all its neighbors become vertices of degree zero. The average number of neighbors of a vertex of degree less than k is $\frac{\sum_q q\theta(k-q)P(q,t)}{\sum_q \theta(k-q)P(q,t)}$. Hence the second term shows the number of vertices whose degrees become zero. On the other hand, the degree distribution of the end vertices of a randomly chosen edge is $\frac{qP(q)}{\langle q\rangle}$. When we remove the edges incident to the nearest neighbors of the leaf, the number of vertices of degree q is decreased by the mean degree of the leaf with probability

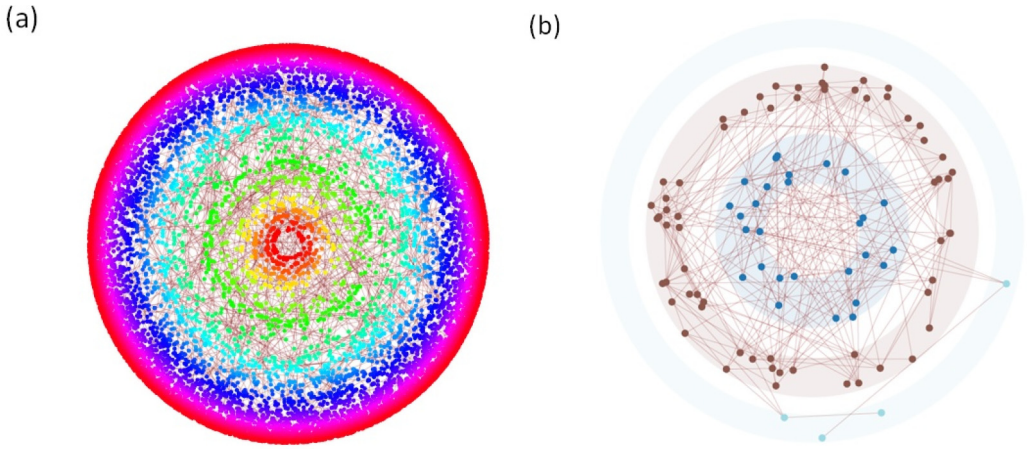

FIG. 9. Graphical visualization of the Gk-core decomposition of (a) astrophysics and (b) transcriptional regulation networks.

$\frac{qP(q)}{\langle q\rangle}$. Finally the last contribution results from modification of degrees of the second neighbors of the leaf. After removal of all edges incident to the leaf and its nearest neighbors, the number of connections of the second nearest neighbors of the leaf decreases by one. The average number of the second neighbors is equal to the mean degree of the nearest neighbors except one (connection to the leaf), multiplied by the average number of the nearest neighbors of the leaf. Equation (9) is a set of differential equations, describing the evolution of a network during the pruning. For $k = 2$, these equations coincide with the known ones [18]. Solving Eq. (9) iteratively, we can obtain the degree distribution of the network at each time step t.

As we already mentioned, we do not remove the vertices during the leaf removal algorithm, and so the total number of the vertices remains constant. However, at each time step all edges incident to the leaf and the edges incident to all its nearest neighbors are removed. Hence, at each time step the average number of removed edges is equal to the mean number of nearest neighbors multiplied by their mean degree. This results to the following evolution equation for the average number of remained edges in the network:

$$\frac{\dot{L}(t)}{N} = -\frac{\langle q^2\rangle_t}{\langle q\rangle_t}\frac{\sum_q q\theta(k-q)P(q,t)}{\sum_q \theta(k-q)P(q,t)}. \tag{10}$$

We apply the leaf removal algorithm to an uncorrelated network with a degree distribution $P(q, t = 0)$ and a vertex mean degree equal to c_0 as the initial conditions. For each value of k, the algorithms are iterated until no vertices of degree less than k remain. To find the Gk-core, the algorithm must continue until time t_k^* at which $P(1, t_k^*) = P(2, t_k^*) = \cdots = P(k-1, t_k^*) = 0$. Our numerical results for different networks show that $P(1, t)$ is the last probability to become zero; that is, the vertices of degree 1 disappear after all other leaves. This is why during iteration we look at the behavior of $P(1, t)$, and the algorithm stops at time t_k^* for a given k. The remaining subgraph is the Gk-core. For $k = 2$ the algorithm coincides with the ordinary leaf-removal algorithm and the remaining subgraph is the $G2$-core or simply the core. After we find t_k^*, the size and the number of edges of the Gk-core

