Influence maximization: Divide and conquer

Siddharth Patwardhan,[1] Filippo Radicchi,[1,*] and Santo Fortunato[2,1,†]

[1]*Center for Complex Networks and Systems Research, Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, Indiana 47408, USA*

[2]*Indiana University Network Science Institute (IUNI), Indiana Univeristy, Bloomington, Indiana 47408, USA*

(Received 3 October 2022; accepted 3 May 2023; published 24 May 2023)

The problem of influence maximization, i.e., finding the set of nodes having maximal influence on a network, is of great importance for several applications. In the past two decades, many heuristic metrics to spot influencers have been proposed. Here, we introduce a framework to boost the performance of such metrics. The framework consists in dividing the network into sectors of influence, and then selecting the most influential nodes within these sectors. We explore three different methodologies to find sectors in a network: graph partitioning, graph hyperbolic embedding, and community structure. The framework is validated with a systematic analysis of real and synthetic networks. We show that the gain in performance generated by dividing a network into sectors before selecting the influential spreaders increases as the modularity and heterogeneity of the network increase. Also, we show that the division of the network into sectors can be efficiently performed in a time that scales linearly with the network size, thus making the framework applicable to large-scale influence maximization problems.

DOI: 10.1103/PhysRevE.107.054306

I. INTRODUCTION

The spread of news, ideas, rumours, opinions, and awareness in social networks is generally analyzed in terms of processes of information diffusion [1–4]. A well-established feature of this type of processes on real, heterogeneous networks is that a small fraction of nodes may have a disproportionately large influence over the rest of the system [3,5,6]. Therefore, influence maximization (IM)—the problem of finding the optimal set of nodes that have the most influence or the largest collective reach on the network—is central for potentially many applications [3,7].

Kempe *et al.* were the first to formalize the IM problem [8]. They showed that the problem is NP-hard, and that solutions to the IM problem can only be approximated. Also, they proposed a greedy optimization algorithm guaranteeing a solution that is within a factor $(1 - 1/e) \simeq 0.63$ from the optimal solution for two main classes of spreading models. Greedy optimization consists in building the set of influential spreaders in a network sequentially by adding one spreader at a time to the set. At each stage of the algorithm, the best spreader is chosen as the node, among those outside the current set of optimal spreaders, that generates the largest increment in the influence of the set of spreaders. Importantly, the gain in influence that a candidate spreader could bring is estimated by adding it to the current set of already selected spreaders, and simulating numerically the spreading process. This procedure, although computationally expensive, allows for properly assessing the combined influence that multiple spreaders usually have in a network. The original recipe by Kempe *et al.* can be applied to relatively small networks only. Followup studies further improved upon the complexity of the greedy algorithm proposed by Kempe *et al.,* allowing for the study of IM problems in larger settings [3,9–13]. Speedup is also possible by first dividing the network into sectors, and then performing greedy optimization within each sector separately [3,10–13]. In these approaches, sectors are generally identified in terms of network communities. Finding communities in networks is a task that can be performed in a time that grows linearly with the network size [14]. However, since these algorithms still rely on the estimation of the influence function via numerical simulations, they can only be used to deal with IM problems on networks of moderate size.

As more efficient alternatives, several purely topological metrics of node centrality were proposed to quantify the influence of the nodes [5,9,15–17]. The assumption behind this approach is that a topological centrality metric is a good proxy for dynamical influence. As the computation of a network centrality metric does not involve simulating the actual spreading process, centrality-based algorithms can be applied to study the IM problem in large-scale networks. However, their performance in approximating solutions to the IM problem is systematically worse than that of the greedy algorithm [6].

A common drawback of centrality-based algorithms is assuming that each seed acts as an independent spreader in the network so that the influence of a set of spreaders is given by the sum of the influence of each individual seed. This is clearly a weak assumption. For example, it is well known that, even in the case of simple contagion models like the independent cascade model, solutions of the IM problem consist of influential

*filiradi@indiana.edu
†santo@indiana.edu

nodes that are sufficiently far apart in the network [18,19]. Two main ways of alleviating this issue are considered in the literature. A first way consists in defining an adaptive version of the centrality metric at hand, so that the effect of the already selected spreaders is discounted from the estimation of the influence of the nodes under observation. This trick is able to greatly improve the performance of even basic degree centrality, whose adaptive version excels in performance [6]. A second way proposed by Chen *et al.* is first partitioning the network into sectors, and then estimating nodes' influence within their own sectors [20]. The rationale behind this procedure is that sectors represent relatively independent parts of a network, thus selecting seeds from different sectors represents a straightforward way of reducing the overlap between portions of the network that multiple spreaders are able to influence. The rationale is similar to the one used in greedy optimization performed on network communities [3,10–13]; however, sectors in Chen *et al.* are obtained by clustering nodes on the basis of the node2vec algorithm embedding [21]. One of the advantages of using geometric embedding instead of community structure is the possibility of having full control on the number of sectors used in the division of the network. On the other hand, identifying sectors in a high-dimensional space as the one generated by node2vec is computationally expensive. Further, in the procedure by Chen *et al.*, the number of sectors is set equal to the number of spreaders that should be identified, requiring therefore finding sectors afresh whenever the size of the seed set is varied. The result is an algorithm that does not scale well with the system size. More recently, an approach similar to the one by Chen *et al.* was also considered by Rajeh and Cherifi [22]. Instead of relying on graph embedding, Rajeh and Cherifi divide the network into sectors by leveraging the clusters obtained by popular community detection algorithms like Louvain [23] and Infomap [24].

In this paper, we generalize and combine the above ideas into a scalable approach. We propose a pipeline consisting in dividing the network into sectors and then choosing influential spreaders based on the division of the network into sectors. Scalability is obtained by imposing the number of sectors to be independent from the number of spreaders. We explore three different methodologies to divide the network into sectors, namely graph partitioning, community structure, and hyperbolic graph embedding. The first two methods allow us to identify sectors in the graph in a time that grows linearly with the network size. The use of centrality metrics like adaptive degree centrality that also can be computed in linear time allows us to produce solutions to the IM problem in large networks. Hyperbolic embedding requires instead a time that grows quadratically with the network size, but allows for a flexible and straightforward way of identifying network sectors. The method can be used only in sufficiently small networks.

We systematically validate our approach on a large corpus of real-world networks, demonstrating its effectiveness in approximating solutions to the IM problem. Furthermore, we leverage the Lancichinetti-Fortunato- Radicchi (LFR) network model [25] to show that the method is particularly useful in solving IM problems on modular and heterogeneous networks.

II. METHODS

A. Networks

1. Real networks

We take advantage of a corpus of 52 undirected and unweighted real-world networks. Sizes of these networks range from $N = 500$ to $N = 26\,498$ nodes. The upper bound on the maximum size of the networks analyzed is due to the high complexity of the greedy optimization algorithm, which we use as the baseline for estimating the performance of the other algorithms. We consider networks from different domains. Specifically, our corpus of networks includes social, technological, information, biological, and transportation networks. Details about the analyzed networks can be found in Appendix A.

2. LFR model

To systematically analyze the dependence of the proposed algorithm's performance on the modularity and the heterogeneity of the network structure, we use the LFR network model [25], commonly adopted as benchmark for community detection algorithms [26]. The LFR model allows us to generate synthetic networks with power-law distributions of degree and community size. Parameters of the model are the power-law exponent of the degree distribution τ_1, the average degree $\langle k \rangle$, the maximum degree $k_{\max}$, the power-law exponent of the community size distribution τ_2, and the mixing parameter μ, which is the average fraction of neighbors outside the community of a node. In our experiments, we vary the values of the parameters τ_1 and μ, while we keep the values of the other parameters fixed. τ_1 and μ are particularly important as they control fundamental features of the networks. The parameter τ_1 allows us to control for the heterogeneity of the degree distribution of the network. Low τ_1 values yield heterogeneous networks; high values of τ_1 yield networks with homogeneous degree distributions. The parameter μ controls for the strength of the planted community structure. Low values of μ indicate well separated and pronounced communities; the larger μ is, the less strong the community structure is.

