Journal of Complex Networks (2020) **2**, Advance Access Publication on 5 August 2019,
Issue Publication on 22 June 2020
doi: 10.1093/comnet/cnz029

Review

Influencer identification in dynamical complex systems

SEN PEI[†,‡]
Department of Environmental Health Sciences, Mailman School of Public Health,
Columbia University, 722 West 168th Street, New York, NY 10032, USA
[†]Corresponding author. Email: sp3449@cumc.columbia.edu

JIANNAN WANG[‡]
Research Institute of Frontier Science, Beihang University, No. 37 Xueyuan Road, Beijing 100191,
China and Levich Institute and Physics Department, City College of New York,
160 Convent Avenue, New York, NY 10031, USA

AND

FLAVIANO MORONE AND HERNÁN A. MAKSE
Levich Institute and Physics Department, City College of New York, 160 Convent Avenue, New York, NY 10031, USA
[‡]These authors contributed equally to this work.

Edited by: Jesus Gomez-Gardenes

[Received on 6 April 2019; editorial decision on 1 July 2019; accepted on 13 July 2019]

The integrity and functionality of many real-world complex systems hinge on a small set of pivotal nodes, or influencers. In different contexts, these influencers are defined as either structurally important nodes that maintain the connectivity of networks, or dynamically crucial units that can disproportionately impact certain dynamical processes. In practice, identification of the optimal set of influencers in a given system has profound implications in a variety of disciplines. In this review, we survey recent advances in the study of influencer identification developed from different perspectives, and present state-of-the-art solutions designed for different objectives. In particular, we first discuss the problem of finding the minimal number of nodes whose removal would breakdown the network (i.e. the optimal percolation or network dismantle problem), and then survey methods to locate the essential nodes that are capable of shaping global dynamics with either continuous (e.g. independent cascading models) or discontinuous phase transitions (e.g. threshold models). We conclude the review with a summary and an outlook.

Keywords: influencer identification; spreading dynamics; optimal percolation; k-core percolation; threshold models.

1. Introduction

A wide variety of phenomena in nature and society can be unified under the umbrella of dynamical complex systems. Important social and biological processes such as epidemic outbreaks in population [1], information diffusion in social media [2], signal transmission in brain networks [3] and dynamical evolution of ecosystems [4] all boil down to interactions among large numbers of building units of each system, and therefore can be properly described by dynamical models in complex networks [5–8]. In these

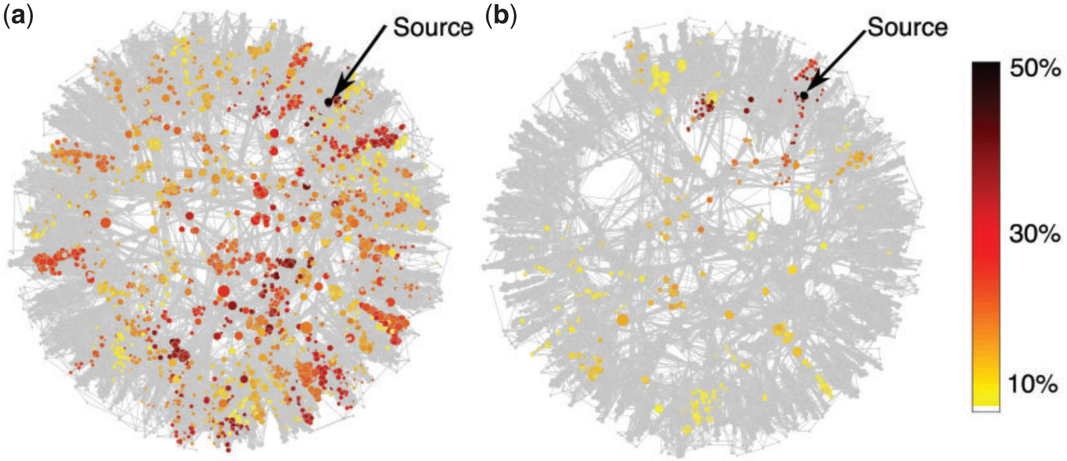

FIG. 1. Simulations of epidemic processes initiated from different origins. We use a susceptible-infected-removed (SIR) model to simulate epidemic outbreaks in a patient-to-patient contact network from multiple Swedish hospitals [18]. Given the same set of epidemiological parameters (infectious rate $\beta = 0.01$ and recover rate $\mu = 1$), outbreaks initiated from two origins have dramatically different outcomes. In (a) and (b), the epidemic sources have the same number of connections, but the source in (a) locates in a more central region with a higher k-core index. The colour of each node indicates the probability of infection during 1,000 independent realizations of the SIR model.

systems, complex interactions at microscopic scale lead to the abundant dynamical behaviours we observe at macroscopic level. As a result, understanding how network structure impacts the function of dynamical complex systems becomes a central topic in modern network science.

In network science, it has been well established that the collective dynamics of complex systems can be shaped by a small number of essential nodes, or influencers. For example, opinion leaders in social media are capable of influencing the public viewpoint on certain trending topics [9]; critical regions in brain are essential in the formation of memory networks [3, 10–12]; and keystone species in ecology are responsible for the integrity and stability of ecosystems [13–16]. Numerical simulations of epidemic processes have also demonstrated that the location of epidemic origin is critical for the final outbreak size [17] (see Fig. 1 for an example).

In general, influencers can be vaguely defined as the nodes that are disproportionately 'important' to the function of complex systems. However, in a given context, influencers may have a more specific definition: in social networks, influencers are opinion leaders who can influence a large number of people; in brain networks, influencers are important regions that maintain the connection across different functional parts; in ecological systems, influencers are keystone species whose extinction would collapse the network; and in epidemic spreading, influencers are superspreaders who transmit infectious diseases to a large population. In previous studies, abundant works exist dedicated to explore how to find influencers in a specific system. (For instance, in social science, various centrality measures have been developed to rank users' importance in social networks [19].) Due to its vast scope, here we do not attempt to summarize all relevant works, but instead focus on two important problems with wide applications.

- First, how to find the *structural* influencers whose removal would fragment the network? This problem, named optimal percolation [20] or network dismantle [21], is purely structural and does not involve with dynamical processes.

- Second, how to find the *dynamical* influencers who can lead to the largest cascading following a spread model? This problem, named influence maximization [22], depends on both network structure and dynamical rules.

In our following discussion, we refer to the above two problems uniformly as influencer identification. The specific definition of influencers is thus context-dependent. Solutions to these two problems can be applied in real-world applications ranging from maximization of marketing in social networks [22–24], optimization of immunization campaigns [25–27] to protection of networks under malicious attacks [28–30].

Real-world dynamical complex systems generally fall into two major classes:

- *Systems with only positive interactions.* For instance, in online social media, social ties can facilitate the spread of information among users [9]; in human population, physical contacts may transmit infectious diseases from person to person [31]; and in mutualistic ecosystems, cooperations between different species benefit their existence in ecology [32].
- *Systems with both positive and negative interactions.* For instance, in neural systems, synaptic connections can be either excitatory or inhibitory [33]; in gene regulatory networks, molecular regulators can activate or inhibit the expression of certain genes [34]; and in ecosystems, both mutualistic and predator–prey relationships coexist among different species [32].

For systems with only positive interactions, influencer identification can be defined using key topological structures such as the giant component (GC) [35–37] and k-core [38, 39]. On the contrary, systems with both positive and negative interactions do not admit the classical definitions of the GC and k-core, so the influencer identification problem in these systems need to be treated with a different theory. In this review, we only consider the former case where all links have positive interactions, and the case of inhibition/activation interactions will be treated elsewhere.

For structural influencer identification, the solution only depends on the network structure. However, for dynamical influencer identification, spread models can be further divided into two classes with continuous (second order) and discontinuous (first order) phase transitions. In regular percolation process [35] and ICMs [40], the GC emerges continuously from zero size as links are gradually occupied [41]. In contrast, in k-core percolation [42] and threshold models [43], k-core structure with non-zero size can appear abruptly as more nodes are activated [44, 45]. For these two types of dynamical models, approaches to find influencers are qualitatively different. We therefore discuss the influencer identification problem for models with continuous and discontinuous phase transitions separately.

Heuristically, influencers can be selected by picking vital individual spreaders one by one using a greedy approach, in which the influence of single nodes is estimated via Monte Carlo simulations [22] or various centrality measures [19]. However, influencer identification is intrinsically an NP-hard combinatorial optimization problem [22]. Therefore, a collective point of view that considers interactions among multiple spreaders is required. Recent progresses have translated the influencer identification problem into other closely related optimization problems such as message passing [46], belief propagation (BP) [47], optimal percolation [20], optimal decycling [21, 48] and explosive percolation (EP) [49]. These new approaches have enriched our understanding of feasible directions to tackle the influencer identification problem, and provided a number of sophisticated yet efficient methods that are applicable to large-scale complex systems. Classical centrality-based approaches have been extensively discussed in previous literature. As a result, in this survey, we focus on development from other approaches. Readers who are interested in centralities can find details in Ref. [50–52] and references therein.

The article is organized as follows. In Section 2, the GC and k-core structure, on which the influencer identification problem is defined, are introduced. In this section, we discuss the links between regular percolation and ICMs, as well as the relationship between k-core percolation and threshold models. In Section 3, we discuss the progresses in optimal percolation using collective influence (CI), optimal decycling, EP and large deviations of percolation in details. This section summarizes recent developments in finding the minimal set of nodes to collapse a network, that is, the structural influencer identification problem. In Section 4, approaches for models with continuous transitions using greedy search and message passing are presented. In Section 5, methods developed for threshold models with discontinuous transitions are reported. Section 4 and Section 5 survey the methods to solve the dynamical influencer identification problem, with focus on dynamical models with continuous and discontinuous transitions, respectively. Lastly, we conclude the review with an outlook of further directions in Section 6.

2. GC and k-core structure

The topological feature of a network can be characterized by important structures such as the GC and k-core. These concepts are fundamental in defining the problem of influencer identification in various dynamical processes. In this section, we introduce the regular percolation and k-core percolation processes, which are used to define the influencer identification problem, and elucidate their connections with commonly used spreading models.

2.1 *Percolation and ICMs*

The connectivity of a network is characterized by the number of nodes in the largest connected component, or the GC G_∞. In the random graph theory established by Paul Erdős and Alfréd Rényi in 1960s [35], the percolation process describes the emergence of G_∞ by gradually increasing the probability of connection between any pairs of nodes [53]. In its inverse process, the GC G_∞ of an initially connected network collapses as an increasing fraction q of nodes or links are removed. This removal process, termed site or bond percolation, leads to a continuous phase transition at a critical value of q, above which only fragmented clusters remain, as shown in Fig. 2(a).

For a network $G(V,E)$ with $N = |V|$ nodes and $M = |E|$ edges, we can use a vector $\boldsymbol{n} = (n_1, \ldots, n_N)$ to represent the configuration of whether a node i is removed ($n_i = 0$) or not ($n_i = 1$). After the removal of $q = 1 - \sum_{i=1}^{N} n_i/N$ fraction of nodes, we define the size of remaining GC $G_\infty(q)$ as the ratio of the number of nodes in G_∞ to the network size N. In the classical percolation theory [35], nodes are deleted randomly. At the critical value q_{rand}, G_∞ is completely dismantled and becomes negligible compared with the network size N in a continuous, or second order, phase transition. In thermodynamic limit $N \to \infty$, we have $\lim_{q \to q_{\text{rand}}^-} G_\infty(q) = 0$ and $G_\infty(q) = 0$ for $q \geq q_{\text{rand}}$. In real-world networks, the critical point q_{rand} upon random attack depends on the heterogeneity of network structure. In particular, using generation functions, it was proved that for random networks with a given degree distribution $P(k)$, q_{rand} is estimated by $\langle k \rangle / (\langle k^2 \rangle - \langle k \rangle)$, where $\langle k \rangle = \sum kP(k)$ and $\langle k^2 \rangle = \sum k^2 P(k)$ are the first and second moments of $P(k)$ [54]. This estimation predicts an extreme robustness to random attack, that is, $q_{\text{rand}} \approx 1$, for scale-free networks with a power law degree distribution $P(k) \propto k^{-\gamma}$, which are ubiquitous in real-world systems [55, 56]. Later, more accurate estimations using the reciprocal of the largest eigenvalue of the adjacency matrix and non-backtracking matrix were developed [57, 58].

In random (or regular) percolation, nodes are removed without considering their difference in structural importance. As a matter of fact, if nodes are removed strategically, G_∞ can be destructed well before

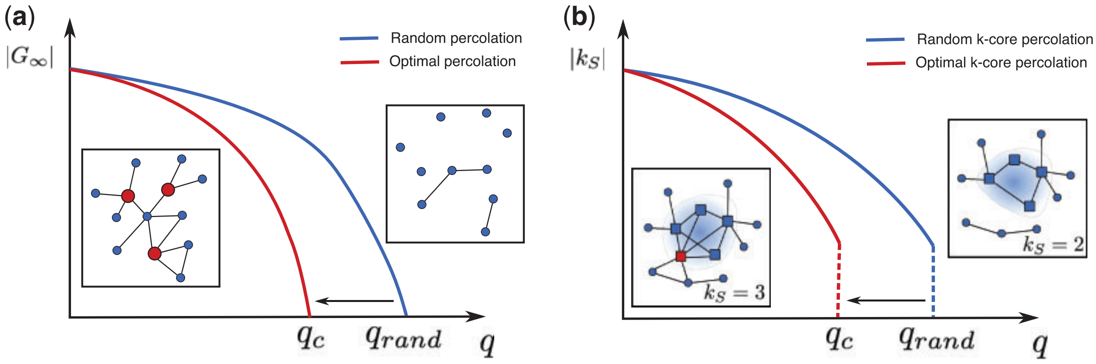

FIG. 2. Phase transition in percolation process. (a) The continuous transition of the GC size G_∞ after the removal of q fraction of nodes. q_{rand} and q_c are the critical values for random and optimal percolation, respectively. Insets show illustrations of a connected G_∞ and fragmented clusters. Red nodes are removed influencers. (b) The discontinuous transition of k-core size $|k_S|$ after the removal of q fraction of nodes. The critical values for random and optimal k-core percolation are marked by q_{rand} and q_c. Left inset shows the k-core of $k_S = 3$. After the removal of the red node, 3-core collapses and only 2-core is left, as shown in the right inset.

the removal of q_{rand} fraction of nodes. For example, G_∞ in scale-free networks is extremely vulnerable to targeted attacks on hubs [28, 29, 59]. The optimized process, deviating from the mean-field dynamics given by classical percolation theory, expedites the collapse of G_∞ and reduces the critical value of q. In statistical physics and network science, a number of works have explored the large deviations of percolation, that is, the deviations from the mean-field theory of percolation [60–64]. At q_{rand}, there exist a number of possible configurations $\boldsymbol{n}$ such that $G_\infty(\boldsymbol{n}) = 0$. As q decreases, fewer configurations satisfy $G_\infty(\boldsymbol{n}) = 0$, until at q_c where only one configuration $\boldsymbol{n}^*$ exists. Below q_c, there is no solution to $G_\infty(q) = 0$. Mathematically, $q_c = \min\{q \in [0,1] | G_\infty(q) = 0\}$. The optimal percolation, or network dismantle problem is to find the unique configuration $\boldsymbol{n}^*$ corresponding to the minimal q_c, and influencers are the nodes with $n_i = 0$ [20]. We note that the general framework of large deviations of percolation includes the optimal percolation problem as an extremal case [60, 63].