TABLE I. Gk-core decomposition of real networks with the number of vertices N and the number of edges L. $k_{\max}$ is the label of the innermost core. $n_{k_{\max}-\text{core}}$ and $n_{2-\text{core}}$ show the relative size of the innermost and outermost cores, respectively. Similarly, $l_{k_{\max}-\text{core}}$ and $l_{2-\text{core}}$ show the normalized number of edges of the innermost and outermost cores, respectively.

Name	N	L	Ref.	$k_{\max}$	$n_{k_{\max}-\text{core}}$	$l_{k_{\max}-\text{core}}$	$n_{2-\text{core}}$	$l_{2-\text{core}}$
E. coli, transcription	97	212	[26]	3	0.319	0.793	0.917	2.051
AS Oregon	6474	12 572	[27]	2	0.001	0.001	0.001	0.001
Astrophysics	16 046	121 251	[28]	31	0.002	0.045	0.769	5.980
C. elegans, neural	297	2148	[29]	3	0.885	6.037	0.915	6.447
Cond-Mat	16 264	47 594	[28]	10	0.006	0.003	0.618	1.884
Dolphins	62	159	[30]	3	0.161	0.290	0.645	1.322
Email-Enron	36 692	183 831	[31]	7	0.0004	0.001	0.389	1.052
Linux	30 834	213 217	[32]	5	0.0003	0.0008	0.147	0.375
petster-friendship-hamster	1858	12 534	[32]	8	0.010	0.047	0.584	2.664
Sociopatterns-Infectious	410	2765	[33]	9	0.056	0.443	0.912	6.090
PGPgiantcompo	10 680	24 316	[34]	17	0.001	0.014	0.158	0.483
US Air Transportation	500	2980	[35]	3	0.008	0.012	0.260	0.494
Yeast-protein	2284	6646	[36]	3	0.003	0.011	0.025	0.052

can be obtained from the following relations:

$$N_{kc} = N[1 - P(0, t_k^*)], \tag{11}$$

$$L_{kc} = L(t_k^*). \tag{12}$$

We apply this approach to the ER random graphs. The Poisson degree distribution rapidly decays, and it is sufficient to consider $q_{\max} = 30$, i.e., we solve the set of the first 31 equations. Figure 8 shows the size and the number of edges calculated from Eqs. (9)–(12) for the ER networks. In this figure we compare the results obtained by solving the rate equations (points) with the analytic results of the previous section (lines) for the general 2-, 3-, and 4-cores.

IV. REAL-WORLD NETWORKS

We apply the k-leaf removal to a number of real-world networks and find cores of these networks. The characteristics of real-world networks, analyzed in the paper, are listed in Table I. The outermost core is the largest core, which corresponds to $k = 2$ and includes other cores. As we increase the value of k, the size of cores is decreased, and the core corresponding to maximum k ($k_{\max}$) is the smallest and innermost core. We present the relative size and number of edges of the outermost and innermost Gk-cores in Table I. We find that many real social networks are decomposed to a large hierarchy of the Gk-cores. For instance, the layers of arXiv networks, e.g., cond-mat, astro-ph, or hep-th, have the highest numbers of the Gk-cores nested into each other among networks analyzed in this paper. In contrast, the food webs and biological networks have a small number of cores. Using the visualization algorithm proposed in Ref. [1], visualization of the astrophysics network in 2005 [28] and transcriptional regulation network [26] are presented as two examples in Fig. 9. The regulation network has a few cores, while the astrophysics network has around 30 cores in our proposed network decomposition scheme. Comparing with the random networks, the real networks have more cores. Similarly to what was found in the ordinary core problem, this difference reveals that other structural features such as correlations and clustering may be significant for the sizes and organization of the Gk-cores.