B. Independent cascade model

In this work, we focus our attention on the independent cascade model (ICM) which is one of the most studied spreading models in the context of influence maximization (IM) [8]. The ICM is a discrete-time contagion model, similar in spirit to the susceptible-infected-recovered model [27]. In the initial configuration, all nodes are in the susceptible state, except for the nodes in the set of spreaders that are in the infected state. At a given time step, each infected node first attempts to infect its susceptible neighbors with probability p, and then recovers. Recovered nodes no longer participate in the dynamics. The dynamics proceeds by repeating the previously described iteration over the newly infected nodes. The spreading process stops once there are no infected nodes remaining in the network. The influence of the set of spreaders is quantified as the size of the outbreak, i.e., the number of nodes that are found in the recovered state at the end of the dynamics. Clearly, this number may differ from realization to realization of the model due to the stochastic nature of the spreading events. The IM

problem consists in finding the set of spreaders leading to the largest average value of the outbreak size [8]. The optimization is constrained by the number of nodes that can compose the set of spreaders. The typical setting in practical applications consists in finding a small set of spreaders in a very large network.

As a function of the spreading probability p, the ICM displays a transition from a nonendemic regime, where the size of the outbreak is small compared to the network size, to an endemic regime, where the outbreak involves a large portion of the nodes in the network. The IM problem is particularly challenging and interesting around the point where such a change of regime occurs. We define it as the pseudocritical value p^* of the ordinary bond-percolation model on the network. Specifically, p^* represents the threshold between the nonendemic and endemic regimes for the ICM started from one randomly chosen seed; this fact follows from the exact mapping of critical SIR-like spreading to bond percolation on networks [28]. We stress that each network is characterized by a different p^* value; the numerical estimation of a network's p^* is performed using the Newman-Ziff algorithm [29,30].

C. The divide-and-conquer algorithm

The input of our algorithm is an unweighted and undirected network $G = (V, E)$, with set of nodes V and set of edges E. We denote the size of the network as $N = |V|$. The algorithm requires also to choose the number k of desired influential spreaders, and the number S of sectors used to divide the network. The divide-and-conquer (DC) algorithm consists of two main components (see Fig. 1). First, we divide the network into S sectors, or vertex subsets, $V_1, V_2, \ldots, V_S$. We have $V = \bigcup_{i=1}^{S} V_i$ and $V_i \cap V_j = \emptyset$ for all $i \neq j$. Second, we form the set of k influential spreaders by adding one node at a time to the set. Starting from an empty set, at each of the k iterations, we first select a random sector with probability proportional to its size, and pick the most influential node in the sector that is not already included in the set of spreaders. One can use any suitable methodology to divide the network and any suitable centrality metric to select influential spreaders from the sectors. Clearly, for $S = 1$ no actual division of the network into sectors is performed. In this case, the selection of influential spreaders is made relying on the centrality metric scores only, thus according to the standard procedure used in the literature [6]. For $S = N$, seed nodes are randomly selected.

We note that the above procedure is conceptually identical to the one introduced by Chen *et al.* [20] and Rajeh and Cherifi [22]. However, there are a few important practical differences. First, Chen *et al.* consider high-dimensional node2vec embeddings only [21]. node2vec requires a nontrivial calibration of several hyperparameters that is known to be essential for task performance, but adds significant computational burden to the procedure [31]. Also, the high-dimensionality of the node2vec-embedding space makes the identification procedure of the sectors nontrivial. Finally, Chen *et al.* impose $S = k$, with one seed selected per sector. This fact implies that increasing the seed set from k to $k + 1$ requires redefining the sectors afresh, an operation that requires a time that grows at least linearly with the network size N. Since in IM problems one typically uses a number of spreaders proportional to the size of the system [6], the resulting complexity of the algorithm is at least quadratic. On the other hand, most community detection algorithms, such as those used by Rajeh and Cherifi, do not allow one to control for the size and number of discovered communities. This has a nontrivial impact on the performance of the selected seeds, as demonstrated in Sec. III C of this work.

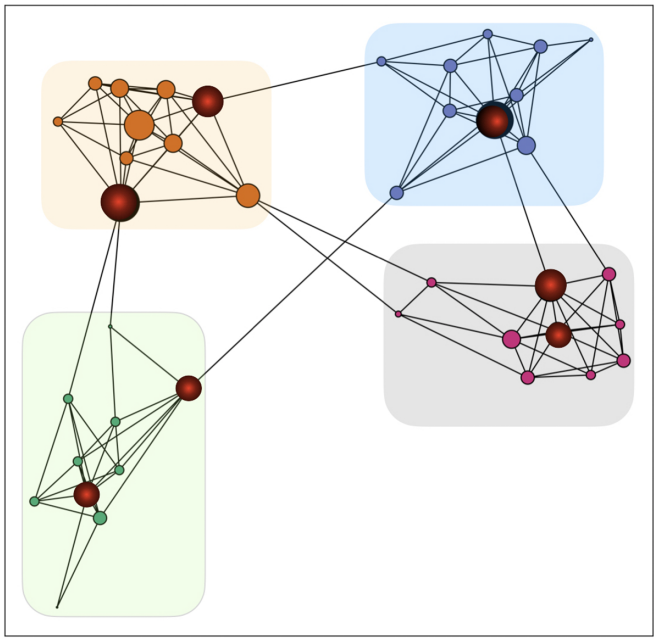

FIG. 1. The divide-and-conquer approach to influence maximization. The network is first divided into sectors of influence, here represented by different colors. Each influential spreader is chosen by first selecting a random sector with probability proportional to its size, and then selecting a node within the sector, that is not yet part of the set of spreaders, according to some criterion, typically the value of a centrality score. The operation is iterated until a desired number of spreaders is selected. The size of each node in the figure is proportional to its degree, here used to proxy nodes' influence. Seven influential spreaders, depicted as bold circles, are selected from the four available sectors.

1. Dividing the network

We consider three possible methods of dividing a network into sectors: (i) graph partitioning, (ii) graph hyperbolic embedding, and (iii) community structure. Below, we briefly summarize each of these methods.

Graph partitioning consists in splitting a graph into an arbitrarily chosen number of sectors of roughly equal size, such that the total number of edges lying between the corresponding subgraphs is minimized [32,33]. To perform graph partitioning, we take advantage of METIS [33], i.e., the algorithm that implements the multilevel partitioning technique introduced in Refs. [34] and [35]. The computational time of METIS grows as $S\,N$ [33].

Graph hyperbolic embedding is another representation that allows to divide a network into sectors. Here, sectors are given

by groups of close-by nodes in vector space. The geometric representation in hyperbolic space offers full control on the size and number of sectors that can be formed. Such a division can be performed efficiently relying on the angular coordinates of the nodes only. This fact greatly simplifies the identification of sectors compared to higher-dimensional embeddings such as those considered by Chen *et al.* [20]. We take advantage of the algorithm named Mercator to map nodes into the hyperbolic disk [36]. Mercator does not have hyperparameters, so no calibration is needed. On the weak side, Mercator performs the embedding of a network with N nodes in a time proportional to N^2, clearly limiting the application of the method to small or medium-sized networks.