By definition, percolation process concerns the pure structural integrity of networks. Nevertheless, a class of spreading dynamics can be mapped to percolation process and therefore studied using percolation theory. One of these dynamics is described by independent cascade models (ICMs) [1, 7, 22]. In ICMs, an individual can be independently infected by any of his/her neighbours in the network. A spreading process starts from a set of infectious 'seeds' in a susceptible population. In each time step, a susceptible individual can become infected by each of his/her infected neighbours with a certain transmission probability. Infected individuals keep infectious for a time of period, and then become susceptible again or immune to infection. The spreading process stops when there is no new infections. In applications, most widely used models include the susceptible-infected (SI) model, the susceptible-infected-susceptible (SIS) model and the susceptible-infected-removed (SIR) model [65–67]. These models are widely used in the simulation [1, 59, 68–71], detection [72–77] and forecast [78–82] of infectious disease spread and information diffusion. ICMs are closely related to percolation: the dynamical spreading process of an ICM can be transformed to a bond percolation with a given occupation probability [37]. As a result, the outcome of a dynamical ICM can be mapped to the static final state of an equivalent percolation process. This mapping bypasses the need to run dynamical models and enables us to analyse the spreading process using tools and properties of the well-studied percolation problem.

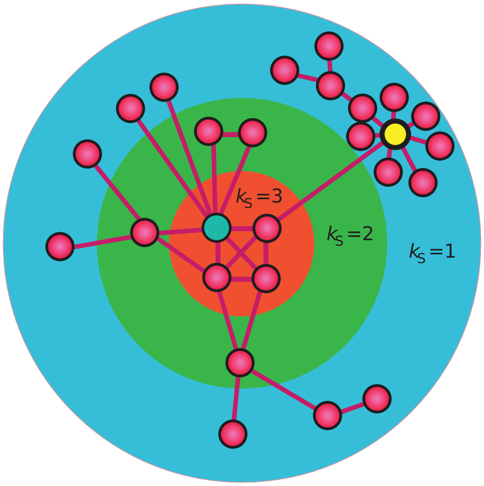

FIG. 3. An example of k-core decomposition. The highlighted blue and yellow nodes have the same degree $k = 8$, but with different k_S values. Figure is adapted from [17] under permission from Springer Nature.

2.2 *The k-core percolation and threshold models*

The k-core decomposition classifies networks into layers with increasingly dense connections. In a network, the k-core is defined as the largest subgraph whose nodes have at least k links [38, 39]. For example, the 1-core of a network is simply its GC; the 2-core is composed of all loops. Each node corresponds to a unique k-core index k_S that indicates the highest k-core it locates. The k-core index k_S is obtained through k-shell decomposition in which nodes are iteratively pruned according to their remaining degrees [83]. This process can be also viewed as a recursive calculation of the Hirsch-index h [84], in which a node is assigned index h if it has at least h neighbours with degree no smaller than h [85]. Nodes with low k_S values are located at the periphery of the network while the centre consists of nodes with high k_S values. An example of k-core decomposition is shown in Fig. 3. Recently, k-core percolation is generalized to multiplex networks [86].

The k-core structure provides higher-order information on network connectivity beyond GC, which can be viewed as a 1-core. Originally proposed on lattices in statistical physics [87], k-core percolation (or bootstrap percolation) describes the formation process of k-core in networks [44]. In a standard k-core percolation, nodes in a given network can be either active or inactive. Initially p fraction of nodes are activated; in later steps, inactive nodes with at least k active neighbours become activated recursively. In the final state, active nodes form the percolated k-core. The reversal process of k-core percolation depicts the destruction of k-core structure. Specifically, q fraction of nodes are removed from the network, and nodes with less than k neighbours are further recursively deleted. The final size of k-core $|k_S|$ is the fraction of nodes left.

The k-core percolation has many important variants developed independently in other disciplines. For instance, in sociology, Granovetter proposed the threshold model of collective behaviour in society in 1978 [43]. In the well-studied version of linear threshold models (LTMs), nodes are activated only when the number of active neighbours exceeds a predefined threshold value. The heterogeneous k-core percolation, in which each node is assigned a local threshold, is a generalization of the classical k-core percolation and a special case of LTM [88–90]. Within this framework, classical GC percolation can be viewed as a special case of LTM where the threshold of each node i is $k_i - 1$ (k_i is the degree of node i) [20]. Further, weights of interactions and non-linear threshold rules have been introduced to describe more complex dynamics [22, 45, 91]. Recently, the k-core was also applied as a precursor of the jamming transition in granular materials [92].

More recently, a generalized k-core percolation was proposed as a generalization of the leaf removal process [93]. In this k-leaf removal algorithm, nodes of degree smaller than k and their nearest neighbours together with all incident links are recursively pruned. The subgraph left after this pruning is called the Generalized k-core, or Gk-core. Similar as k-core percolation, the pruning procedure decomposes the network into layers of nested Gk-cores. However, as indicated by the authors, unlike k-core decomposition that classifies nodes according to their topological properties, the Gk-cores characterize a specific robustness of the network: it is actually the remained network after an epidemic that attacks weak individuals of degree less than k and their neighbours.

The fundamental difference of k-core percolation from GC percolation is that the k-core size $|k_S|$ could undergo a discontinuous, or first order, phase transition under certain circumstances. For example, in Fig. 2(b), the left inset illustrates the 3-core of the network. Upon the removal of the red node, the 3-core is completely destroyed, with only 2-core left as shown in the right inset. In this example, the 3-core disappears abruptly from a non-zero size. Such discontinuous phase transition stems from the threshold rule of percolation, and lies at the heart of catastrophic cascading failures in many real-world systems [94, 95]. A number of seminal works have explored the phase diagram and mechanism of transition in k-core percolation or threshold models [42, 44, 45, 91, 96, 97]. In particular, Watts modelled the global cascade on random networks using a linear threshold model and derived the critical condition for the discontinuous transition [45]. Goltsev *et al.* [44] found the hybrid phase transition in k-core percolation with a discontinuous emergence of k-core as well as a continuous emergence of GC. In Ref. [44], authors demonstrated the crucial role of 'corona', a subset of nodes in the k-core that have exactly k neighbours: a random removal of even one node from the corona will trigger the collapse of a vast region of the k-core around the removed node. Baxter *et al.* [42] further analytically derived the condition for the discontinuous transition of k-core in networks with arbitrary degree distributions. The abrupt jump from a k-core with non-zero size to its collapse can be mathematically explained by a bifurcation of the dynamical system describing the k-core percolation. In such bifurcation, a small change of parameters (e.g. fraction of removed nodes) leads to the discontinuous shift of the stable point from a non-zero solution (k-core with non-zero size) to a zero solution (no k-core). Such bifurcation-induced transition is also responsible for the global cascade and vulnerability in interdependent networks and network of networks [94, 98, 99].

Similar to optimal percolation, the configuration of node removal $\boldsymbol{n}$ can be optimized to induce early transition. For random k-core percolation, at the critical point q_{rand}, we have $\lim_{q \to q_{\text{rand}}^-} |k_S| > 0$, $\lim_{q \to q_{\text{rand}}^+} |k_S| = 0$ and $|k_S|(q) = 0$ for $q > q_{\text{rand}}$. The optimal k-core percolation problem is to find the unique configuration $\boldsymbol{n}^*$ for which $q_c = \min\{q \in [0,1] | \lim_{q' \to q^+} |k_S|(q') = 0\}$ (Fig. 2(b)). In some literature, this problem is also known as the minimal contagious set problem[100–106]. For threshold models, the influencer identification problem is to search for a given number of seeds that can lead to the maximal number of activated nodes.

3. Optimal percolation

Influence maximization is closely related to the optimal percolation problem. In addition, optimal percolation also provides a solution to the optimal immunization problem by dismantling the underlying network on which propagation occurs. Recently, within the message-passing framework, Morone and Makse developed an efficient algorithm, the CI, that gives a good approximation of optimal percolation

[20]. Later, better algorithms based on optimal decycling [21, 48] and EP [49] were proposed. In this section, we discuss these structural approaches to the influence maximization problem.

3.1 *Collective influence*

Considering a network G with N nodes and M edges, the vector $\mathbf{n} = (n_i, \ldots, n_N)$ encodes the configuration of whether node i is removed ($n_i = 0$) or not ($n_i = 1$). Denoting the fraction of removed nodes by $q = 1 - \sum_{i=1}^{N} n_i/N$, optimal percolation aims to find the minimal fraction q_c of nodes such that the GC G_∞ is fully dismantled. Within the message-passing framework, define message $\nu_{i\to j}$ as the probability that node i belongs to G_∞ without being connected to it through node j. Therefore, $\nu_{i\to j} = 1$ if and only if $n_i = 1$ and a least one of i's neighbours other than j is connected to G_∞. For a locally tree-like structure, the messages evolves by the following equations:

$$\nu_{i\to j} = n_i \left[1 - \prod_{m\in\partial i\setminus j} (1 - \nu_{m\to i})\right], \tag{1}$$

where $\partial i \setminus j$ denotes the immediate neighbours of i excluding j. Taking node j back into consideration, the probability that i is connected to the GC is then calculated as

$$\nu_i = n_i \left[1 - \prod_{m\in\partial i} (1 - \nu_{m\to i})\right]. \tag{2}$$

By linearizing Eq. (1) around the fixed point $\{\nu_{i\to j} = 0\}$, the stability of this solution is determined by the largest eigenvalue $\lambda(\mathbf{n}; q)$ of a linear operator $\mathcal{M}$. Specifically, $\mathcal{M}$ is the Jacobian of the system defined on $2M \times 2M$ directed edges as $\mathcal{M}_{m\to n, i\to j} \equiv \frac{\partial \nu_{i\to j}}{\partial \nu_{m\to n}}|_{\{\nu_{i\to j}=0\}}$. A few calculations show that the matrix $\mathcal{M}$ can be represented in terms of the non-backtracking (NB) matrix $\mathcal{B}$ [107] via

$$\mathcal{M}_{m\to n, i\to j} = n_i \mathcal{B}_{m\to n, i\to j}, \tag{3}$$

where the NB matrix is

$$\mathcal{B}_{m\to n, i\to j} = \begin{cases} 1 \text{ if } n = i \text{ and } j \neq m, \\ 0 \text{ otherwise.} \end{cases} \tag{4}$$

The matrix entry $\mathcal{B}_{m\to n, i\to j}$ is non-zero only when $(m \to n, i \to j)$ form a pair of consecutive NB directed edges, that is, $(m \to n, n \to j)$ with $m \neq j$. For NB edges, $\mathcal{B}_{m\to n, n\to j} = 1$.

Following the Frobenius theorem, the largest eigenvalue $\lambda(\mathbf{n}; q)$ is real and positive. The solution $\{\nu_{i\to j} = 0\}$ is stable if $\lambda(\mathbf{n}; q) \leq 1$. In this way, the optimal percolation problem can be solved by finding the optimal configuration $\mathbf{n}^*$ such that $\lambda(\mathbf{n}^*; q_c) = 1$. For $q < q_c$, all configurations lead to $\lambda(\mathbf{n}; q) > 1$. On the contrary, for $q > q_c$, there exist two different circumstances. For some non-optimal configurations, the macroscopic component still exists. On the other hand, there are also configurations such that $\lambda(\mathbf{n}; q) \leq 1$, which correspond to a fully fragmented network. As $q \to q_c^+$, the number of configurations satisfying $\lambda(\mathbf{n}; q) \leq 1$ gradually decreases and eventually vanishes at q_c, where the optimal configuration $\mathbf{n}^*$ is obtained. To develop a scalable algorithm, the eigenvalue can be approximated using the Power Method [108]. For a given number of iterations ℓ, the CI of node i can be defined as:

$$\mathrm{CI}_\ell(i) = (k_i - 1) \sum_{j \in \partial \mathrm{Ball}(i,\ell)} (k_j - 1), \tag{5}$$

where $\partial\mathrm{Ball}(i, \ell)$ is the frontier of the ball of radius ℓ in terms of shortest path centred around node i. By iteratively removing the node with largest CI, the largest eigenvalue of $\mathcal{M}$ can be minimized with high efficiency. After each removal, the CI score of every remaining node in the network is recalculated. This process continues until the network is fully fragmented, that is, $G_\infty \ll 1$. The optimal configuration $\mathbf{n}^*$ and q_c are estimated from this removal process.

For $q < q_c$, the network cannot be fully dismantled. In order to obtain the smallest GC, a greedy reinsertion procedure is performed starting from the optimal configuration $\mathbf{n}^*$. In the reinsertion procedure, an index $c(i)$ is define for each removed node. Specifically, $c(i)$ is the number of clusters that would be joined together if node i is put back in the network. Nodes with the smallest $c(i)$ score are iteratively reinserted until the fraction of removed nodes decreases to q.

The computational complexity of the CI algorithm is $O(N^2)$. In practice, it can be accelerated by limiting the calculation and update of CI inside the $(\ell + 1)$-ball around the removed node. In addition, the complexity can be further reduced to $O(N \log N)$ by sorting the CI scores in a heap structure [109], which makes it scalable to large networks. Simulation results on both synthetic and real-world social networks show that the CI algorithm outperforms the equal graph partitioning (EGP) immunization strategy [26] and frequently used heuristic metrics such as degree centrality, PageRank and k-core index. For a Twitter network with 469,013 users and a Mexico mobile communication network of 1.4×10^7 users, the CI algorithm achieves fully fragmentation with a smaller set of influencers [20] (see Fig. 4). For such massively large-scale networks, a variant of the CI algorithm can be applied without losing performance by removing a finite fraction of nodes instead of one node at each step.