V. CONCLUSION

In this work we have generalized the ordinary core subgraph to the Gk-cores. We proposed the k-leaf removal algorithm as a generalization of the ordinary leaf removal The k-leaf pruning algorithm enables us to decompose large random networks into a hierarchical set of progressively nested subgraphs which we called the Gk-cores. Our approach can also be considered as a generalization of the ordinary k-core decomposition. In our pruning at each time step, not only the vertices of degree less than k but also their nearest neighbors are removed. Following this pruning, the network is decomposed into a hierarchy of progressively nested Gk-cores such that the vertices, belonging to the inner cores, and also their first neighbors are of higher degree and well connected. Using the generating function technique, we found the structural characteristics and the emergence point of the Gk-cores for the Erdős-Rényi and scale-free random networks. Similarly to the ordinary k-core percolation, Gk-cores show a discontinuous phase transition for $k \geqslant 3$. We compared our results with numerical simulations and observed a complete agreement. In addition, we used the rate equation approach to describe the evolution of degree distribution of random networks during the k-leaf pruning algorithm. We checked that the result of the application of this approach to the ER graph completely coincides with the exact result obtained by the analytical calculations. We have applied the k-leaf removal algorithm to a number of real-world networks. Among the real networks explored, the social networks have a large $k_{\max}$.

We emphasize that in contrast to the k-core decomposition, the Gk-cores are not about the classification of vertices in a network according to their properties but rather about the characterization of a specific robustness of this network. Suppose that a network is attacked by a virus infecting and removing weak vertices (of degree less than k) and their nearest neighbors. The Gk-cores show what will remain of the network after this epidemic. The resilience and robustness of a network against this kind of epidemic is characterized by the size of its Gk-core. This may explain why the social networks that we explored have a large $k_{\max}$.

[1] J. I. Alvarez-Hamelin, L. Dell'Asta, A. Barrat, and A. Vespignani, k-core decomposition of internet graphs: Hierarchies, self-similarity and measurement biases, arXiv:cs/0511007 (2005).

[2] S. Carmi, S. Havlin, S. Kirkpatrick, Y. Shavitt, and E. Shir, A model of internet topology using k-shell decomposition, Proc. Natl. Acad. Sci. U S A **104**, 11150 (2007).

[3] S. B. Seidman, Network structure and minimum degree, Social Netw. **5**, 269 (1983).

[4] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, Identification of influential spreaders in complex networks, Nat. Phys. **6**, 888 (2010).

[5] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, k-Core Organization of Complex Networks, Phys. Rev. Lett. **96**, 040601 (2006).

[6] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, Critical phenomena in complex networks, Rev. Mod. Phys. **80**, 1275 (2008).

[7] N. K. Panduranga, J. Gao, X. Yuan, H. E. Stanley, and S. Havlin, Generalized model for k-core percolation and interdependent networks, Phys. Rev. E **96**, 032317 (2017).

[8] N. Azimi-Tafreshi, J. Gómez-Gardeñes, and S. N. Dorogovtsev, k-core percolation on multiplex networks, Phys. Rev. E **90**, 032816 (2014).

[9] R. M. Karp and M. Sipser, Maximum matching in sparse random graphs, in *Proceedings of the 22nd Annual IEEE Sympo-*

sium on Foundations of Computer Science, Nashville, TN, USA (IEEE, Piscataway, NJ, 1981), pp. 364–375.

[10] M. Bauer and O. Golinelli, Core percolation in random graphs: A critical phenomena analysis, Eur. Phys. J. B **24**, 339 (2001).

[11] M. Bauer and O. Golinelli, Exactly Solvable Model with Two Conductor-Insulator Transitions Driven by Impurities, Phys. Rev. Lett. **86**, 2621 (2001).

[12] M. Weigt and A. K. Hartmann, Number of Guards Needed by a Museum: A Phase Transition in Vertex Covering of Random Graphs, Phys. Rev. Lett. **84**, 6118 (2000).

[13] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, Controllability of complex networks, Nature (London) **473**, 167 (2011).