Community structure also can be leveraged to divide the network into sectors by assuming that communities represent sectors. This idea is clearly inspired by the IM algorithms of Refs. [3,11–13]. Roughly speaking, the community structure of a network is a partition of the graph into groups of nodes having higher probability of being connected to each other than to members of other groups [26]. Plenty of algorithms are available on the market to find community structure in networks. Here, we take advantage of the Louvain algorithm [23]. Louvain is known for its speed (i.e., computational complexity grows linearly with the number of nodes in the network). It has major limitations [26], but our procedure does not demand high accuracy in the detection of communities and we do not expect results to be dramatically different if one used another community detection algorithm. For instance, we consider two other popular community detection methods: Infomap [24] and label propagation [37]. Results reported in Appendix B indicate that the performance is only mildly affected by the specific community detection algorithm used, with Louvain slightly outperforming the other two methods. Note that Rajeh and Cherifi also explore this idea in [22] using Louvain and Infomap [23,24]. Compared to graph partitioning and graph embedding, an apparent issue in using community structure to define sectors of influence is that community detection algorithms do not generally offer the possibility to control for the size and the number of communities. Community detection methods that allow one to tune the size and/or number of communities to be found exist in the literature [14]. However, we do not consider these methods in the present analysis. We expect these methods to generate network sectors similar to those obtained via graph partitioning.

2. Conquering the network

After the network is divided into sectors, we select sectors at random proportionally to their size. Qualitatively similar results are obtained if sectors are selected proportionally to their total degree (see Appendix C). The most influential node in the selected sector is determined on the basis of topological centrality metrics. This procedure is similar to the one used by Chen *et al.* [9], but different from the one considered in Refs. [3,10–13]. We limit our attention only to metrics that can be computed in a time that grows almost linearly with the network size. We rely on the following metrics.

Adaptive degree centrality is a simple but powerful metric for approximating nodes' influence in IM problems [9]. The metric is designed for the sequential construction of a set of spreaders; in such a procedure, the adaptive degree centrality of a node is given by the total number of connections that a node has towards other nodes that are not included in the current set of spreaders. Erkol *et al.* show that adaptive degree is the most effective heuristic for IM through a systematic comparison of 18 centrality metrics [6]. Unless otherwise specified, all our implementations of the DC algorithm rely on adaptive degree centrality.

Collective influence is a natural generalization of adaptive degree centrality [5]. When computed for node i, the metric is a function of the degrees of the nodes that are at shortest-path distance ℓ from node i. ℓ is a free integer parameter. For $\ell = 0$, the metric reduces to adaptive degree centrality. We report results obtained for $\ell = 2$, which is a standard setting in IM problems [6].

Eigenvector centrality measures a nodes importance while considering the relative importance of its neighbors. It assigns relative scores to all nodes in the network such that an edge to a more central node contributes more to a node's score than an edge to a less central node [38].

D. Notation

For the sake of compactness, we adopt the following notation for the various methods used to approximate solutions of the IM on networks. The strategy used to proxy the influence of individual nodes is denoted by lower-case letters. Specifically, we use g to denote greedy optimization, and r to indicate random selection. For the metrics of centrality we use a to indicate adaptive degree centrality, c for collective influence, and e for eigenvector degree centrality. If the above metrics of centrality are used within our proposed DC scheme, then we use a notation where the lower-case letter of the centrality metric is preceded by an upper-case letter indicating the specific method used to define sectors. We use P to denote graph partitioning, E for hyperbolic graph embedding, and C for community structure. For example, the method m that leverages hyperbolic graph embedding to boost the performance of adaptive degree centrality is denoted as m = Ea; the method m that uses community structure in combination with eigenvector centrality is denoted as m = Ce.

E. Metrics of performance

We measure the performance of each method using a metric similar to the one defined in Ref. [6]. Indicate with $\mathcal{X}_{\mathrm{m}}^{(k)} = \{x_{\mathrm{m}}^{(1)}, x_{\mathrm{m}}^{(2)}, \ldots, x_{\mathrm{m}}^{(k)}\}$ the set of the k seeds identified by the method. We estimate the average value of the outbreak size generated by the set $\mathcal{X}_{m}^{(k)}$ by performing 500 simulations of the ICM. Indicate this quantity as $O_{\mathrm{m}}^{(k)}$. We then compute the sum

$$A_{\mathrm{m}} = \sum_{k=1}^{11} O_{\mathrm{m}}^{(r_k)}, \tag{1}$$

where $r_k = \lfloor [0.01 + (k-1)0.004]\, N \rfloor$ and $\lfloor \cdot \rfloor$ is the floor function. This metric approximates the overall performance of the method m in building sets of influential spreaders of sizes

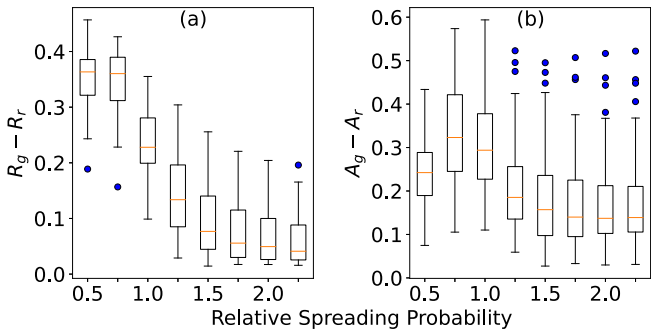

FIG. 2. Influence maximization on real-world networks. (a) For each real network, we evaluate the critical spreading probability p^*. Set of spreaders are identified either using greedy optimization or random selection. We then evaluate the performance metric of Eqs. (2) using 500 ICM realizations for each value of the spreading probability p. We plot the difference $R_g - R_r$ as a function of the relative spreading rate, i.e., the ratio p/p^*. Results stem from the 52 real networks considered in our analysis. The orange line in the box plot represents the median value. The boxes show the first and third quartiles of the data, and the whiskers extend from the box to include the 1.5 interquartile range. The blue points are the data points not included within the error bars. (b) Same as in panel (a), but we plot $A_g - A_r$, as defined in Eq. (1), as a function of the ratio p/p^*.

ranging from 1% to 5% of the network size. The increment 0.004 only serves to divide this range in 10 bins of equal size. We finally compute the ratio

$$R_m = \frac{A_m}{A_g}. \tag{2}$$

According to the above metric, the performance of the method is measured relatively to the baseline provided by greedy optimization, i.e., A_g. The normalization serves to make values of the metric comparable across networks of different size.

III. RESULTS

A. Spreading probability

The value of the spreading probability p has a considerable impact on the outcome of the spreading process, and consequently on the properties of the associated IM problem. Trivially, for $p = 0$ or $p = 1$, any strategy for choosing the set of spreaders is equivalent in terms of performance. The problem becomes non trivial in the vicinity of the pseudo-critical point p^*, where uncertainty in the outcome of the spreading process is maximal if seeds are chosen at random, but appropriately setting the initial condition of the spreading should strongly determine the actual size of the outbreak. In this section, we emphasize the importance of studying the spreading process near the critical threshold p^*. We show results for the 52 networks in our corpus in Fig. 2. We plot $R_g - R_r$ as a function of the relative spreading probability, i.e., p/p^*. Note that each network has its own p^* value. The curve $R_g - R_r$ assumes high values for $p \leqslant p^*$ and drops quickly for $p \geqslant p^*$. The discrepancy between the random and greedy selection strategies is also well characterized by the difference $A_g - A_r$, which peaks around $p \simeq p^*$. Assuming that a generic algorithm for IM displays a performance that is bounded above by the greedy algorithm and bounded below by random selection, we deduce that $p \simeq p^*$ is the regime of the dynamics where different algorithms to approximate the IM problem should be compared. We use the setup $p = p^*$ in all the experiments conducted in this paper.

B. Number of sectors

The proposed DC approach involves first dividing the nodes into S subsets, and then determining the most central nodes within the various sectors. The choice of the parameter S influences the performance and the efficiency of the approach.

We note that the conquer component of the algorithm has computational complexity that is independent of S. For example, computing adaptive degree centrality requires a time that grows as $N \log N$ [39]. However, computing other centrality metrics may be more demanding than that.