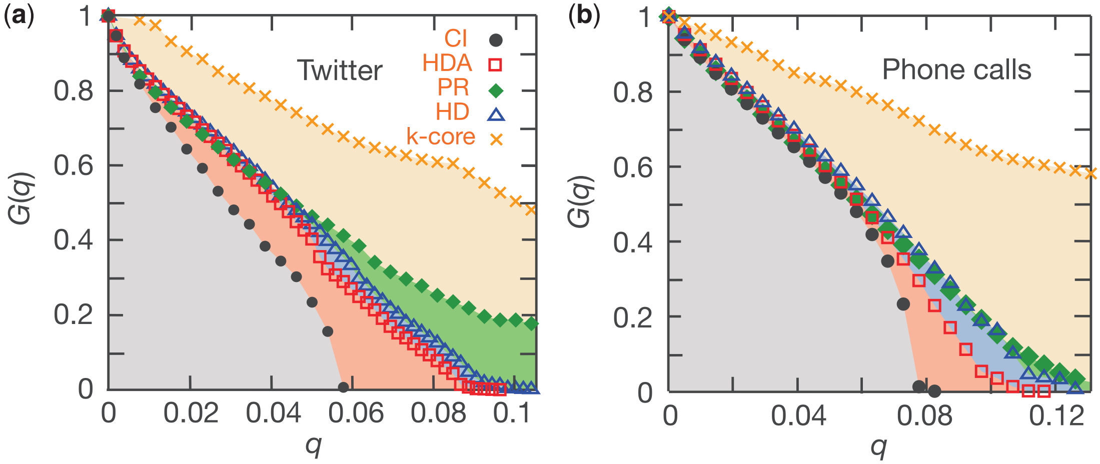

FIG. 4. Performance of CI in large-scale real social networks. GCs $G(q)$ for a Twitter network (a) and a social network of mobile phone users in Mexico (b) computed using CI, HDA (high degree adaptive), PR (PageRank), HD (high degree) and k-core strategies are compared. Figure is reused from [20] permitted by Springer Nature.

Given a finite radius ℓ, the CI algorithm is local in nature. To incorporate the influence of a node at the global level, Morone *et al.* [109] improved the CI algorithm using a message-passing approach and proposed the CI propagation algorithm (CI_P). As a variant of the CI algorithm in the limit of $\ell \to \infty$, the CI_P algorithm is able to reach the analytical optimal percolation threshold of random cubic graphs [110]. Another BP variant of CI algorithm, CI_BP, was also proposed. Combining the dynamics of the SIR model with message-passing updating rules, CI_BP achieves similar performance with CI_P. However, the improvements of CI_P and CI_BP over CI are made at the expense of increasing computational complexity from $O(N \log N)$ to $O(N^2 \log N)$. Kobayashi and Masuda recently developed an immunization algorithm for networks with community structure combining the CI algorithm and coarse graining procedure in which communities were regarded as supernodes [111]. From a mesoscopic scale, nodes connecting different communities can be identified at a cost of $O((N^2/N_C) \log N)$ (N_C is the number of communities). The optimal percolation problem was also studied on multiplex networks. Osat *et al.* [112] showed that characteristics in multiplex networks such as edge overlap and interlayer degree–degree correlation could profoundly change the properties of influencers. Neglecting the multiplex structure of a network would lead to significant inaccuracies about its robustness. In applications, the CI theory has been used to locate superspreaders of information in real-world social media [113], find sources of fake news in Twitter during the 2016 US presidential election [114, 115], single out critical regions in brain networks [10, 116], infer personal economic status [117], improve cooperation in evolutionary games [118] and control biological networks [119–122].

As demonstrated in Ref. [20], the optimal percolation problem can be mapped exactly onto the influence maximization problem for the LTM with threshold $k_i - 1$ (k_i is the degree of node i). As a result, the CI algorithm, designed for optimal percolation, also provides a solution to the influence maximization problem for this specific transmission model. For LTMs with other threshold values, the CI algorithm was generalized to solve the influence maximization problem with first-order transitions, which will be addressed later. In addition, a detailed discussion on the relation of the CI algorithm with the SIR model can be found in Ref. [20].

3.2 *Optimal decycling-based algorithms*

Recent works have shown that the optimal percolation problem is closely related to the optimal decycling problem, or minimum feedback vertex set (FVS) problem [21, 48]. A FVS is the set of nodes whose removal would break all the loops in the network [123]. The optimal decycling problem is, in fact, analogous to find the FVS with smallest number of nodes. The rationale behind the connection between optimal percolation and optimal decycling is that, for sparse random networks, short loops rarely exist in small connected components [124–126]. If the long loops in the GC are cut, the network will break into small tree fragments. As indicated by Braunstein *et al.* [21], the optimal decycling threshold q_c^{dec} acts as an upper bound of the optimal percolation threshold q_c. For random networks with light-tailed degree distribution (finite second moment), the minimal size of decycling set is equal to the minimal size of dismantling set in the limit $N \to \infty$. The optimal decycling problem is itself an NP-hard problem, but can be solved via BP algorithms approximately. Two approaches based on decycling algorithm were developed recently [21, 48]. Both of them apply a three-stage procedure: first decycle the network with minimal number of nodes, then break the tree into small components and finally reinsert some nodes to the network without increasing the size of the largest component. Compared with the CI algorithm, these two algorithms take into account the global topology of the network and achieve a better performance.

The belief-propagation-guided decimation (BPD) algorithm proposed by Mugisha and Zhou is based on the spin glass model of the FVS problem [127]. In order to transform the global acyclic constraint into local ones, a variable A_i, which takes the value 0, i or $j \in \partial i$, is assigned to each node [127]. If node i is removed from G, $A_i = 0$. Otherwise, $A_i = i$ if it is a root of a tree or $A_i = j$ if node i has a parental node j. Given a microscopic configuration $\mathbf{A} = \{A_1, A_2, \ldots, A_N\}$, the fraction of removed nodes is represented by:

$$q^{\text{dec}}(\mathbf{A}) = \frac{1}{N}\sum_{i=1}^{N} \delta^0_{A_i}, \tag{6}$$

where δ^l_n is the Kronecker delta function ($\delta^l_n = 1$ if $n = l$ and 0 otherwise). For each edge (i,j) in the network, an edge factor $C_{ij}(A_i, A_j)$ is defined as [127]:

$$\begin{aligned} C_{ij}(A_i, A_j) = \delta^0_{A_i}\delta^0_{A_j} + \delta^0_{A_i}(1 - \delta^0_{A_j} - \delta^i_{A_j}) + \delta^0_{A_j}(1 - \delta^0_{A_i} - \delta^j_{A_i}) \\ + \delta^j_{A_i}(1 - \delta^0_{A_j} - \delta^i_{A_j}) + \delta^i_{A_j}(1 - \delta^0_{A_i} - \delta^j_{A_i}). \end{aligned} \tag{7}$$

The edge factor $C_{ij}(A_i, A_j)$ is either 1 or 0. The edge (i,j) is regarded as satisfied if $C_{ij}(A_i, A_j) = 1$, and unsatisfied otherwise. For a configuration $\mathbf{A}$, if all edges in a network G are satisfied, we define $\mathbf{A}$ as a solution of G. The definition of satisfied edges relaxes the original problem of acyclic components to allow at most one cycle in the remained components. Indeed, it has been proven that all remaining nodes in a graph for a solution $\mathbf{A}$ form a subgraph consisting of several components that each contains at most one cycle. Considering all solutions of the network, a partition function of the system is defined as:

$$Z(\mu) = \sum_{\mathbf{A}} e^{\mu N(1-q^{\text{dec}}(\mathbf{A}))} \prod_{(i,j)\in G} C_{ij}(A_i, A_j), \tag{8}$$

where μ is the inverse of temperature. At the limit of zero temperature, the partition function is contributed exclusively by the optimal configuration $\mathbf{A}^*$ with the minimal fraction q^{dec}_c.

Under locally tree-like assumption, the marginal probability $q^0_i(t)$ for node i to be removed from the remaining network $G(t)$ can be calculated through iterations of a set of BP equations [48]. At each time step t, the BP equations are iterated for a given number of rounds and the removal probability q^0_i is calculated for each node. The node with the highest probability q^0_i is removed from the network even if the BP equations do not converge to a fixed point. The process stops when the network becomes acyclic. If the largest component G_∞ remains extensive, it can be further fragmented by iteratively deleting nodes that lead to the smallest GC. The BPD algorithm can be well applied to networks with rare short loops. However, for a large number of networks with abundant communities, the nodes in FVS set are usually more than necessary to dismantle the network stricture. Therefore, a reinsertion process can be proceeded without significantly increasing the size of G_∞. This process can be done through a greedy algorithm, in which the nodes that cause the least increase in G_∞ are reinserted one after another until the size of G_∞ reaches a predefined threshold.

The BPD algorithm is scalable to large networks with a computational complexity of $O(N \log N)$. Simulations on random network ensembles and real-world networks indicate that the BPD algorithm is superior to the CI algorithm in optimal percolation problem (see Fig. 5). However, as shown in Ref. [109], the BPD algorithm is relatively slower than the CI algorithm. For large random Erdős–Rényi (ER) networks and scale-free networks, the BPD algorithm manages to fragment the network by removing a smaller set of nodes compared with CI algorithm. In particular, the percolation threshold is close

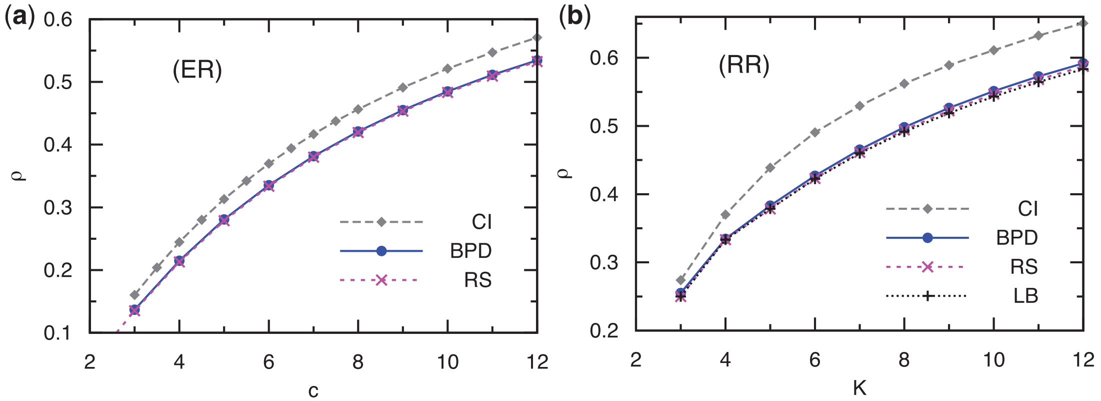

FIG. 5. Performance of the BPD algorithm in ER and RR networks. Fraction of removed nodes ρ to break ER random networks of mean degree c (a) and regular random networks of degree K (b). Diamonds show the results obtained from CI. Crosses represent results of the replica-symmetric (RS) mean-field theory. Plus symbols in (b) show the mathematical lower bound (LB) on the minimum size of target nodes. Figure reuse from [48] is permitted by American Physical Society.

to the minimal value predicted by the replica-symmetric (RS) mean-field theory [110, 127]. In the CI algorithm, the size of G_∞ decreases almost linearly with the increase of q. In contrast, G_∞ features an abrupt collapse under the BPD algorithm. This results from the intrinsic nature of the FVS problem and the efficiency of tree dismantling. With the existence of such collapse, the BPD process can work as an efficient attack strategy, leaving no warning to the system before its total failure. In a recent work on dismantling efficiency and network fractality [128], it was found that the BPD algorithm outperforms the CI algorithm no matter whether the network is fractal or not, while the CI algorithm works better on non-fractal networks, which have high ratios of long-range shortcuts to short-range connections.

Braunstein *et al.* [21] considered the optimal decycling problem from a different point of view. In this work, the optimal percolation problem was named as network dismantle. Noticing that a network is acyclic if and only if its 2-core is empty, authors mapped the decyling process to a 2-core percolation. Assume a set of nodes $S \subset V$ are initially removed from the network. The 2-core percolation can be described by the evolution of time-dependent binary variables $x_i^t(S)$ for $1 \leq i \leq N$. Starting from the initial setting $x_i^0(S) = 1$ for removed nodes $i \in S$ and $x_i^0(S) = 0$ for $i \notin S$ at $t = 0$, the evolution follows [21]

$$x_i^{(t+1)} = \begin{cases} 1 & \text{if } x_i^t(S) = 1 \\ \mathbb{I}\left[\sum_{j\in\partial i}\left(1 - x_j^t(S)\right) \leq 1\right] & \text{if } x_i^t(S) = 0, \end{cases} \tag{9}$$

where the indicator function $\mathbb{I}$ is 1 if the argument is true and 0 otherwise. As x_i^t can only change from 0 to 1, the equations admit a fixed solution $x_i^*(S)$ as $t \to \infty$. In particular, $x_i^*(S) = 0$ iff i belongs to the 2-core of $G \setminus S$. If $x_i^*(S) = 1$ for all nodes, $G \setminus S$ contains no loops and S is called a decycling set. To find the minimal decycling set, it is convenient to introduce the probability distribution over decycling sets S using the Boltzmann distribution in statistical physics

$$\hat{\eta}(S) = \frac{1}{Z(\mu)} e^{\mu|S|} \prod_{i\in V} \mathbb{I}\left[x_i^*(S) = 1\right], \tag{10}$$

where $|S|$ is the number of nodes in S, μ is the inverse temperature and $Z(\mu)$ is the partition function that normalizes the distribution. The minimal size of decycling sets can be calculated in the zero-temperature limit: $q_c^{\mathrm{dec}} = \lim_{\mu\to-\infty} \frac{1}{N\mu} \ln Z(\mu)$.