[14] T. Jia, and M. Pósfai, Connecting Core Percolation and Controllability of Complex Networks, Sci. Rep. **4**, 5379 (2014).

[15] M. Weigt and A. K. Hartmann, *Phase Transitions in Combinatorial Optimization Problems* (Wiley-VCH, Weinheim, 2005).

[16] J. H. Zhao, Y. Habibulla, and H. J. Zhou, Statistical mechanics of the minimum dominating set problem, J. Stat. Phys. **159**, 1154 (2015).

[17] J. H. Zhao, H. J. Zhou, and Y. Y. Liu, Inducing effect on the percolation transition in complex networks, Nat. Commun. **4**, 2412 (2013).

[18] M. Weigt, Dynamics of heuristic optimization algorithms on random graphs, Eur. Phys. J. B **28**, 369 (2002).

[19] N. Azimi-Tafreshi, S. N. Dorogovtsev, and J. F. F. Mendes, Core organization of directed complex networks, Phys. Rev. E **87**, 032815 (2013).

[20] Y.-Y. Liu, E. Csóka, H. Zhou, and M. Pósfai, Core Percolation on Complex Networks, Phys. Rev. Lett. **109**, 205703 (2012).

[21] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, Random graphs with arbitrary degree distributions and their applications, Phys. Rev. E **64**, 026118 (2001).

[22] M. Catanzaro and R. Pastor-Satorras, Analytic solution of a static scale-free network model, Eur. Phys. J. B **44**, 241 (2005).

[23] K.-I. Goh, B. Kahng, and D. Kim, Universal Behavior of Load Distribution in Scale-Free Networks, Phys. Rev. Lett. **87**, 278701 (2001).

[24] M. Boguñá, R. Pastor-Satorras, and A. Vespignani, Cut-offs and finite size effects in scale-free networks, Eur. Phys. J. B **38**, 205 (2004).

[25] J.-S. Lee, K.-I. Goh, B. Kahng, and D. Kim, Intrinsic degree-correlations in the static model of scale-free networks, Eur. Phys. J. B **49**, 231 (2006).

[26] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, Network motifs in the transcriptional regulation network of *escherichia coli*, Nat. Genet. **31**, 64 (2002).

[27] J. Leskovec, J. Kleinberg, and C. Faloutsos, Graphs over time: Densification laws, shrinking diameters and possible explanations, in *KDD '05: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, Illinois, USA* (ACM New York, NY, USA, 2005), pp. 177–187.

[28] M. E. J. Newman, The structure of scientific collaboration networks, Proc. Natl. Acad. Sci. U S A **98**, 404 (2001).

[29] J. D. Watts, and S. H. Strogatz, Collective dynamics of small-world networks, Nature (London) **393**, 440 (1998).

[30] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, Behav. Ecol. Sociobiol. **54**, 396 (2003).

[31] M. W. Mahoney, A. Dasgupta, K. J. Lang, and J. Leskovec, Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters, Internet Math. **6**, 29 (2009).

[32] J. Kunegis, KONECT: The Koblenz Network Collection, in *WWW'13 Companion Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil* (ACM New York, NY, USA, 2013), p. 1343.

[33] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. Van den Broeck, What's in a crowd? Analysis of face-to-face behavioral networks, J. Theor. Biol. **271**, 166 (2011).

[34] M. Boguñá, R. Pastor-Satorras, A. Diáz-Guilera, and A. Arenas, Models of social networks based on social distance attachment, Phys. Rev. E **70**, 056122 (2004).

[35] V. Colizza, R. Pastor-Satorras, and A. Vespignani, Reaction-diffusion processes and metapopulation models in heterogeneous networks, Nat. Phys. **3**, 276 (2007).

[36] D. Bu, Y. Zhao, L. Cai, H. Xue, X. Zhu, H. Lu, J. Zhang, S. Sun, L. Ling, N. Zhang, G. Li, and R. Chen, Topological structure analysis of the protein-protein interaction network in budding yeast, Nucleic Acids Res. **31**, 2443 (2003).