The computational complexity of the divide component of the algorithm depends on the specific method utilized. Finding communities with Louvain requires a time that grows slightly superlinearly with the network size N [23]; the number of communities S is not a freely tunable parameter, thus the computational time does not have any explicit dependence on it. Embedding a graph in hyperbolic space with Mercator requires a time that grows quadratically with the system size [36]. Once the embedding is given, the S sectors can be found by first sorting the angular coordinates of the nodes, thus requiring a time that grows slightly superlinearly with N, and then obtaining S slices in a time that grows linearly with S. The computational complexity of METIS grows as SN [33]; it is therefore advisable choosing S growing at most logarithmically with the network size N in order to avoid significant computational burden.

We find that using a value of S between 10 and 20 yields the optimal relative outbreak size for the real networks in

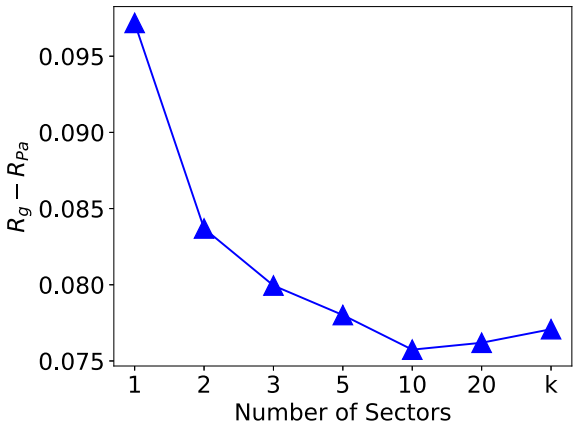

FIG. 3. Sectors of influence in real-world networks. We display the average performance of the DC approach based on graph partitioning and adaptive degree centrality, i.e., $R_g - R_{Pa}$ [Eq. (2)], as a function of the number of sectors S. $S = k$ indicates that sectors are varied between $\lfloor 0.01N \rfloor$ to $\lfloor 0.05N \rfloor$ as we compute the metric of Eq. (1). Performance values shown in the figure are averaged over the 52 networks in our corpus. The outbreaks sizes were obtained from 500 independent simulations of the ICM.

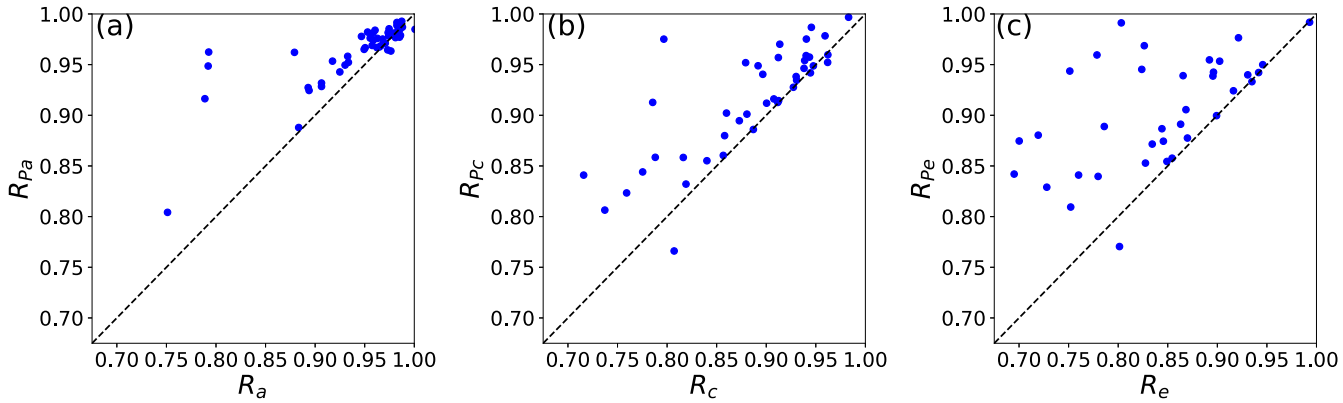

FIG. 4. Performance of the divide-and-conquer algorithm on real networks. (a) Each point in the graph is a real-world network. Their coordinates are given by the estimated ratios R_{Pa} and R_{a}, representing the performance of the divide-and-conquer algorithm leveraging adaptive degree using ten sectors and the one using only one sector, respectively. The dashed line indicates equal performance of the two methods. (b) Same as in panel (a), but for R_{Pc} and R_{c}, i.e., influence of nodes is estimated using collective influence (parameter $\ell = 2$ in this tests). (c) Same as in panel (a), but for R_{Pe} and R_{e}, i.e., influence of nodes is estimated using eigenvector centrality.

our corpus. Clearly, not all the networks are characterized by the same optimal S value; however, we see that any value of $S > 1$ gives us some advantage over $S = 1$. In this paper we set the value of $S = 10$, unless specified otherwise. We justify this choice of S by comparing the metric R_{Pa} defined in Eq. (2) for different values of S. We compare the performance for $S = 1, 2, 3, 5, 10, 20$ in Fig. 3. In the figure, we include also results obtained by setting S equal to the number of k influencers. Please note that this number is not constant, but varied between $\lfloor 0.01N \rfloor$ to $\lfloor 0.05N \rfloor$ while estimating Eq. (1). We see that $S = 10$ is the best choice for our approach.

C. Influence maximization in real and synthetic networks

We consider critical ICM dynamics, and monitor how the size of the outbreak changes as a function of the size of the seed set. We use different variants of the DC algorithm based on graph partitioning, where the influence of individual nodes is estimated based on adaptive degree centrality, collective influence and eigenvector centrality, respectively. We consider $S = 10$ and $S = 1$ sectors. For $S = 1$, there is effectively no divide component in the DC algorithm, thus making it equivalent to the traditional approach to the IM problem considered in the literature [6]. In Fig. 4 we compare directly the metrics of performance of Eq. (2) obtained with $S = 10$ and $S = 1$ over the entire corpus of real networks. The ratios for $S = 1$ are indicated by R_{a}, R_{c}. and R_{e}, for adaptive degree, collective influence, and eigenvector centrality, respectively; for $S = 10$, the ratios are instead indicated as R_{Pa}, R_{Pc}, and R_{Pe}. The scatter plots show that following the divide and conquer strategy one obtains higher scores than selecting influencers from the network as a whole. This holds true regardless of the centrality metric used to proxy the influence of the individual nodes.

Finally, we study how the performance of the DC algorithm depends on the type of method implemented to divide the network into sectors. We find that $R_{\mathrm{Pa}} \geqslant R_{\mathrm{Ca}}$ for 44 out of 52 real networks, meaning that graph partitioning

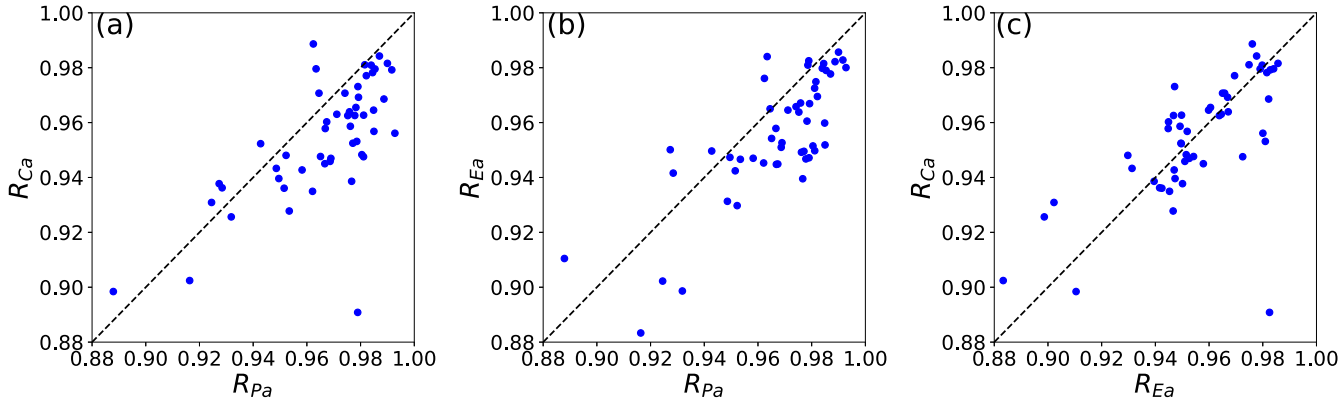

FIG. 5. Performance of the divide-and-conquer algorithm on real networks. (a) Each point in the graph is a real-world network. Their coordinates are given by the estimated R_{Pa} and R_{Ca} values, representing the performance of the divide-and-conquer (DC) algorithm leveraging graph partition and community structure, respectively. In both cases, after the network is divided into sectors, the influence of individual nodes is estimated using adaptive degree centrality. The dashed lines indicate equal performance between the two methods. (b) Same as in panel (a), but comparing R_{Pa} and R_{Ea}, i.e., the performance the DC algorithm based on graph hyperbolic embedding. (c) Same as in (a) and (b), but comparing R_{Ca} and R_{Ea}.