Since x_i^* depends on S in a global way, it is difficult to compute $Z(\mu)$ directly. To solve this problem, authors transformed the global constraint $\prod_{i\in V} \mathbb{I}\left[x_i^*(S) = 1\right]$ to its local equivalent. The node removal process in 2-core percolation can be described by an integer $t_i(S) = \min\{t : x_i^t(S) = 1\}$ defined for each node i, which encodes the time when node i is removed from the network. For $i \in S$, it is straightforward that $t_i(S) = 0$. For $i \notin S$, $t_i(S)$ depends locally on its neighbours according to

$$t_i(S) = \phi_i(\{t_j\}_{j\in\partial i}) = 1 + \max\nolimits_2(\{t_j(S)\}_{j\in\partial i}), \tag{11}$$

where the function $\max_2$ returns the second largest value in its argument. Under this parameterization, the partition function can be rewritten as

$$Z(\mu) = \sum_{\{t_i\}} e^{\mu \sum_i \psi_i(t_i)} \prod_{i\in V} \mathbb{I}[t_i < \infty]\mathbb{I}\left[t_i = \phi_i(\{t_j\}_{j\in\partial i})\right], \tag{12}$$

where $\psi_i(t_i) = \mathbb{I}[t_i = 0]$.

The exact computation of Eq. (12) is NP-hard. In calculation, a simplification of the partition function in Eq. (12) can be performed by restricting t_i to be no larger than T. All values t_i larger than T are regarded as infinity. Under this simplification, trees with diameters larger than $T + 1$ are considered to be part of a long cycle. Given a large enough T, the effect of this simplification is negligible. For locally tree-like graphs, the partition function can be computed by the cavity method [129, 130], in which 'messages' are exchanged between neighbouring nodes. For each link $i \to j$, a message $\eta_{ij}(t_i, t_j)$ as a function of activation times t_i and t_j is introduced. The messages satisfy the self-consistent BP equations [21]:

$$\eta_{ij}(t_i, t_j) \propto \sum_{\{t_k\}_{k\in\partial i\backslash j}} e^{\mu\psi_i(t_i)} \mathbb{I}\left[t_i = \phi_i(\{t_k\}_{k\in\partial i})\right] \prod_{k\in\partial i\backslash j} \eta_{ki}(t_k, t_i). \tag{13}$$

As the temperature approaching zero ($\mu \to -\infty$), probabilities of the messages $\eta_{ij}(t_i, t_j)$ in the BP equations concentrate on the solution to Eq. (9) that minimizes the cost function $\sum_i \psi_i(t_i)$. To develop an algorithm that finds the optimal decycling set, a slightly different cost function is used: $\psi_i(t_i) = \mathbb{I}[t_i = 0] + \varepsilon_i(t_i)$, where $\varepsilon_i(t_i)$ is a randomly chosen small cost. Further, the 2-core percolation process is relaxed to allow $t_i \geq 1 + \max_2(\{t_j\}_{j\in\partial i})$ in Eq. (9). Define $h_i(t_i)$ as the minimal cost to dismantle the 2-core under the condition that node i is removed at t_i. The optimal decycling set is determined by $S^* = \{i \in V | t_i^* = 0\}$, where $t_i^* = \arg\min h_i(t_i)$. In calculation, $h_i(t_i)$ can be computed using Min-Sum algorithm, which is derived at the zero temperature limit of BP equations. Concretely, messages $h_i(t_i)$ are solved by iterating a set of equations [21]. In most cases, convergence can be reached within a small number of iterations, with a computational complexity $O(MT)$ in each iteration. In case the Min-Sum equations do not converge, a reinforcement procedure is applied to damp the system [131].

In the acyclic network $G\backslash S^*$, there may still exist some extensive tree components. These large trees can be fragmented efficiently via a greedy tree-breaking procedure with computational complexity of $O(N(\log N+T))$. In addition, for networks containing many short loops, a reverse greedy (RG) reinsertion procedure is applied to recover the nodes that do not increase the size of the GC, as performed in the CI and BPD algorithms. The computational cost of this RG procedure is $k_{\max}C' \log(k_{\max}C')$, where $k_{\max}$ is the maximal degree and C' is the upper bound of G_∞ size.

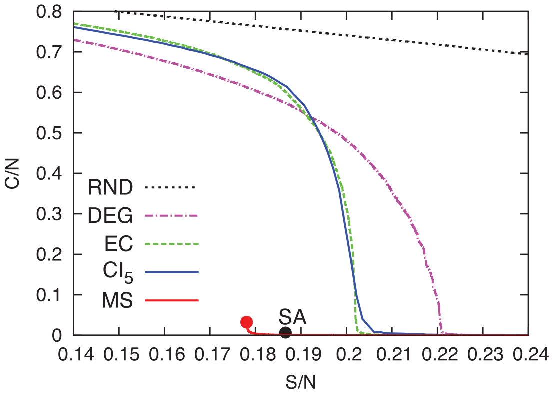

FIG. 6. Performance of the Min-Sum algorithm in an ER network. Relative size of the largest component C/N as a function of the fraction of nodes S/N removed from an ER network (size N = 78,125 and average degree d = 3.5). Comparisons are performed for the Min-Sum algorithm (MS), random (RND), adaptive largest degree (DEG), adaptive eigenvector centrality (EC), adaptive CI and simulated annealing (SA). Figure reuse from [21] is permitted by National Academy of Sciences.

Simulations on both synthetic and real-world social networks demonstrate the effectiveness of this decycling-based algorithm. For an ER random graph of size $N = 78,125$ and average degree $d = 3.5$, the G_∞ size deceases to 0.032 when 17.81% of nodes are removed. Compared with metrics of degree centrality, eigenvector centrality and the CI algorithm with $\ell = 5$, it was found that the three-stage algorithm is superior in dismantling the GC (see Fig. 6). The Monto Carlo-based simulated annealing (SA) algorithm gives a competitive result. However, its computational complexity is much higher. For the same Twitter network analysed in Ref. [20], the Min-Sum algorithm with RG performs equally well with SA, removing only 3.4% of nodes to break the GC (smaller than 1,000 nodes). In comparison, CI needs to remove 4.6% of nodes to achieve the same fragmentation performance.

Inspired by the decycling-based algorithm, a simple and faster heuristic algorithm with complexity $O(N)$, CoreHD, was developed [132]. Starting from the 2-core of a network G, CoreHD recursively removes nodes with the highest degree in the 2-core until G is fully dismantled. Despite its simpleness, CoreHD is reported to perform better than the CI algorithm. Specially, for large random networks, the performance of CoreHD is close to the theoretical solution predicted by replica-symmetry and 1RSB approximation [100]. In addition, this simple algorithm is amenable to rigorous analysis, performing well even on loopy networks which are not accessible for typical message-passing algorithms. In a recent work by Schmidt *et al.* [133], the CoreHD algorithm was analysed rigorously by translating the node removal in the CoreHD algorithm to a random process on the degree distribution of the network. The mapped dynamics, described by a set of coupled non-linear ordinary differential equations, characterize the behaviour of the CoreHD algorithm on random graphs. In the analysis, new upper bounds on the size of the minimal contagious sets in random graphs were proposed, which improves the best known results [100, 110]. The CoreHD analysis also inspired an improved heuristic algorithm, WEAK-NEIGHBOR, that works for both optimal percolation and k-core percolation [133]. Details of this algorithm will be introduced in the next section.

3.3 *EP-based immunization*

Another approach of optimal percolation was developed by Clusella *et al.* [49] based on EP. In contrast with ordinary bond percolation which usually exhibits second- or higher-order phase transitions, EP features an unusual threshold behaviour—an explosive emergence of the GC at the critical point [134–138]. To obtain an explosive transition, Achlioptas *et al.* [134] proposed a modified edge addition procedure, wherein, at each step, two candidate edges are chosen randomly, but only one of them is actually occupied. Given the weight of a node measured by the size of the connected component it belongs to, the edge with the minimal sum or product of nodes' weights is selected. These two procedures are referred to as the min-cluster and min-product rule. Compared with the random occupation of edges in ordinary percolation, the min-cluster or min-product rule favours the connection between small components, hereby suppresses the generation of an extensive component.

The explosive immunization (EI) algorithm adopts an inverse strategy that starts from a configuration where all nodes are virtually removed ($q = 1$). Then less 'dangerous' nodes are progressively unvaccinated. The procedure is performed in two schemes for $q > q_c$ and $q < q_c$, each of which uses a score to rank nodes in terms of their suitability to be unvaccinated. Similar to the construction of EP, in each time step, m candidates (typically $m \approx 10^3$) are randomly selected. For $q > q_c$, the node with the lowest blocking ability (the weakest blocker) is put back into the network. The blocking ability is quantified by a score $\sigma_i^{(1)}$, which is a synthesis of the size of clusters it would join and its local effective connectivity. Specifically, the score $\sigma_i^{(1)}$ is defined as [49]: $\sigma_i^{(1)} = k_i^{(\text{eff})} + \sum_{\mathcal{C}\subset\mathcal{N}_i}(\sqrt{|\mathcal{C}|} - 1)$, where $\mathcal{N}_i$ is the set of all components connected to node i and $|\mathcal{C}|$ is the size of a component $\mathcal{C}$. $k_i^{(\text{eff})}$ measures the 'effective' connectivity of node i based on the local structure of its neighbourhood and can be determined by a set of closed equations [49]: $k_i^{(\text{eff})} = k_i - L_i - M_i(\{k_j^{(\text{eff})}\}_{j\in\partial i})$, where k_i is the degree of node i, L_i is the number of leaves in the neighbourhood of node i and M_i returns the number of strong hubs. The strong hubs are defined recursively as nodes with $k_i^{(\text{eff})}$ larger than a threshold value (set as 6 in applications). The terms L_i and M_i are subtracted from k_i since leaves have no contribution to connectivity and hubs are more likely to be removed in explosive immunization.

In the first part of the EI algorithm, the node with the lowest $\sigma_i^{(1)}$ score among m candidates is unvaccinated in each iteration. This process eventually reaches a critical fraction of immunized nodes q_c where the G_∞ size exceeds a small threshold value. In the region of $q < q_c$, however, the same procedure will lead to an abrupt jump of the G_∞ size when two large components are joined together. As a consequence, in the second part at $q < q_c$, another score $\sigma_i^{(2)}$ is used to suppress such explosive growth of the GC. The definition of $\sigma_i^{(2)}$ reads [49]

$$\sigma_i^{(2)} = \begin{cases} \infty & \text{if } G_\infty \subsetneq \mathcal{N}_i, \\ |\mathcal{N}_i| & \text{else, if } \arg\min_i |\mathcal{N}_i| \text{ is unique}, \\ |\mathcal{N}_i| + \epsilon|\mathcal{C}_2| & \text{else}, \end{cases} \tag{14}$$

where $|\mathcal{N}_i|$ is the number of components connected to i, $\mathcal{C}_2$ is the second largest component in $\mathcal{N}_i$ and ϵ is a small positive number. According to the score $\sigma_i^{(2)}$, the selection is made only among the neighbourhood of G_∞. The candidate with the smallest number of neighbouring components is favoured; if it is not unique, the one with the smallest $|\mathcal{C}_2|$ is selected. This process is proceeded recursively until the fraction of vaccinated nodes q reaches the expected value.

Using the Newman–Ziff percolation algorithm in identifying susceptible components [139], the explosive immunization algorithm is computationally efficient, which scales as $O(N \log N)$. In addition, it can be accelerated further by considering a small number of candidates. Simulations on both synthetic and real-world networks indicate that the explosive immunization algorithm outperforms the CI algorithm (see Fig. 7). As a matter of fact, it achieves the smallest percolation threshold q_c except for the BP algorithms in Ref. [21, 48].

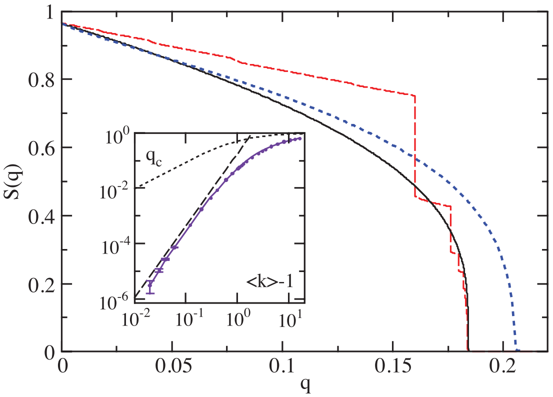

FIG. 7. Performance of the EI algorithm in an ER network. Relative size of the largest clusters $S(q)$ after q fraction of nodes are removed from ER networks with size $N = 10^6$ and average degree $\langle k \rangle = 3.5$. The red dashed curve is obtained by EI with score $\sigma^{(1)}$. The black continuous curve is obtained with $\sigma^{(2)}$ for $q < q^*$ ($S(q^*) = 1/500$). The blue dotted line is the result of CI. The inset shows the relationship between q_c and $\langle k \rangle - 1$. The dotted line is the result for random strategy. Figure reuse from [49] is permitted by American Physical Society.

3.4 *Graph partition-based algorithm*

In an earlier work, the optimal immunization problem was solved by an EGP immunization strategy based on the heuristic optimal partitioning of graphs [26]. In EGP, the network is fragmented into small connected clusters of approximately equal size. In a targeted attack on high-degree nodes, clusters after fragmentation have a broad distribution of sizes, including many small clusters. The targeted strategy may select high-degree nodes in these small clusters, which are unnecessary in breaking down the network. The EGP method avoids fragmenting small clusters, as the clusters all have similar sizes. In the EGP method, a network is first separated into two components with arbitrary size ratio by a minimal number of separators, solved using the nested dissection (ND) algorithm [140]. Then the network can be partitioned into any desirable number of same size clusters by applying ND algorithm recursively. This greedy graph-partitioning strategy provides 5–50% improvement over the targeted strategy on model networks and real-world networks.

The original network dismantle problem was recently extended to a generalized network dismantle (GND) problem in which the cost of removing a node is considered [141]. In real-world systems, attacking important nodes typically requires a high cost as they are usually well protected. The GND problem seeks to find a set of nodes whose removal would fragment a network at the minimal cost.

Authors solved this problem by recursively applying node-weighted partition, that is, partition a network into two parts of same size by removing a minimal number of edges. Specifically, define $v_i = +1$ if node i belongs to a subgraph M and $v_i = -1$ if node i belongs to its complement $\bar{M}$. Assuming that the cost of cutting a link (i,j) equals the cost of removing nodes i and j, a node-weighted spectral cut objective function was proposed [141]:

$$\frac{1}{2}\sum_{i,j} -\frac{1}{2}(v_i v_j - 1)A_{i,j}(w_i + w_j - 1), \tag{15}$$

where A is the adjacency matrix, and w_i is the cost of removing node i. The optimization problem was then written in matrix notation as minimizing $v^T L_w v/4$ subject to $\sum_i v_i = 0$, where $L_w = D_B - B$ is the node-weighted Laplacian of the matrix $B = AW + WA - A$ (W and D_B are diagonal matrices with elements $W_{ii} = w_i$ and $(D_B)_{ii} = \sum_j B_{ij}$.)