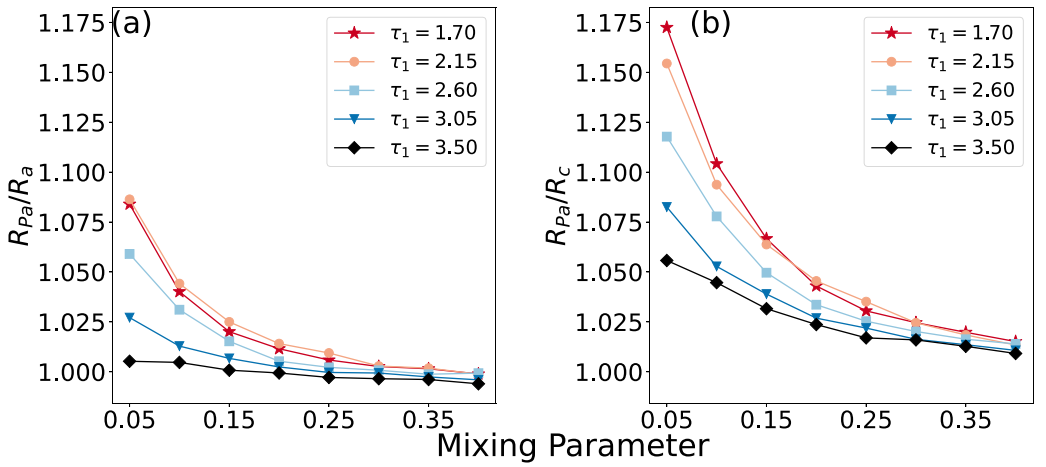

FIG. 6. Performance of the divide-and-conquer algorithm on synthetic networks. (a) We generate synthetic networks using the LFR model [25]. We consider networks with $N = 1000$ nodes, community size power-law exponent $\tau_2 = 1$, average degree $\langle k \rangle = 10$, and maximum degree $k_{\max} = 70$. We plot the ratio $R_{\mathrm{Pa}}/R_{\mathrm{a}}$ as a function of the mixing parameter μ. Different curves correspond to different values of the degree exponent τ_1. (b) We consider the same networks as in panel (a), but we plot $R_{\mathrm{Pa}}/R_{\mathrm{c}}$ as a function of μ.

is better suited than community structure to define sectors of influence in a real network [Fig. 5(a)]. The same result holds for the comparison R_{Pa} vs R_{Ea} [Fig. 5(b)]. Graph embedding and community structure yield instead similar performance [Fig. 5(c)].

We generate LFR networks with $N = 1000$ nodes [25]. We vary the mixing parameter μ from 0.05 to 0.40 to control for the strength of the planted community structure and the degree exponent τ_1 from 1.7 to 4.0 to tune the heterogeneity of the degree distribution. We set the community size power-law exponent $\tau_2 = 1.0$, the average degree $\langle k \rangle = 10$, and the maximum degree $k_{\max} = 70$.

For each network, we identify the best set of seed nodes using three different strategies. Two of these strategies do not involve the division of the network in any sectors; we simply identify the top spreaders via adaptive degree centrality and collective influence with $\ell = 2$ in the entire network. The third strategy takes advantage of the DC algorithm with $S = 10$ sectors defined using graph partitioning; top influencers are identified based on adaptive degree centrality on the various sectors. In Fig. 6, we display the ratios $R_{\mathrm{Pa}}/R_{\mathrm{a}}$ and $R_{\mathrm{Pa}}/R_{\mathrm{c}}$ as functions of the mixing parameter μ of the model. Results are obtained by averaging the ratios over 50 realizations of the network model and of the procedure for the identification of the spreaders. We report results for different values of the degree exponent τ_1. As in the case of real networks, dividing the network into sectors allows us to obtain better solutions to the IM problem than those obtained without any division. The gain in performance increases as the degree heterogeneity of the nodes and the strength of the modular structure of the network increase.

The DC approach performs better than a purely centrality-based one on modular networks as these networks allow for a meaningful division into sectors. Given the modular structure of the graph, choosing spreaders from different sectors is required for successful spreading, as the low density of inter-community edges makes spreading across modules difficult. On the other hand, a purely centrality-based selection protocol likely leads to selecting many influential spreaders within the same set of modules, thereby leading to suboptimal redundancy. Community detection algorithms help in finding these sectors. However, for many real networks, they often find communities with highly heterogeneous size distributions, i.e., some communities are considerably large in comparison to the network. Choosing influencers from large communities leads to the same problems as of traditional approaches that do not rely on the division of the graph into sectors of influence. Graph partitioning and graph hyperbolic embedding are able to circumvent this issue but graph partitioning is able to find more meaningful clusters for IM, as seen in Fig. 5.

Furthermore, the advantage of using a DC strategy instead of a purely centrality-based one is more apparent in heterogeneous networks than in homogeneous networks. This is the consequence of the fact that large-degree nodes tend to be in large communities; thus, a purely centrality-based selection protocol still selects redundant spreaders from a limited set of clusters, rather than distributing them in different parts of the graph. The DC approach clearly overcomes such a limitation. This fact is confirmed in the results of Fig. 9, where we quantify the number of influential spreaders selected in each planted community by the two different protocols.

IV. DISCUSSION

We proposed a two-step strategy to search for effective influencers in networks. By dividing the graph into sectors and finding influencers independently in each sector, via widely adopted centrality scores, we showed that it is possible to increase the relative outbreak size with respect to algorithms sorting nodes based on their centrality in the whole network. The improvement is larger, the more modular the graph is and the more heterogeneous its degree distribution is. The gain produced by our distributed approach does not come at the expense of the time complexity of the procedure, as the division of the network into (a constant number of) sectors can be done in linear time, so the total complexity is dominated by the calculation of the centrality scores. Our numerical experiments show that graph partitioning techniques are highly effective

TABLE I. List of the real networks analyzed in the study. From left to right we report the name of the network, its type, the number of nodes in the giant component, the number of edges in the giant component, the percolation threshold, references to studies where the network is presented and analyzed, and the URL where the network can be found.

Network	Type	N	E	p^*	Ref.	URL
US Air Transportation	transportation	500	2980	0.026	[40]	[41]
URV email	social	1133	5451	0.056	[42]	[43]
Political blogs	information	1222	16714	0.015	[44]	[41]
Air traffic	transportation	1226	2408	0.163	[45]	[46]
Petster, hamster	social	1788	12476	0.025	[45]	[46]
UC Irvine	social	1893	13835	0.023	[47]	[46]
Yeast, protein	biological	2224	6609	0.071	[48]	[41]
Adolescent health	social	2539	10455	0.117	[49,50]	[46]
USFCA	social	2672	65244	0.011	[51–53]	[41]
Japanese	information	2698	7995	0.030	[54]	[55]
Open flights	transportation	2905	15645	0.020	[45,56]	[46]
Pepperdine	social	3440	152003	0.007	[51–53]	[41]
Wesleyan	social	3591	138034	0.009	[51–53]	[41]
Mich	social	3745	81901	0.011	[51–53]	[41]
Bitcoin Alpha	social	3775	14120	0.027	[57–59]	[59]
Bucknell	social	3824	158863	0.008	[51–53]	[41]
Howard	social	4047	204850	0.006	[51–53]	[41]
GR-QC, 1993-2003	social	4158	13422	0.091	[59,60]	[59]
Tennis	social	4338	81865	0.007	[61]	[62]
US Power Grid	technological	4941	6594	0.437	[63]	[64]
HT09	social	5352	18481	0.025	[65]	[66]
Hep-Th, 1995-1999	social	5835	13815	0.108	[67]	[64]
Bitcoin OTC	social	5875	21489	0.023	[57–59]	[59]
Reactome	biological	5973	145778	0.011	[45,68]	[46]
Jung	technological	6120	50290	0.009	[45,69]	[46]
Gnutella, Aug. 8, 2002	technological	6299	20776	0.046	[59,60,70]	[59]
JDK	technological	6434	53658	0.009	[45]	[46]
UChicago	social	6561	208088	0.008	[51–53]	[41]
UC	social	6810	155320	0.010	[51–53]	[41]
Wikipedia elections	social	7066	100736	0.008	[59,71,72]	[59]
English	information	7377	44205	0.011	[54]	[55]
Gnutella, Aug. 9, 2002	technological	8104	26008	0.045	[59,60,70]	[59]
French	information	8308	23832	0.022	[54]	[55]
Hep-Th, 1993–2003	social	8638	24806	0.072	[59,60]	[59]
Gnutella, Aug. 6, 2002	technological	8717	31525	0.065	[59,60,70]	[59]
Gnutella, Aug. 5, 2002	technological	8842	31837	0.056	[59,60,70]	[59]
PGP	social	10680	24316	0.064	[73]	[43]
Gnutella, Aug. 4, 2002	technological	10876	39994	0.076	[59,60,70]	[59]
Hep-Ph, 1993–2003	social	11204	117619	0.005	[59,60]	[59]
Spanish 1	information	11558	43050	0.012	[54]	[59]
DBLP, citations	information	12495	49563	0.032	[45,74]	[46]
Spanish 2	information	12643	55019	0.012	[45]	[46]
Cond-Mat, 1995–1999	social	13861	44619	0.064	[59,67]	[59]
Astrophysics	social	14845	119652	0.018	[67]	[64]
AstroPhys, 1993–2003	social	21363	91286	0.037	[59,60]	[59]
Gnutella, Aug. 25, 2002	technological	22663	54693	0.115	[59,60,70]	[59]
Internet	technological	22963	48436	0.019	None	[64]
Thesaurus	information	23132	297094	0.011	[45,75]	[46]
Cora	information	23166	89157	0.045	[45,76]	[46]
AS Caida	technological	26475	53381	0.021	[59,77]	[59]
Gnutella, Aug. 24, 2002	technological	26498	65359	0.106	[59,60,70]	[59]

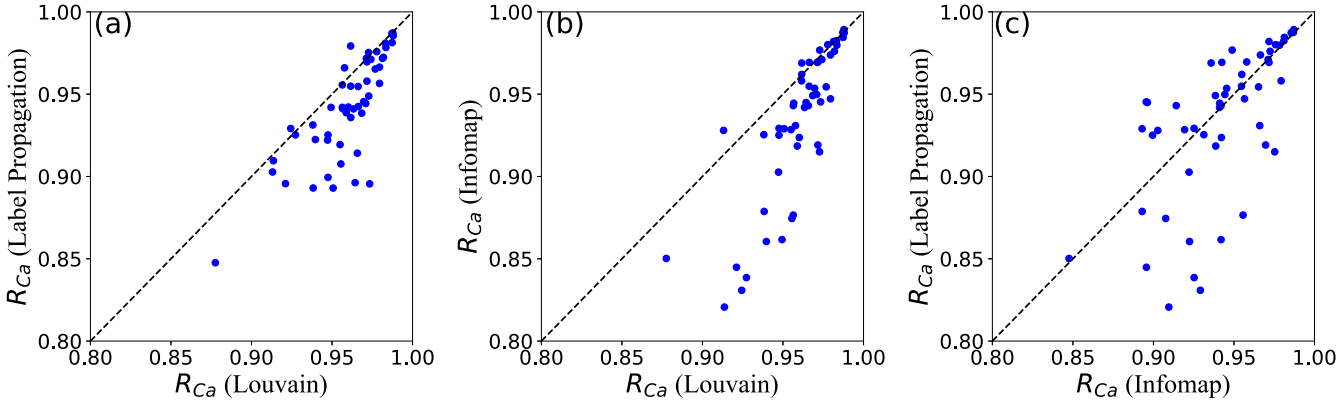

FIG. 7. Performance of the divide-and-conquer algorithm on real networks with different community detection algorithms. (a) Each point in the graph is a real-world network. Their coordinates are given by the estimated ratios R_{Ca} obtained using either Louvain or label propagation to detect communities. The dashed line indicates equal performance of the two methods. (b) Same as in panel (a), but for the comparison between Louvain and Infomap. (c) Same as in panel (a), but for the comparison between Infomap and label propagation.

at identifying the sectors, in comparison to other previously explored approaches like community detection [20] and graph embedding [22]. We expect a similar performance to occur if sectors are identified by community detection methods that allow to tune the size and/or number of communities to be detected. We leave such an extension to future research.

ACKNOWLEDGMENTS

We thank Şirag Erkol for useful advice in the development of the spreading maximization routine. This project was partially supported by the Army Research Office under Contract No. W911NF-21-1-0194, by the Air Force Office of Scientific Research under Awards No. FA9550-19-1-0391 and No. FA9550-21-1-0446, and by the National Science Foundation under Award No. 1927418. The funders had no role in study design, data collection and analysis, the decision to publish, or any opinions, findings, and conclusions or recommendations expressed in the paper.

APPENDIX A: REAL NETWORKS

Table I summarizes the information of the 52 networks considered in the corpus. We report the name of the network, its type, the number of nodes and edges in the giant component, the critical percolation threshold, references to studies where the network is presented and analyzed, and the URL for the data.

APPENDIX B: COMMUNITY DETECTION ALGORITHMS

We consider three popular community detection algorithms in our approach to divide the network into sectors; Louvain [23], Infomap [24], and label propagation [37]. After the network is divided into sectors, we choose the most influential nodes from each sector. Note that the number of influencers picked from each sector is proportional to the size of the sector. Again, we consider critical ICM dynamics and monitor how the size of the outbreak changes as a function of the size of the seed set. We compare the metric R_{Ca} of Eq. (2) obtained by finding communities with the three different community detection methods. We see comparable performance across the methods, with Louvain yielding the best performance over the corpus of 52 real networks (see Fig. 7).

APPENDIX C: DEGREE-BASED SAMPLING

In Fig. 8, we display the comparison between the size-based and the degree-based sampling of sectors. Here, sectors are given by communities identified using Louvain. The size-based sampling strategy is the one used in the entire paper: sectors are selected at random proportionally to the total number of nodes they contain. In the degree-based sampling, the weight of each sector is instead given by the sum of the nodes' degrees within the sector. Results of Fig. 8 indicate that performance is not much affected by the sampling strategy adopted.