The problem with integer constraint $v_i \in \{+1, -1\}$ is difficult to solve. As a result, the problem is relaxed to allow a real number $v_i \in \mathbf{R}$. For the relaxed problem, the solution of v is analytically given by the second-smallest eigenvector of L_w, denoted by $v^{(2)}$. To approximate this solution, the matrix L_w is transformed so that $v^{(2)}$ becomes the second-largest eigenvector. The eigenvector problem is solved by power iteration, with the initial vector set perpendicular to the largest eigenvector of the transformed matrix. Once $v^{(2)}$ is obtained, the separating edges are those connecting nodes with $v_i \geq 0$ to nodes with $v_i < 0$. The set of nodes to be removed are optimized to cover all separating edges with minimal cost, which is transformed to the weighted vertex cover problem [142]. Finally, a reinsertion procedure is applied to find the nodes that are not necessary to fragment networks.

The GND algorithm has complexity $O(N \log^{2+\epsilon}(N))$, which can be applied to large-scale networks. For non-unit costs, the GND algorithm outperforms current state-of-the-art; for unit cost, it performs better than or comparable to state-of-the-art [141].

3.5 *Large deviations of percolation*

The optimal percolation problem can be studied within the framework of large deviations of percolation. Generally, in the BP equations that describe the percolation process, the *inverse* temperature β in the Boltzmann distribution of configurations $\mathbf{n}$, $e^{-\beta\mathcal{E}(\mathbf{n})}$ ($\mathcal{E}(\mathbf{n})$ is energy defined by the size of GC for $\mathbf{n}$), controls the deviation of dynamics from random percolation. For instance, an infinity temperature ($\beta = 0$) corresponds to the random scenario, where each configuration is equally possible. As the temperature decreases, the dynamics start to deviate from the random scenario to more extreme cases: the distribution of configurations will concentrate on rare configurations with lower energy, that is, smaller GC. Particularly, at zero temperature $\beta \to \infty$, only the configuration with the smallest GC exists with non-zero probability. In this way, the optimal percolation problem can be interpreted as an extreme case of the large deviations of percolation.

Recently, properties of large deviations of percolation have been analysed using Monte Carlo Markov Chains [61] and BP [62]. In particular, Bianconi [60, 62] developed a large deviation theory of percolation that characterizes the response of a sparse network to rare events. This general theory contains both continuous transitions observed for random initial damage and discontinuous transitions corresponding to rate configurations of the initial damage that suppresses the GC size. This large deviation theory of percolation was also generalized to multiplex networks [63], based on which a new metric, sageguard centrality, was developed to single out the nodes that control the response of the entire multiplex network to random damage [64]. It was found that the sageguard centrality correlates well with nodes in the optimal percolation problem.

3.6 *Summary*

It is interesting that the optimal percolation, or network dismantle problem, can be solved from quite different approaches: the CI algorithm optimizes the stability of zero solution by minimizing the spectral radius of the NB matrix; the BPD and network dismantle algorithms aim to optimally remove cycles in the network; the EI algorithm attempts to gradually identify less vital nodes so that an explosive collapse of network would occur if the remaining critical nodes are attacked; the EGP and GND algorithms work by recursively partitioning the network into equal-size components; and large deviations of percolation considers the rare events deviated from random percolation. In terms of implementation, CI proceeds as a

TABLE 1 *Summary of methods developed for optimal percolation. N is the network size and M is the number of links. 'NR' stands for 'not reported'.*

Name	Description	Complexity	References
Collective Influence (CI)	Stability of the NB matrix, greedy approach, easy to interpret and implement, adapted in real-world problems	$O(N \log N)$	[20]
Network dismantle	Optimal decycling, belief-propagation approach, Min-Sum algorithm, solve by iteration until convergence, compute the optimal node set simultaneously	$O(MT)$ per iteration	[21]
Belief-propagation-guided decimation (BPD)	Minimum feedback vertex set, spin glass model, belief-propagation approach, no convergence needed in BP iteration, select nodes iteratively	$O(N \log N)$	[48]
Explosive Immunization (EI)	Explosive percolation, iteratively select less important nodes from candidates, based on score defined for each node	$O(N \log N)$	[49]
Equal graph partitioning (EGP)	Recursively partition networks into clusters of similar size, avoid breaking small clusters	NR	[26]
Generalized network dismantle (GND)	Consider costs of removing nodes, node-weighted partition, recursive equal-size partition, solved by spectral properties of a Laplacian and weighted vertex cover	$O(N \log^{2+\epsilon}(N))$	[141]

greedy adaptive algorithm, which is straightforward to implement; the BPD, network dismantle algorithm and large deviations of percolation need to iterate BP or Min-Sum equations to find the solution; the EI algorithm iteratively selects unvaccinated nodes from a number of candidates; and the EGP and GND algorithms apply graph partition recursively with different techniques. Most of these algorithms require a reinsertion process that excludes unnecessary nodes from the optimal node set. In essence, to solve an intrinsically global optimization problem, most approaches have to transform it to another problem that can be solved locally. For instance, CI defines a centrality based on local structure; the BP equations in the BPD and network dismantle algorithms incorporate local constraints compatible with the global constraints; the score calculation in the EI algorithm depend on local connectivity; and the EGP and GND algorithms are designed to recursively partition smaller local clusters. More features of these algorithm are summarized in Table 1.

4. Dynamics with continuous transitions

The problem of influencer identification in ICMs was originated from the work of Domingos and Richardson [24, 143], who aimed to advertise a product though viral marketing. Instead of viewing market as a set of independent entities, they treated it as a networked system where the potential profit contributed by a customer is mostly determined by his/her interactions with others. This problem was later formalized by Kempe *et al.* [22] into a well-defined combinatorial optimization problem: considering an ICM in a network G and an integer k, how to find the optimal set of k seeds that initiates the largest scale propagation? The intrinsic difficulty of this problem is rooted in the exponentially growing configuration space with k. In fact, it was proven to be among the class of the hardest optimization problems—NP hard [22], and thus can only be solved approximately via heuristic approaches in polynomial time.

4.1 *Greedy algorithms*

One of the most intuitive solutions is to use greedy algorithm that selects the k most influential single spreaders to approximate the optimal set of influencers. In this approach, the influence of single influencers can be estimated by averaging a large number of Monte Carlo simulations of spreading processes initiated by each node. As proposed in Kempe *et al.* [22], the optimal set of influencers S is obtained by recursively adding the node that leads to the largest marginal increase to the total influence. The influence function $\sigma(S)$, defined as the expected number of active nodes given the seed set S, can be calculated by Monte Carlo simulations. The marginal contribution of an individual influencer i, $\sigma_S(i)$, can then be computed through $\sigma_S(i) = \sigma(S \cup \{i\}) - \sigma(S)$. For a general class of spreading models including ICMs, the influence function $\sigma(S)$ was proven to satisfy the characteristic of the so-called submodularity [144, 145]—a function $\sigma(\cdot)$ is submodular if the marginal gain from adding an element to a set S is at least as high as the marginal gain from adding the same element to a superset of S. In 1978, Nemhauser *et al.* mathematically proved that, for problems with submodular property, a greedy heuristic always finds a solution whose value is at least $1 - [(K-1)/K]^K$ times the optimal value [144, 145]. Here, K is the size of seed set. This bound has a limiting value of $1 - 1/e$, which is independent of the size of network or seed set. Leveraging on this theoretical result, the simple greedy algorithm for these models is guaranteed to approximate the optimal influence within a factor of $1 - 1/e \approx 63\%$, that is, $\sigma(S) \geq (1 - 1/e)\sigma(S^*)$, where S is obtained from the greedy algorithm and S^* is the actual optimal set.

In case the cost of removing each node is not identical, the result of this basic greedy algorithm can be far from optimal. In such circumstance, a naive modification of the basic greedy algorithm can be made by favouring the node with maximum benefit–cost ratio. Unfortunately, this intuitive generalization can perform arbitrarily worse than the optimal solution S^*. In order to guarantee a relatively good performance, Leskovec *et al.* [146] proposed the Cost-Effective Forward (CEF) algorithm. As a combination of the benefit–cost and unit-cost greedy algorithms, the CEF algorithm provides a constant factor $(1 - 1/e)/2$ approximation of the maximal influence. Even though each of the two basic greedy algorithms can perform arbitrarily bad, it was proved that for a given circumstance, at least one of them could obtain a relatively good performance.

Due to the heavy computational burden of massive Monte Carlo simulations, greedy algorithms are unscalable to large-scale networks. This can be partly alleviated by exploiting the sparsity of cost reductions [146]. Furthermore, by exploiting the submodular property of the influence function, the number of simulations can be significantly reduced in practice. Given that the marginal increment of a node is monotonically decreasing with the growth of S, there is no need to recompute the marginal increments for all nodes at each time step. Specifically, if the marginal increment of a node i in previous time steps is already smaller than that of another node j in current time step, the recomputation for $\sigma(i)$ is unnecessary as it is definitely smaller than $\sigma(j)$. In calculations, the marginal influence of each node $\sigma(i)$ is marked valid initially. Before the next influencer is selected, the nodes are scanned in a decreasing order of their influence. If $\sigma(i)$ for the top node i is invalid, it is recomputed and inserted into the existing order using a priority queue. If the recomputation leads to a new value that ranks at the top, it should be added into S without calculating the marginal increments for any other nodes. This cost-effective lazy forward (CELF) algorithm leads to far fewer evaluations of the influence function and achieves up to a factor of 700 improvement in speed compared to CEF with equal performance. Further improvement of CELF can be made by recording the node with largest marginal gain among the nodes that are already examined in the current iteration in a heap data structure [147]. This technique can improve the efficiency of CELF by another 35–55%.

Further improvement of greedy algorithms was achieved using the connection between ICMs and percolation. As indicated before, ICMs can be mapped to a bond percolation. Based on this idea, Chen *et al.* [148] performed a bond percolation on a graph G to estimate the influence of a seed set. Specifically, each link in a graph G is randomly selected with the predefined transmission probability, and the selected links form a subgraph G'. Then the influence function $\sigma(S)$ can be quantified by the number of vertices reachable from S in G', where each edge in G' is regarded as a real propagation path. With this simplification, the influence of a single node i can be obtained with a linear scan of the graph G' and its marginal increment to S is either 0 or $\sigma(i)$, depending on whether i is in the influence range of S or not. This procedure provides $O(N)$ speedup to the basic greedy algorithm. In implementation, it can be proceeded in combination with CELF to avoid unnecessary evaluations.

Despite above improvements of greedy algorithms for IC, it is still prohibitive for massively large social networks with millions of users. In order to reach the trade-off between performance and computational efficiency, Chen *et al.* [148] also proposed a heuristic degree discount algorithm. The basic idea of the degree discount algorithm is that $\sigma(i)$ should be quantified by its degree discounted by the number of its neighbours that are already included in S. For ICMs with a small propagation probability, the indirect influence between multi-hop neighbours is negligible so we can only take into account the direct influence between immediate neighbours. Under this assumption, a more precise metric was proposed. The performance of this algorithm nearly matches that of the basic greedy algorithms. Furthermore, it is far more efficient in combined use of the heap data structure and scalable for large-scale networks.

Another scalable variant of the basic greedy algorithm was developed based on local influence regions [149]. The maximum influence arborescence (MIA) algorithm assumes that propagations tend to be along the maximum influence paths (MIP) between each pair of nodes, which are defined as the path with the highest propagation probability among all possible ensembles. For a given pair of nodes, the MIP between them can be computed efficiently using the Dijkstra shortest-path algorithm [150, 151]. The union of MIPs starting or ending at a node i form an arborescence structure, which defines its local influence region denoted by $\delta(i)$. The global influence of a set S is then quantified by the size of the union of all local influence regions: $\sigma(S) = |\bigcup_{i\in S}\delta(i)|$, where $|\cdot|$ denotes the size of a set. A tuning parameter is introduced so that all MIPs with probability below θ are discarded. By adjusting the parameter θ, the size of the local influence regions can be altered so that trade-off between computational efficiency and performance is achieved. Based on such approximations, the local marginal increment of a node can be calculated with significantly high efficiency. As the local influence function is also submodular, the basic greedy algorithm guarantees the $1 - 1/e$ approximation bound for influence maximization. The linearity of local marginal influence allows for the efficient update of incremental influence during iterations. More importantly, the update is only required in a local influence region around the selected influencer.

Wang *et al.* [152] proposed a community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In the algorithm, communities with regional information diffusion are first detected, and influential nodes are then located by selecting certain communities using a dynamic programming algorithm. As shown in recent works, modularity of networks has significant impact on information diffusion [153–155]. In the general idea, the community-based greedy algorithm considers information diffusion within each community to disentangle their interactions, thus simplifies the process of selecting multiple influencers. This algorithm was found to be more than an order of magnitude faster than typical greedy algorithms. In a recent work by Hu *et al.* [156], authors employed percolation theory to show that spreading processes of ICM are limited to a local area in most occasions. Therefore, local structure can identify and quantify influential global spreaders in large-scale social networks. An efficient percolation-based greedy algorithm was proposed.

In another line of research, instead of using Monte Carlo simulations, centrality metrics based on the topological structure of the underlying network were adopted to estimate nodes' influence. These metrics are independent of specific spreading processes thus can be calculated with high computational efficiency. In addition, they also shed light on the impact of network topology on spreading processes, which is of great significance in both accelerating and confining propagations. Instead of actually running the spreading process, these metrics are mostly based on the local or global topology of a node in the network, for instance, number of immediate neighbours [28, 29, 59, 157], global position [17, 158–162], number of shortest paths [163–166], random walks [167–169], eigenvectors [170–173], path counting [174–177], etc. Even though the optimal metric that performs best for all spreading dynamics on all underlying networks does not seem to exist [178–180], these centrality-based approaches are still persistently used due to their simplicity and relative satisfactory performance in some occasions.

4.2 *Message-passing approach*

Although the greedy optimization guarantees to approximate the maximum influence by a constant factor, it often suffers from the drawback of being trapped into local optimum. From an optimization point of view, the message-passing approach, which has been well developed in statistical physics [129, 181], can avoid such undesirable situation. In addition, message-passing algorithms usually scales almost linearly with the number of edges, which makes it applicable to large real-world networks. Based on message-passing approach, Altarelli *et al.* [46] developed the BP and max-sum (MS) algorithms for the problem of optimal immunization for SIR and SIS model.