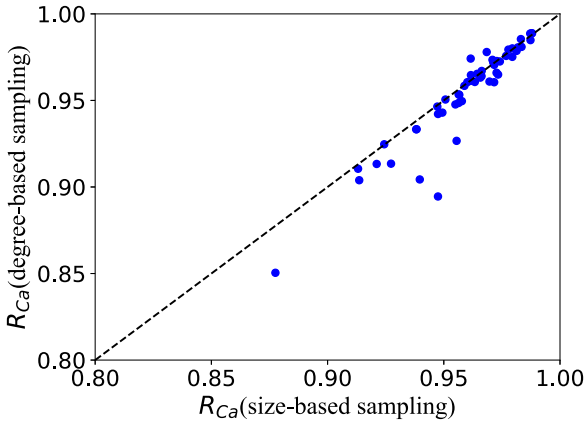

FIG. 8. Performance of the divide-and-conquer algorithm on real networks with a degree-based sampling strategy. Each point in the graph is a real-world network. Their coordinates are given by the estimated ratios R_{Ca} obtained using either the size-based or the degree-based sampling strategy for the sectors, as described in Appendix B.

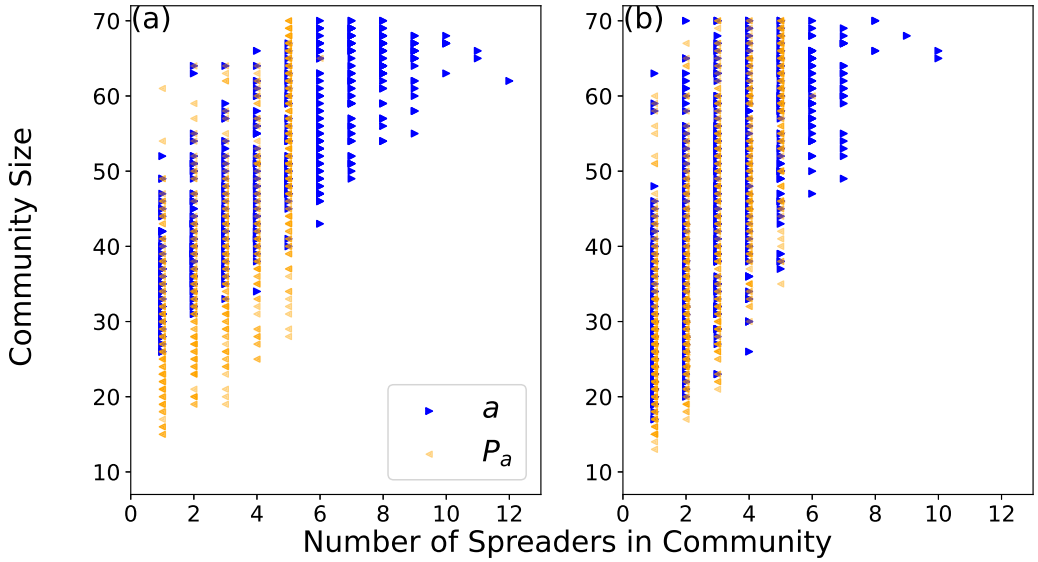

FIG. 9. Selection of influential spreaders in synthetic networks. (a) Each point in the plot represents a planted community in the LFR model. For each community, we display the number of influential spreaders selected from the community and the size of the community. Different colors and symbols correspond to sets of influential spreaders selected using adaptive degree centrality (a, blue) or graph partitioning and adaptive degree centrality (P_a, orange). Results are obtained for $N = 1000$, $\mu = 0.05$, $\tau_2 = 1$, $\langle k \rangle = 10$, $k_{\max} = 70$, and $\tau_1 = 1.7$. The latter yields a heterogeneous degree distribution. We consider 50 instances of the LFR model. In each network instance, we select the top 50 spreaders. (b) Same in panel (a), but for $\tau_1 = 3.5$, which yields a homogeneous degree distribution.

APPENDIX D: EFFECT OF DEGREE HETEROGENEITY

In Fig. 9, we highlight differences in the choice of the influential spreaders in networks with strong modular structure, but variable degree heterogeneity. The DC approach tends to select influential spreaders from the various planted communities in a more uniform manner than a purely centrality-based selection protocol; such a difference is more apparent in networks with heterogeneous degree distributions than in networks with homogeneous degree distributions.

[1] D. Notarmuzi, C. Castellano, A. Flammini, D. Mazzilli, and F. Radicchi, Universality, criticality and complexity of information propagation in social media, Nat. Commun. **13**, 1308 (2022).

[2] M. Newman, *Networks* (Oxford University Press, Oxford, 2018).

[3] S. Banerjee, M. Jenamani, and D. K. Pratihar, A survey on influence maximization in a social network, Knowl. Inf. Syst. **62**, 3417 (2020).

[4] C. Granell, S. Gómez, and A. Arenas, Dynamical Interplay between Awareness and Epidemic Spreading in Multiplex Networks, Phys. Rev. Lett. **111**, 128701 (2013).

[5] F. Morone and H. A. Makse, Influence maximization in complex networks through optimal percolation, Nature (London) **524**, 65 (2015).

[6] Ş. Erkol, C. Castellano, and F. Radicchi, Systematic comparison between methods for the detection of influential spreaders in complex networks, Sci. Rep. **9**, 15095 (2019).

[7] P. Domingos and M. Richardson, Mining the network value of customers, in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2001), pp. 57–66.

[8] D. Kempe, J. Kleinberg, and É. Tardos, Maximizing the spread of influence through a social network, in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2003), pp. 137–146.

[9] W. Chen, Y. Wang, and S. Yang, Efficient influence maximization in social networks, in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2009), pp. 199–208.

[10] Y.-C. Chen, W.-Y. Zhu, W.-C. Peng, W.-C. Lee, and S.-Y. Lee, Cim: community-based influence maximization in social networks, ACM Trans. Intell. Syst. Technol. **5**, 1 (2014).

[11] J. Shang, S. Zhou, X. Li, L. Liu, and H. Wu, Cofim: A community-based framework for influence maximization on large-scale networks, Knowl. Based Syst. **117**, 88 (2017).

[12] A. Bozorgi, H. Haghighi, M. S. Zahedi, and M. Rezvani, Incim: A community-based algorithm for influence maximization problem under the linear threshold model, Inf. Process. Manag. **52**, 1188 (2016).

[13] E. Bagheri, G. Dastghaibyfard, and A. Hamzeh, An efficient and fast influence maximization algorithm based on community detection, in *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)* (IEEE, Piscataway, NJ, 2016), pp. 1636–1641.

[14] S. Fortunato, Community detection in graphs, Phys. Rep. **486**, 75 (2010).