For each configuration $\mathbf{s} = (s_1, s_2, \ldots, s_N)$, the following energy function is considered

$$\varepsilon(\mathbf{s}, \mathbf{m}) = \mu \sum_{i \in V} s_i c_i + \epsilon \sum_{i \in V} m_i, \tag{16}$$

where $s_i \in \{0, 1\}$ ($s_i = 1$ if i is immunized, and $s_i = 0$ otherwise), c_i is the cost of immunizing node i and m_i is the probability that i is eventually infected in the case of SIR model, or the probability that it is infected in the stationary state in the case of SIS model. The parameters μ and ϵ control the trade-off between the cost of immunization and the cost in treating infected patients. The constraint on all feasible configurations is manifested by the local update equations of m_i. Based on the energy function $\varepsilon(\mathbf{s}, \mathbf{m})$, a Boltzmann weight $e^{-\beta\mathcal{E}}$ is assigned to each feasible configuration, where β is the inverse temperature. Take the SIR model for an example, the probability m_{ij} that node i is infected in the absence of it neighbouring node j satisfies a set of equations:

$$m_{ij} = q + (1 - q) \left[1 - \prod_{k \in \partial i \setminus j} (1 - p m_{ki}) \right], \tag{17}$$

where q is the self-infection probability, p is the transmission probability and $\partial i \setminus j$ denotes the neighbours of node i excluding j. Then the marginal probability m_i that node i is eventually infected is

$$m_i = q + (1-q)\left[1 - \prod_{k\in\partial i}(1 - pm_{ki})\right]. \tag{18}$$

Based on the locally tree-like assumption, BP equations can be derived and solved through iteration making use of the properties of convolutions of messages. As $\beta \to \infty$, the Boltzmann distribution is concentrated on the optimal configuration with the lowest energy cost. In addition, the MS equations can be developed to find the nearly optimal set of immunized nodes. In simulations, MS algorithm performs better than the topological-based heuristics, greedy algorithm as well as SA.

In a recent work by Min [182], the message-passing approach was used to calculate analytically the expected size of epidemic outbreaks originated from a single seed. It was found that, while the probability of triggering an epidemic outbreak depends on the location of the seed, the final size of the outbreak is insensitive to the seed once it occurs. This approach is also applicable to weighted networks.

For ICMs, two important problems are connected: the optimal selection of nodes to either minimize or maximize the influence. The minimization problem, equivalent to optimal percolation, aims to find the 'superblockers' that should be removed to make G_∞ as small as possible. Instead, 'superspreaders' are those that maximize the average influence if selected as seeds. Radicchi and Castellano [183] performed an extensive analysis over a range of real-world networks and found that these two optimization problems are not equivalent, that is, superblockers are not superspreaders. The identification of superblockers is based purely on the topology of the network, while superspreaders in influence maximization problem are strongly dependent on the parameters of the spreading dynamics.

4.3 *Summary*

We summarize features of the methods introduced in this section in Table 2. For greedy approaches, the central task is to estimate the influence of each node, using either Monte Carlo simulation or local structural information. Following this idea, its improvement is designed along two directions: avoiding unnecessary simulations or develop better local proxies for influence. The performance of greedy algorithms is guaranteed for dynamics with submodular property. The message-passing approach calculates the spreading outcomes by solving a set of BP equations, thus considers the problem from a global viewpoint. In addition, there is no requirement for the submodular property.

5. Dynamics with discontinuous transitions

Threshold models and k-core percolation are frequently used to describe cascading processes with discontinuous phase transitions in various disciplines, for instance, failure propagation in infrastructure [94], diffusion of innovations in social networks [184] and adoption of new behaviours [185]. By definition, k-core percolation is a special case of a more general class of threshold models where each node has a fixed threshold k. The fundamental difference from threshold models to ICMs is that, in threshold models, the state of a node is collectively determined by the states of all its neighbours. As a consequence, the impact of perturbing one node can propagate to a vast area of the network through long-range chains of interactions, manifested by a discontinuous phase transition in network dynamics. In this section, we first introduce methods developed for LTMs using greedy strategy, BP and CI, and then discuss

TABLE 2 *Summary of some methods developed for influence maximization in ICMs.*

Name	Description	Ref
Monte Carlo simulations	Greedy approach, submodular function, performance guaranteed within a factor $(1 - 1/e)$	[22]
Cost-Effective Forward algorithm (CEF)	Consider non-unit cost, performance guaranteed within a factor $(1 - 1/e)/2$	[146]
Cost-Effective lazy forward (CELF)	Fewer Monte Carlo simulations, higher efficiency, heap structure	[146]
Percolation-based approach	Map to bond percolation, use subgraph to estimate influence	[148]
Degree discount algorithm	Direct influence between immediate neighbours	[148]
Maximum influence arborescence (MIA)	Maximum influence path, Dijkstra shortest-path algorithm, arborescence structure, trade-off between computational efficiency and performance	[149]
Community-based algorithm	Community detection, dynamic programming algorithm	[152]
Message-passing approach	Belief propagation, Max-Sum algorithm, SIR and SIS model, solved through iteration	[46]
Message-passing approach	Expected size of epidemic outbreaks, insensitive to origin, applicable to weighted networks	[182]

algorithms designed for k-core percolation. Note that algorithms designed for LTMs are applicable to k-core percolation.

5.1 *Linear threshold models*

LTMs have several different forms. A typical LTM is defined on a weighted network $G = (V, E, \omega)$, where $\omega : V \times V \to [0, 1]$ is a weight function and $\omega = 0$ iff the corresponding edge does not exist. Similar to ICMs, the spreading process in LTMs is initiated by a set of seeds while all other nodes are inactive. In following steps, a node is activated if the sum of weights of its active neighbours reaches its predefined threshold value θ_i, that is, $\sum_{j\in\partial i} \omega_{ij} \geq \theta_i$, where ∂i stands for the set of neighbours of node i. In another form, a node is activated if it has at least a certain number of active neighbours. In this subsection, we discuss algorithms based on greedy approach, BP and CI.

5.1.1 *Greedy approach* The greedy algorithm is also applicable to LTMs. For a special class of LTMs where the weight of each edge and the threshold of each node are drawn uniformly from the interval [0, 1], it was proved that its influence function is submodular [22]. Therefore, the influence maximization problem in this class of LTMs can be approximately solved by greedy algorithms.

Like ICMs, a LTM can be also mapped to a modified percolation process defined as follows: each node i picks at most one of its incoming edges, with probability ω_{ji} to select the edge from j to i and $1 - \sum_j \omega_{ji}$ to select none. The selected edges are defined as live. Considering the subgraph G' composed of live edges, Kempe *et al.* [22] proved that for a given set S, the number of nodes activated by S in LTMs has the same distribution with the number of reachable nodes of S in the subgraph G'.

Using the same mapping, Chen *et al.* [149] gave an efficient approximation of the influence of an individual node in a local subgraph. In cases where the weights ω_{ij} and ω_{ji} are not symmetrical, the undirected graph G can be transformed into an equivalent directed graph, where edges from i to j and from j to i are both included. Using the randomized algorithm of Cohen [186], the influence of a set S is quantified by the number of nodes reachable from S in the subgraph G'. Although computing the exact influence in a network is #P-hard, this approximation based on directed acyclic graph (DAG) can be finished within linear time. In order to further accelerate the calculation, a local DAG (LDAG) is considered instead of DAG. Validation of this approximation is supported by the exponential decay of influence with the propagation length. The construction of LDAG should include a majority part of influence from other nodes while discarding the nodes with small influence. Similar to the idea in Ref. [149], a threshold is introduced to control the size of LDAG, so that the trade-off between efficiency and accuracy can be tuned. Once the LDAG is constructed, the incremental influence of each node can be quantified with great efficiency. As a result, the LDAG algorithm is scalable to networks with millions of nodes and is among the best greedy algorithms in performance.

The LDAG algorithm assumes that the influence of a node is mainly bounded within its LDAG. However, if the spreading process starting from a node can reach outside its LDAG, the estimation of influence in the LDAG algorithm might be inaccurate. Besides, the algorithm depends heavily on the proper choice of a high quality LDAG, which is an NP-hard problem itself. To avoid these problems, Goyal *et al.* [147] developed the SIMPATH algorithm in which the influence of a node is quantified by enumerating the simple paths starting from it. Although this problem is also #P-hard, it can be well approximated with high efficiency by enumerating paths within a small neighbourhood. With this approximation, the influence of a set S can be calculated as the sum of influence of each node in it on appropriately induced subgraphs. Similar to the arborescence structures constructed in Ref. [149], a tuning parameter is introduced to control the size of the neighbourhood, which leads to a direct trade-off between the accuracy and computational efficiency. To reduce the number of estimation calls in SIMPATH, a vertex cover optimization was introduced so that only the influence of nodes in the vertex cover set needs to be computed. For the rest of the nodes, their influence can be derived from their neighbours. Besides, as the seed set S grows larger, a look ahead optimization can be made to accelerate the estimation: It picks the top l most promising candidates as a batch in the start of an iteration and shares the marginal gain computation within the batch. Extensive experiments on real datasets show that compared with the basic greedy algorithm, the SIMPATH algorithm is more efficient, consumes less memory and produces seed sets with larger influence.

The performance of most greedy algorithms mentioned above is guaranteed thanks to the submodular property of the influence function. However, for a general LTM with fixed weights and thresholds, the influence function is not always submodular [22]. An important class of LTM that may not be submodular is defined as follows: a node i is activated only after a certain number m_i of its neighbours are activated. The variation of threshold m_i can lead to two qualitatively different classes of cascades featured by either continuous or discontinuous phase transitions. For instance, in the special case when $m_i = k_i - 1$ (k_i is the degree of node i), the scale of propagation experiences a continuous phase transition [20]. In contrast, for k-core percolation and bootstrap percolation, a first-order, or discontinuous phase transition may appear [44]. Solutions to the influence maximization problem in LTMs without submodular property require a better understanding of the physical mechanism of the spreading process, and will be introduced in detail in following subsections.

5.1.2 *BP algorithms* For the influence maximization problem on a general LTM, Altarelli *et al.* [60] regarded it as a non-typical trajectory deviated from the average behaviour of dynamics initiated by randomly chosen seeds. To explore the dynamical properties of non-typical trajectories of general LTMs, Altarelli *et al.* [47] proposed a BP algorithm that could estimate statistical properties of non-typical trajectories and found the initial conditions that lead to cascading with desired properties. In contrast to ICMs, the trajectory of a given LTM is determined solely by its initial condition. Due to the irreversibility of LTM dynamics, the spreading process can be parameterized by a configuration $\mathbf{t} = (t_1, t_2, ...t_N)$, where $t_i \in \mathbf{T} = \{0, 1, ...T, \infty\}$ is the activation time of node i. Considering the properties of LTM, the dynamical rule can be represented by the constraint on the activation time of a node and its neighbours [47]: $t_i = \phi_i(\{t_j\})$, where

$$t_i = \phi_i(\{t_j\}) = \min\left\{t \in \mathbf{T} : \sum_{j\in\partial i} \omega_{ji}\mathbb{I}[t_j < t] \geq \theta_i\right\}. \tag{19}$$

Based on this static parametrization of LTM, the following Boltzmann distribution is considered:

$$P(\mathbf{t}) = \frac{1}{Z}e^{-\beta\varepsilon(\mathbf{t})}\prod_{i\in V}\psi_i(t_i, \{t_j\}), \tag{20}$$

where $\psi_i(t_i, \{t_j\}) = \mathbb{I}[t_i = 0] + \mathbb{I}[t_i = \phi_i(\{t_j\})]$, $Z = \sum_{\mathbf{t}} e^{-\beta\varepsilon(\mathbf{t})}\prod_{i\in V}\psi_i(t_i, \{t_j\})$. The most common form of the energy function is $\varepsilon(\mathbf{t}) = \sum_i \varepsilon_i(t_i)$, where $\varepsilon_i(t_i) = \mathbb{I}[t_i = 0] - \epsilon\mathbb{I}[t_i < \infty]$. For $\epsilon = 0$, the distribution degenerates to the spreading dynamics initiated by a random set of seeds.

In order to avoid short loops in the factor graph that describes the constraints of a configuration, a dual factor graph is constructed with a variable node (t_i, t_j) introduced to each edge (i, j). The obtained dual factor graph is locally tree-like if the original network is so. This allows for the application of the cavity method. Denote $P_j(t_j)$ as the marginal probability that node j is activated at time t_j. In a tree-like factor graph, it can be calculated as

$$P_j(t_j) \propto \sum_{\{t_i\}_{i\in\partial j}} e^{-\beta\varepsilon_j(t_j)}\psi_j(t_j, \{t_i\})\prod_{i\in\partial j}H_{ij}(t_i, t_j), \tag{21}$$

where $H_{ij}(t_i, t_j)$ is defined as the probability that nodes i and j get activated at t_i and t_j, respectively in the absence of the constraint ψ_j and the energy term ε_j. This equation computes the contribution from all neighbours of node j. In the dual factor graph, $H_{ij}(t_i, t_j)$, named cavity marginals or 'beliefs', satisfy local constraints described by a set of BP equations. In particular, the recursive relation of the cavity marginal $H_{ij}(t_i, t_j)$ on the dual factor graph defines the following belief BP equations [47]:

$$H_{ij}(t_i, t_j) \propto e^{-\beta\varepsilon_i(t_i)}\sum_{\{t_k\}}\psi_i(t_i, \{t_k\})\prod_k H_{ki}(t_k, t_i). \tag{22}$$

Here $\psi_i(t_i, \{t_k\})$ is the local constraint on links connected to node i (except node j), the product term computes the contribution of 'beliefs' from the neighbours of node i excluding node j, the summation term considers different occasions of t_k for neighbours of node i and $e^{-\beta\varepsilon_i(t_i)}$ defines the weight for energy $\varepsilon_i(t_i)$ using the Boltzmann distribution. The BP equations are solved through iteration. Once the fixed values of the cavity marginals are obtained, the marginal $P_j(t_j)$ and other statistics of non-typical trajectories, such as the entropy and distribution of activation time, can be subsequently computed.

On homogeneous random regular graphs, the BP equations can be simplified to a self-consistent equation of a single marginal. Analysis for different threshold values indicates quantitative difference

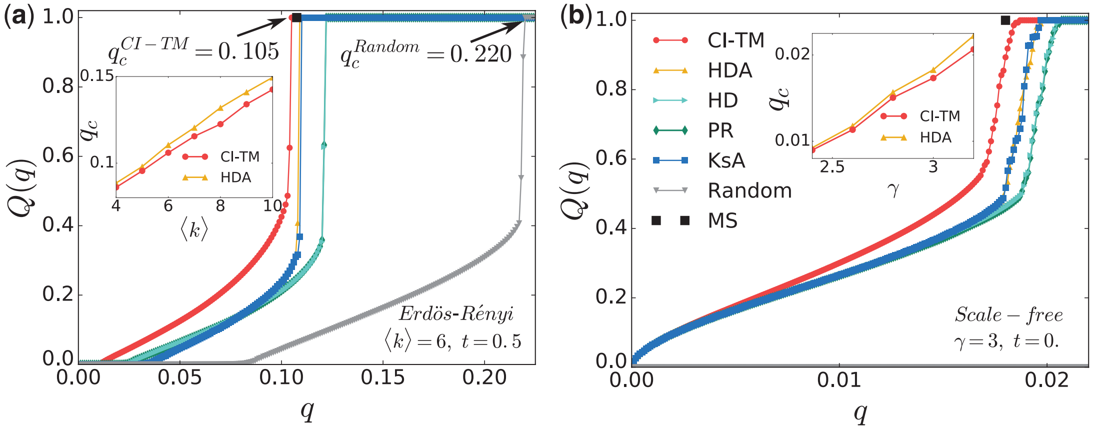

FIG. 8. Performance of the CI-TM algorithm in ER and SF networks. (a) Size of active GC $Q(q)$ as a function of the fraction of seed q for ER networks ($N = 2 \times 10^5$, $\langle k \rangle = 6$). The CI-TM algorithm is compared with high degree adaptive (HDA), high degree (HD), PageRank (PR), k-core adaptive (KsA), Random and the Max-Sum (MS) algorithm. Inset shows the critical values q_c identified by HDA and CI-TM for different mean degrees. (b) Results for scale-free networks ($N = 2 \times 10^5$, $\gamma = 3$). Inset presents the critical values q_c for different power-law exponents γ. Figure reuse from [188] is permitted by Springer Nature.

in the distribution of activation time $P(t)$ for the regimes of continuous and discontinuous transitions. Specifically, for continuous transitions, $P(t)$ is monotonically decreasing. On the contrary, $P(t)$ shows a second peak for discontinuous transitions, corresponding to the abrupt cascade activation. In order to obtain the optimal set of seeds, Max-Sum equations can be derived by setting the inverse temperature $\beta \to \infty$ in the energy function [47]. Authors performed numerical experiments on a real-world network (the Epinions network) with an energy function $\varepsilon(\mathbf{t}) = \sum_i \{c_i \mathbb{I}[t_i = 0] - r_i \mathbb{I}[t_i < \infty]\}$, where c_i is the cost of seeding node i and r_i is the revenue generated by the activation of node r. The Max-Sum algorithm was compared with competing methods including greedy algorithm based on energy computation (GA), greedy algorithm based on HITS (HITS), high degree (Hubs) and SA. The Max-Sum algorithm outperforms other approaches by selecting the seed set that best trade-offs the revenue and cost. The performance of Max-Sum algorithms on synthetic networks also outperforms a range of centrality metrics, as shown in Fig. 8.

Extending the work under the assumption of replica symmetry, Guggiola and Semerjian [100] studied the minimal contagious set problem for LTM dynamics with and without a constraint on the maximal activation time T. In this theoretically impressive work, authors aim to find the theoretical limit of the minimal contagious set (i.e. the minimal seed set that can activate the entire graph) in random regular graphs using the cavity method with the effect of replica symmetry breaking. Following the theoretical development, a survey propagation like algorithm [187] is investigated on single instances of random regular graphs to find the exact seed set. It was found that the survey propagation algorithm achieves near-optimal performance for small activation time limit. For a large activation time limit, authors reported convergence issues in iteration that cannot be effectively solved by a simple damping. However, stopping the iterations after a predefined time proved to be a pragmatic and satisfactory strategy. In this work, authors tested the algorithm on random regular graphs; in practice, how survey propagation algorithm works for more realistic networks needs to be tested. Readers interested in the survey propagation algorithm can find details in Ref. [100].

5.1.3 *CI in threshold model* The CI theory can be generalized to deal with the influence maximization in general LTMs [188]. For a network $G(V,E)$ with N nodes and M links, we use the vector $\mathbf{n} = (n_1, n_2, \ldots, n_N)$ to record whether a node i is chosen as a seed ($n_i = 1$) or not ($n_i = 0$). The LTM spreading starts from a $q = \sum_i n_i/N$ fraction of active seeds and evolves following a threshold rule: a node i becomes active if it has at least m_i active neighbours. Here, the threshold m_i is an integer ranging from 1 to the degree of node i. Further, we introduce ν_i to indicate the final state of node i: active ($\nu_i = 1$) or inactive ($\nu_i = 0$). For a given q fraction of seeds, the influence maximization problem aims to find the optimal set of seeds so that the size of active population is maximized.

For each link $i \to j$, we introduce a binary variable $\nu_{i\to j}$ as the indicator of i being in the active state assuming node j is disconnected from the network. For locally tree-like networks, $\nu_{i\to j}$ satisfies a set of self-consistent message-passing equations. Different from the case of optimal percolation in Ref. [20], the zero solution is not a fixed point. As a consequence, the stability analysis around zero solution in Ref. [20] is no longer valid for LTMs. However, the solution can be approximated through iteration of the linearized system. By linearizing the equations, it was found that the subsequent activation of nodes in each iteration only depends on the number of subcritical nodes, defined as the nodes with $m_i - 1$ active neighbours (i.e. nodes whose activation can be triggered by one more active neighbour). Moreover, subcritical nodes can form long subcritical paths that generate long-range cascade of activation, which is core to the discontinuous transition in LTM dynamics. Following this idea, the CI-TM (Collective Influence in Threshold Model) algorithm was proposed that recursively selects nodes with the largest CI-TM score. The CI-TM score enumerates the number of subcritical paths starting from each node, and uses that to quantify nodes' spreading capability. With an $O(N \log N)$ computational complexity, the efficient CI-TM algorithm is applicable to large-scale networks. In numerical simulations, the CI-TM algorithm outperforms the greedy algorithm and several widely used heuristic centralities, and achieves comparable performance to the Max-Sum algorithm in synthetic random networks (see Fig. 8).

5.2 *The k-core percolation*

Because k-core percolation is a special case of LTMs, influence maximization algorithms developed for general LTMs can be naturally extended to work for k-core percolation.

In statistical physics and combinatorial optimization, several theoretical works have explored the lower and upper bounds on the size of the minimal set to destroy the k-core. In the evaluation of approximating algorithms, these results can help us to assess how far the estimated size of minimal contagious set is from the theoretical limit. For instance, Bau *et al.* [110] studied the decycling numbers of random regular graphs. As stated before, the decycling process is equivalent to destroying the 2-core of networks. For a random cubic graph G that all nodes have degree 3, it was proven that the decycling number $\phi(G) = \lceil N/4 + 1/2 \rceil$ as the graph size $N \to \infty$. For a general random d-regular graph G with N nodes ($d \geq 4$), authors proved that $\phi(G)/N$ is bounded below and above asymptotically almost surely by certain constants that depend solely on d. In particular, the lower and upper bounds can be calculated by solving an algebraic equation and a set of differential equations, respectively. Janson and Thomason [189] found that, for sparse random graphs or random regular networks with N nodes with $N \to \infty$, the number of nodes that must be removed so that no component with more than k nodes exists is essentially the same for all values of k if $k \to \infty$ and $k = o(N)$. Reichman [103] showed that the size of a contagious set is bounded from above by $\sum_{v\in V} \min\left\{1, \frac{k}{d(v)+1}\right\}$ in the destruction of k-core ($d(v)$ is the degree of node v). Later, using the cavity method with replica symmetry breaking, Guggiola and Semerjian [100] obtained several conjectures on the size of minimal contagious sets for k-core percolation in random regular graphs. In particular, authors conjectured that the minimal contagious set size is 1/6 for 5-regular random graphs with a threshold of 3, and 1/4 for 6-regular with threshold 4. In addition, they also proposed the conjecture for (k+1)-regular networks with the threshold k that the minimal contagious set size is $1 - 2(\ln k)/k - 2/k + O(1/(k \ln k))$. According to this conjecture, the minimal contagious set size 3-regular (cubic) random graphs with a threshold of 2 is 1/4, which is in agreement with the decycling number of cubic random graphs $\phi(G)/N \to 1/4$ obtained in Ref. [110]. Sun *et al.* [190] also proposed a lower bound of the network dismantling problem by analysing specific 2-core subnetworks of many real-world networks that have heterogeneous degree distribution. Coja-Oghlan *et al.* [191] explored the minimal contagious set problem on graphs with expansion properties.

Recently, Schmidt *et al.* [133] studied the minimal contagious sets for k-core percolation in random networks. In this work, authors proposed a generalized CoreHD algorithm, in which nodes with the highest degree in the k-core are recursively removed until the k-core completely collapses. To analyse the property of this algorithm, the generalized CoreHD-guided k-core removal was translated to a random process on the degree distribution of the graph [192, 193]. The running time of the process, characterized by a set of non-linear ordinary differential equations, describes the behaviour of the algorithm on a random graph. By analysing the stopping time, new upper bounds on the minimal contagious set were obtained, which improve the best currently known ones in Ref. [100, 110]. This approach is applicable not only to random regular graphs, but also to random networks generated from the configuration model with a given degree distribution. Inspired by the analysis of the CoreHD algorithm, an improved algorithm, called WEAK-NEIGHBOR, was developed. In this algorithm, instead of removing high-degree nodes, nodes with the highest value $k_i - \sum_{j \in \partial i} k_j/k_i$ in the k-core are removed (k_i is the degree of node i). For networks with bounded degree, the algorithm has $O(N)$ complexity, where N is the network size. In numerical experiments, the WEAK-NEIGHBOR algorithm improves over the generalized CoreHD algorithm and CI-TM algorithm in a range of k-core percolation processes in random regular graphs.

5.3 *Summary*

For LTMs, the major effort in greedy approach is to develop more efficient and accurate estimation of marginal increments using local network structure. This pursuit has inspired different techniques designed for this goal. Most greedy methods quantify the marginal increment by the number of nodes that would be activated if a node is selected as a seed. The CI-TM algorithm, in contrast, uses the number of subcritical paths attached to a node to estimate the marginal increment. BP approaches solve the problem as a global issue through iteration, and can flexibly incorporate the cost of activating seeds. Apart from devising practical methods to solve the influence maximization problem for LTMs, analytical works on random regular graphs would help to identify how far away current approaches are from the theoretical limit of the size of optimal seed set. Features about the introduced methods are summarized in Table 3.

6. Conclusions and discussions

With an increasing number of real-world complex systems formulated as networks, a theory for identifying influencers is required to facilitate a better understanding and control of various dynamical complex systems. Over the years, this problem has been extensively studied in different contexts by physicists, mathematicians, sociologists, computer scientists, etc. In this survey, we review recent advances in this area. Because this topic spans a wide spectrum of research, we cannot report every relevant work exhaustively. However, we try to organize the survey in a way such that recent developments made in several fields of broad interest are covered.

TABLE 3 *Summary of some methods developed for influence maximization in LTMs and k-core percolation.*

Name	Description	References
Monte Carlo simulations	Greedy approach, submodular function, performance guaranteed within a factor $(1 - 1/e)$	[22]
Percolation-based approach	Map to bond percolation, use subgraph to estimate influence, $O(N)$ complexity	[149]
Local directed acyclic graph (LDAG)	Decay of influence with the propagation length, discard the nodes with small influence	[149]
SIMPATH algorithm	Enumerating simple paths, look ahead optimization, trade-off between accuracy and efficiency	[147]
Belief-propagation algorithm	Large deviations of LTM dynamics, consider cost and revenue, Max-Sum equations, solved by iteration until convergence	[47]
Survey propagation like algorithm	Near-optimal performance, solved by iteration until convergence, applied to random regular graphs	[100]
Collective Influence in Threshold Model (CI-TM)	Greedy approach, based on linearized message-passing equations, use subcritical clusters to estimate influence, $O(N \log N)$ complexity	[188]
CoreHD	Remove high-degree nodes in 2-core, perform well on loopy networks, $O(N)$ complexity	[132]
WEAK-NEIGHBOR	Improvement over the generalized CoreHD algorithm and CI-TM algorithm, $O(N)$ complexity	[133]

Despite great advances in influencer identification, many ongoing problems and directions exist that need to be addressed in future works. First, as shown in several theoretical works, even for homogeneous structure such as random regular networks, there is still a gap between the result obtained from the state-of-the-art algorithms and its theoretical limit. This provides a room that we can improve in algorithm design. Second, the topological structure of real-world complex systems can be much more complicated than the case considered in ideal conditions. In a recent comparative analysis, it was found that recently proposed techniques perform well only on specific network types [194]. Further, connections may be time-varying in temporal networks [195], or possess complicated interlayer interactions in multiplex networks [196, 197]. Third, in many systems, links are often of different types with distinct functions. These systems cannot be described by the simple network structure discussed before, and do not even admit a formal definition of influencers. In future works, these open problems remain to be explored in more detail.

In terms of applications, use of influencer identification theory in biological, social and engineering systems is still very limited. As some advanced methodologies in statistical physics are technical and challenging to interpret, applying the latest progresses of influencer identification in specific real-world systems can better illustrate and disseminate these techniques. Moreover, current methods are mostly developed under ideal conditions. In real-world systems, errors or noises inevitably exist [198, 199]. How to quantify and alleviate the impact of errors or noises is of great practical values in applications.

In addition, certain non-dynamical factors beyond the simplified assumption in pure modelling studies, for example, human activity [200–208], homophily [209–211], complex contagion [185, 212] and social influence bias [213], may need to be considered. This calls for a deeper understanding of the systems under study and a more integrative application of the influencer identification theory.

Funding

The National Institutes of Health (R01EB022720, U54CA137788 and U54CA132378 to H.A.M.); National Science Foundation (1515022 to H.A.M.); Army Research Laboratory (W911NF-09-2-0053 to H.A.M.); and China Scholarship Council and the Academic Excellence Foundation of BUAA for PhD Students to J.W.

References

1. Pastor-Satorras, R., Castellano, C., Van Mieghem, P. & Vespignani, A. (2015) Epidemic processes in complex networks. *Rev. Mod. Phys.*, **87**, 925.