[15] T. Martin, X. Zhang, and M. E. J. Newman, Localization and centrality in networks, Phys. Rev. E **90**, 052808 (2014).
[16] S. Brin and L. Page, The anatomy of a large-scale hypertextual web search engine, Comput. Netw. ISDN Syst. **30**, 107 (1998).
[17] L. Lü, T. Zhou, Q.-M. Zhang, and H. E. Stanley, The H-index of a network node and its relation to degree and coreness, Nat. Commun. **7**, 10168 (2016).
[18] G. Schoenebeck, B. Tao, and F.-Y. Yu, Think globally, act locally: On the optimal seeding for nonsubmodular influence maximization, Inf. Comput. **285**, 104919 (2022).
[19] P. Holme, Three faces of node importance in network epidemiology: Exact results for small graphs, Phys. Rev. E **96**, 062305 (2017).
[20] D. Chen, P. Du, B. Fang, D. Wang, and X. Huang, A node embedding-based influential spreaders identification approach, Mathematics **8**, 1554 (2020).
[21] A. Grover and J. Leskovec, node2vec: Scalable feature learning for networks, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2016), pp. 855–864.
[22] S. Rajeh and H. Cherifi, Ranking influential nodes in complex networks with community structure, PLoS One **17**, e0273610 (2022).
[23] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech.: Theory Exp. (2008) P10008.
[24] M. Rosvall and C. T. Bergstrom, Maps of random walks on complex networks reveal community structure, Proc. Natl. Acad. Sci. USA **105**, 1118 (2008).
[25] A. Lancichinetti, S. Fortunato, and F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E **78**, 046110 (2008).
[26] S. Fortunato and D. Hric, Community detection in networks: A user guide, Phys. Rep. **659**, 1 (2016).
[27] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, Epidemic processes in complex networks, Rev. Mod. Phys. **87**, 925 (2015).
[28] P. Grassberger, On the critical behavior of the general epidemic process and dynamical percolation, Mathematical Biosciences **63**, 157 (1983).
[29] M. E. J. Newman and R. M. Ziff, Efficient Monte Carlo Algorithm and High-Precision Results for Percolation, Phys. Rev. Lett. **85**, 4104 (2000).
[30] F. Radicchi, Predicting percolation thresholds in networks, Phys. Rev. E **91**, 010801(R) (2015).
[31] Y.-J. Zhang, K.-C. Yang, and F. Radicchi, Systematic comparison of graph embedding methods in practical tasks, Phys. Rev. E **104**, 044315 (2021).
[32] C.-E. Bichot and P. Siarry, *Graph Partitioning* (John Wiley & Sons, New York, 2013).
[33] G. Karypis and V. Kumar, METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices 1997 (unpublished).
[34] G. Karypis and V. Kumar, Multilevel algorithms for multi-constraint graph partitioning, in *SC'98: Proceedings of the 1998 ACM/IEEE Conference on Supercomputing* (IEEE, Piscataway, NJ, 1998), pp. 28–28.
[35] G. Karypis and V. Kumar, Multilevel k-way partitioning scheme for irregular graphs, J. Parallel Distrib. Comput. **48**, 96 (1998).
[36] G. García-Pérez, A. Allard, M Á. Serrano, and M. Boguñá, Mercator: uncovering faithful hyperbolic embeddings of complex networks, New J. Phys. **21**, 123033 (2019).
[37] G. Cordasco and L. Gargano, Community detection via semi-synchronous label propagation algorithms, in *2010 IEEE International Workshop on Business Applications of Social Network Analysis (BASNA)* (IEEE, Piscataway, NJ, 2010), pp. 1–8.
[38] P. Bonacich, Factoring and weighting approaches to status scores and clique identification, J. Math. Sociol. **2**, 113 (1972).
[39] F. Morone, B. Min, L. Bo, R. Mari, and H. A. Makse, Collective influence algorithm to find influencers via optimal percolation in massively large social media, Sci. Rep. **6**, 30062 (2016).
[40] V. Colizza, R. Pastor-Satorras, and A. Vespignani, Reaction–diffusion processes and metapopulation models in heterogeneous networks, Nat. Phys. **3**, 276 (2007).
[41] R. A. Rossi and N. K. Ahmed, The network data repository with interactive graph analytics and visualization, http://networkrepository.com.
[42] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas, Self-similar community structure in a network of human interactions, Phys. Rev. E **68**, 065103(R) (2003).
[43] A. Arenas, Urv email data, https://deim.urv.cat/∼alexandre.arenas/data/xarxes/email.zip.
[44] L. A. Adamic and N. Glance, The political blogosphere and the 2004 U.S. election: divided they blog, in *Proceedings of the 3rd International Workshop on Link Discovery* (ACM, New York, 2005), pp. 36–43.
[45] J. Kunegis, KONECT: the Koblenz network collection, in *WWW '13 Companion: Proceedings of the 22nd International Conference on World Wide Web* (ACM, New York, 2013), pp. 1343–1350.
[46] J. Kunegis, The konect project, http://konect.cc/networks/.
[47] T. Opsahl and P. Panzarasa, Clustering in weighted networks, Soc. Netw. **31**, 155 (2009).
[48] D. Bu, Y. Zhao, L. Cai, H. Xue, X. Zhu, H. Lu, J. Zhang, S. Sun, L. Ling, N. Zhang *et al.*, Topological structure analysis of the protein–protein interaction network in budding yeast, Nucleic Acids Res. **31**, 2443 (2003).
[49] J. Moody, Peer influence groups: identifying dense clusters in large networks, Soc. Netw. **23**, 261 (2001).
[50] A. Clauset, E. Tucker, and M. Sainz, The Colorado index of complex networks, 2016 (unpublished).
[51] A. L. Traud, P. J. Mucha, and M. A. Porter, Social structure of Facebook networks, Physica A **391**, 4165 (2012).
[52] V. Red, E. D. Kelsic, P. J. Mucha, and M. A. Porter, Comparing community structure to characteristics in online collegiate social networks, SIAM Rev. **53**, 526 (2011).
[53] R. A. Rossi and N. K. Ahmed, The network data repository with interactive graph analytics and visualization, in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (AAAI, Washington, 2015).
[54] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon, Superfamilies of evolved and designed networks, Science **303**, 1538 (2004).

[55] Uri alon lab: Collection of complex networks, http://wws.weizmann.ac.il/mcb/UriAlon/index.php?q=download/collection-complex-networks.

[56] T. Opsahl, F. Agneessens, and J. Skvoretz, Node centrality in weighted networks: Generalizing degree and shortest paths, Soc. Netw. **32**, 245 (2010).

[57] S. Kumar, F. Spezzano, V. S. Subrahmanian, and C. Faloutsos, Edge weight prediction in weighted signed networks, in *2016 IEEE 16th International Conference on Data Mining (ICDM)* (IEEE, Piscataway, NJ, 2016), pp. 221–230.

[58] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, and V. S. Subrahmanian, REV2: Fraudulent user prediction in rating platforms, in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (ACM, New York, 2018), pp. 333–341.

[59] J. Leskovec and A. Krevl, SNAP Datasets: Stanford large network dataset collection, 2014, http://snap.stanford.edu/data.

[60] J. Leskovec, J. Kleinberg, and C. Faloutsos, Graph evolution: Densification and shrinking diameters, ACM Trans. Knowl. Discov. Data **1**, 2 (2007).

[61] F. Radicchi, Who is the best player ever? a complex network analysis of the history of professional tennis, PLoS One **6**, e17249 (2011).

[62] https://github.com/JeffSackmann/tennis_atp.

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

[64] M. Newman, Network data, http://www-personal.umich.edu/~mejn/netdata/.

[65] 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).

[66] Hypertext 2009 dynamic contact network, http://www.sociopatterns.org/datasets/hypertext-2009-dynamic-contact-network.

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

[68] G Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. R. Gopinath, G. R. Wu, L. Matthews *et al.*, Reactome: a knowledgebase of biological pathways, Nucleic Acids Res. **33**, D428 (2004).

[69] L. Šubelj and M. Bajec, Software systems through complex networks science: Review, analysis and applications, in *Proceedings of the First International Workshop on Software Mining* (ACM, New York, 2012), pp. 9–16.

[70] M. Ripeanu, I. Foster, and A. Iamnitchi, Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design, arXiv:cs/0209028.

[71] J. Leskovec, D. Huttenlocher, and J. Kleinberg, Signed networks in social media, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (ACM, New York, 2010), pp. 1361–1370.

[72] J. Leskovec, D. Huttenlocher, and J. Kleinberg, Predicting positive and negative links in online social networks, in *Proceedings of the 19th International Conference on World Wide Web* (ACM, New York, 2010), pp. 641–650.

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

[74] M. Ley, The dblp computer science bibliography: Evolution, research issues, perspectives, in *String Processing and Information Retrieval* (Springer, Berlin, 2002), pp. 1–10.

[75] G. R. Kiss, C. Armstrong, R. Milroy, and J. Piper, *An Associative Thesaurus of English and Its Computer Analysis*, in The Computer and Literary Studies (Edinburgh University Press, Edinburgh, 1973), pp. 153–165.

[76] L. Šubelj and M. Bajec, Model of complex networks based on citation dynamics, in *WWW '13 Companion: Proceedings of the 22nd International Conference on World Wide Web* (ACM, New York, 2013), pp. 527–530.

[77] J. Leskovec, J. Kleinberg, and C. Faloutsos, Graphs over time: densification laws, shrinking diameters and possible explanations, in *Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (ACM, New York, 2005), pp. 177–187.