2. Zhang, Z.-K., Liu, C., Zhan, X.-X., Lu, X., Zhang, C.-X. & Zhang, Y.-C. (2016) Dynamics of information diffusion and its applications on complex networks. *Phys. Rep.*, **651**, 1–34.

3. Bullmore, E. & Sporns, O. (2009) Complex brain networks: graph theoretical analysis of structural and functional systems. *Nat. Rev. Neurosci.*, **10**, 186.

4. Montoya, J. M., Pimm, S. L. & Solé, R. V. (2006) Ecological networks and their fragility. *Nature*, **442**, 259.

5. Newman, M. E. J. (2003) The structure and function of complex networks. *SIAM Rev.*, **45**, 167–256.

6. Barrat, A., Barthelemy, M. & Vespignani, A. (2008) *Dynamical Processes on Complex Networks*. New York: Cambridge University Press.

7. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. & Hwang, D.-U. (2006) Complex networks: structure and dynamics. *Phys. Rep.*, **424**, 175–308.

8. Albert, R. & Barabási, A.-L. (2002) Statistical mechanics of complex networks. *Rev. Mod. Phys.*, **74**, 47.

9. Watts, D. J. & Dodds, P. S. (2007) Influentials, networks, and public opinion formation. *J. Consumer Res.*, **34**, 441–458.

10. Del Ferraro, G., Moreno, A., Min, B., Morone, F., Pérez-Ramírez, Ú., Pérez-Cervera, L., Parra, L. C., Holodny, A., Canals, S. & Makse, H. A. (2018) Finding influential nodes for integration in brain networks using optimal percolation theory. *Nat. Commun.*, **9**, 2274.

11. Reis, S. D., Hu, Y., Babino, A., Andrade Jr, J. S., Canals, S., Sigman, M. & Makse, H. A. (2014) Avoiding catastrophic failure in correlated networks of networks. *Nat. Phys.*, **10**, 762.

12. Zamora-López, G., Zhou, C. & Kurths, J. (2010) Cortical hubs form a module for multisensory integration on top of the hierarchy of cortical networks. *Front. Neuroinform.*, **4**, 1.

13. May, R. M. (1972) Will a large complex system be stable? *Nature*, **238**, 413.

14. Scheffer, M., Carpenter, S. R., Lenton, T. M., Bascompte, J., Brock, W., Dakos, V., Van de Koppel, J., Van de Leemput, I. A., Levin, S. A., Van Nes, E. H. & Pascual, M. (2012) Anticipating critical transitions. *Science*, **338**, 344–348.

15. Mills, L. S., Soulé, M. E. & Doak, D. F. (1993) The keystone-species concept in ecology and conservation. *BioScience*, **43**, 219–224.

16. Morone, F., Del Ferraro, G. & Makse, H. A. (2019) The k-core as a predictor of structural collapse in mutualistic ecosystems. *Nat. Phys.*, **15**, 95.

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

18. Pei, S., Morone, F., Liljeros, F., Makse, H. A. & Shaman, J. (2018) Inference and control of the nosocomial transmission of Methicillin-resistant *Staphylococcus aureus*. *eLife*, **7**, e40977.

19. Freeman, L. C. (1978) Centrality in social networks conceptual clarification. *Soc. Netw.*, **1**, 215–239.

20. Morone, F. & Makse, H. A. (2015) Influence maximization in complex networks through optimal percolation. *Nature*, **524**, 65.

21. Braunstein, A., Dall'Asta, L., Semerjian, G. & Zdeborová, L. (2016) Network dismantling. *Proc. Natl. Acad. Sci. USA*, **113**, 12368–12373.

22. Kempe, D., Kleinberg, J. & Tardos, É. (2003) Maximizing the spread of influence through a social network. *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. *Washington, DC, USA*. New York, NY, USA: ACM, pp. 137–146.

23. Leskovec, J., Adamic, L. A. & Huberman, B. A. (2007) The dynamics of viral marketing. *ACM Trans. Web (TWEB)*, **1**, 5.

24. Richardson, M. & Domingos, P. (2002) Mining knowledge-sharing sites for viral marketing. *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. *Washington, DC, USA*. New York, NY, USA: ACM, pp. 61–70.

25. Pastor-Satorras, R. & Vespignani, A. (2002) Immunization of complex networks. *Phys. Rev. E*, **65**, 036104.

26. Chen, Y., Paul, G., Havlin, S., Liljeros, F. & Stanley, H. E. (2008) Finding a better immunization strategy. *Phys. Rev. Lett.*, **101**, 058701.

27. Cohen, R., Havlin, S. & Ben-Avraham, D. (2003) Efficient immunization strategies for computer networks and populations. *Phys. Rev. Lett.*, **91**, 247901.

28. Albert, R., Jeong, H. & Barabási, A.-L. (2000) Error and attack tolerance of complex networks. *Nature*, **406**, 378.

29. Cohen, R., Erez, K., Ben-Avraham, D. & Havlin, S. (2001) Breakdown of the Internet under intentional attack. *Phys. Rev. Lett.*, **86**, 3682.

30. Latora, V. & Marchiori, M. (2005) Vulnerability and protection of infrastructure networks. *Phys. Rev. E*, **71**, 015103.

31. Keeling, M. J. & Rohani, P. (2011) *Modeling Infectious Diseases in Humans and Animals*. Princeton: Princeton University Press.

32. Thébault, E. & Fontaine, C. (2010) Stability of ecological communities and the architecture of mutualistic and trophic networks. *Science*, **329**, 853–856.

33. Kandel, E. R., Schwartz, J. H., Jessell, T. M., of Biochemistry, D., Jessell, M. B. T., Siegelbaum, S. & Hudspeth, A. (2000) *Principles of Neural Science*, vol. 4. New York: McGraw-Hill.

34. Davidson, E. & Levin, M. (2005) Gene regulatory networks. *Proc. Natl. Acad. Sci. USA*, **102**, 4935–4935.

35. Erdős, P. & Rényi, A. (1960) On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, **5**, 17–60.

36. Callaway, D. S., Newman, M. E. J., Strogatz, S. H. & Watts, D. J. (2000) Network robustness and fragility: percolation on random graphs. *Phys. Rev. Lett.*, **85**, 5468.

37. Newman, M. E. J. (2002) Spread of epidemic disease on networks. *Phys. Rev. E*, **66**, 016128.

38. Seidman, S. B. (1983) Network structure and minimum degree. *Soc. Netw.*, **5**, 269–287.

39. Dorogovtsev, S. N., Goltsev, A. V. & Mendes, J. F. F. (2006) k-Core organization of complex networks. *Phys. Rev. Lett.*, **96**, 040601.

40. Kermack, W. O. & McKendrick, A. G. (1932) Contributions to the mathematical theory of epidemics. II.—The problem of endemicity. *Proc. R. Soc. Lond. A*, **138**, 55–83.

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

42. Baxter, G. J., Dorogovtsev, S. N., Goltsev, A. V. & Mendes, J. F. (2010) Bootstrap percolation on complex networks. *Phys. Rev. E*, **82**, 011103.

43. Granovetter, M. (1978) Threshold models of collective behavior. *Am. J. Sociol.*, **83**, 1420–1443.

44. Goltsev, A. V., Dorogovtsev, S. N. & Mendes, J. F. F. (2006) k-core (bootstrap) percolation on complex networks: critical phenomena and nonlocal effects. *Phys. Rev. E*, **73**, 056101.

45. Watts, D. J. (2002) A simple model of global cascades on random networks. *Proc. Natl. Acad. Sci. USA*, **99**, 5766–5771.

46. Altarelli, F., Braunstein, A., Dall'Asta, L., Wakeling, J. R. & Zecchina, R. (2014) Containing epidemic outbreaks by message-passing techniques. *Phys. Rev. X*, **4**, 021024.

47. Altarelli, F., Braunstein, A., Dall'Asta, L. & Zecchina, R. (2013) Optimizing spread dynamics on graphs by message passing. *J. Stat. Mech.*, **2013**, P09011.

48. Mugisha, S. & Zhou, H.-J. (2016) Identifying optimal targets of network attack by belief propagation. *Phys. Rev. E*, **94**, 012305.

49. Clusella, P., Grassberger, P., Pérez-Reche, F. J. & Politi, A. (2016) Immunization and targeted destruction of networks using explosive percolation. *Phys. Rev. Lett.*, **117**, 208301.

50. Pei, S. & Makse, H. A. (2013) Spreading dynamics in complex networks. *J. Stat. Mech.*, **2013**, P12002.

51. Lü, L., Chen, D., Ren, X.-L., Zhang, Q.-M., Zhang, Y.-C. & Zhou, T. (2016) Vital nodes identification in complex networks. *Phys. Rep.*, **650**, 1–63.

52. Pei, S., Morone, F. & Makse, H. A. (2018) Theories for influencer identification in complex networks. *Complex Spreading Phenomena in Social Systems* (S. Lehmann & Y.-Y. Ahn eds). Cham, Switzerland: Springer, pp. 125–148.

53. Bollobás, B. (1998) Random graphs. *Modern Graph Theory*. New York: Springer, pp. 215–252.

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

55. Barabási, A.-L. & Albert, R. (1999) Emergence of scaling in random networks. *Science*, **286**, 509–512.

56. Clauset, A., Shalizi, C. R. & Newman, M. E. J. (2009) Power-law distributions in empirical data. *SIAM Rev.*, **51**, 661–703.

57. Radicchi, F. (2015) Predicting percolation thresholds in networks. *Phys. Rev. E*, **91**, 010801.

58. Karrer, B., Newman, M. E. J. & Zdeborová, L. (2014) Percolation on sparse networks. *Phys. Rev. Lett.*, **113**, 208702.

59. Pastor-Satorras, R. & Vespignani, A. (2001) Epidemic spreading in scale-free networks. *Phys. Rev. Lett.*, **86**, 3200.

60. Altarelli, F., Braunstein, A., Dall'Asta, L. & Zecchina, R. (2013) Large deviations of cascade processes on graphs. *Phys. Rev. E*, **87**, 062115.

61. Hartmann, A. K. (2011) Large-deviation properties of largest component for random graphs. *Eur. Phys. J. B*, **84**, 627–634.

62. Bianconi, G. (2018) Rare events and discontinuous percolation transitions. *Phys. Rev. E*, **97**, 022314.

63. Bianconi, G. (2019) Large deviation theory of percolation on multiplex networks. *J. Stat. Mech.*, **2019**, 023405.

64. Coghi, F., Radicchi, F. & Bianconi, G. (2018) Controlling the uncertain response of real multiplex networks to random damage. *Phys. Rev. E*, **98**, 062317.

65. Hethcote, H. W. (2000) The mathematics of infectious diseases. *SIAM Rev.*, **42**, 599–653.

66. Anderson, R. M. & May, R. M. (1992) *Infectious Diseases of Humans: Dynamics and Control*. Oxford: Oxford University Press.

67. Diekmann, O. & Heesterbeek, J. A. P. (2000) *Mathematical Epidemiology of Infectious Diseases: Model Building, Analysis and Interpretation*, vol. 5. West Sussex, England: John Wiley & Sons.

68. Pastor-Satorras, R. & Vespignani, A. (2001) Epidemic dynamics and endemic states in complex networks. *Phys. Rev. E*, **63**, 066117.

69. Moreno, Y., Nekovee, M. & Pacheco, A. F. (2004) Dynamics of rumor spreading in complex networks. *Phys. Rev. E*, **69**, 066130.

70. Li, W., Tang, S., Pei, S., Yan, S., Jiang, S., Teng, X. & Zheng, Z. (2014) The rumor diffusion process with emerging independent spreaders in complex networks. *Physica A*, **397**, 121–128.

71. Yan, S., Tang, S., Pei, S., Jiang, S. & Zheng, Z. (2014) Dynamical immunization strategy for seasonal epidemics. *Phys. Rev. E*, **90**, 022808.

72. Altarelli, F., Braunstein, A., Dall'Asta, L., Lage-Castellanos, A. & Zecchina, R. (2014) Bayesian inference of epidemics on networks via belief propagation. *Phys. Rev. Lett.*, **112**, 118701.

73. Lokhov, A. Y., Mézard, M., Ohta, H. & Zdeborová, L. (2014) Inferring the origin of an epidemic with a dynamic message-passing algorithm. *Phys. Rev. E*, **90**, 012801.

74. Shah, D. & Zaman, T. (2010) Detecting sources of computer viruses in networks: theory and experiment. *ACM SIGMETRICS Performance Evaluation Review*. New York, NY, USA: ACM, pp. 203–214.

75. Pei, S., Tang, S. & Zheng, Z. (2015) Detecting the influence of spreading in social networks with excitable sensor networks. *PLoS One*, **10**, e0124848.

76. Shah, D. & Zaman, T. (2011) Rumors in a network: who's the culprit? *IEEE Trans. Inf. Theory*, **57**, 5163–5181.

77. Comin, C. H. & da Fontoura Costa, L. (2011) Identifying the starting point of a spreading process in complex networks. *Phys. Rev. E*, **84**, 056105.

78. Pei, S., Kandula, S., Yang, W. & Shaman, J. (2018) Forecasting the spatial transmission of influenza in the United States. *Proc. Natl. Acad. Sci. USA*, **115**, 2752–2757.

79. Scarpino, S. V. & Petri, G. (2019) On the predictability of infectious disease outbreaks. *Nat. Commun.*, **10**, 898.

80. Pei, S. & Shaman, J. (2017) Counteracting structural errors in ensemble forecast of influenza outbreaks. *Nat. Commun.*, **8**, 925.

81. Kandula, S., Yamana, T., Pei, S., Yang, W., Morita, H. & Shaman, J. (2018) Evaluation of mechanistic and statistical methods in forecasting influenza-like illness. *J. R. Soc. Interface*, **15**, 20180174.

82. Pei, S., Cane, M. A. & Shaman, J. (2019) Predictability in process-based ensemble forecast of influenza. *PLoS Comput. Biol.*, **15**, e1006783.

83. Batagelj, V. & Zaveršnik, M. (2011) Fast algorithms for determining (generalized) core groups in social networks. *Adv. Data Anal. Class.*, **5**, 129–145.

84. Lü, L., Zhou, T., Zhang, Q.-M. & Stanley, H. E. (2016) The H-index of a network node and its relation to degree and coreness. *Nat. Commun.*, **7**, 10168.

85. Hirsch, J. E. (2005) An index to quantify an individual's scientific research output. *Proc. Natl. Acad. Sci. USA*, **102**, 16569–16572.

86. Azimi-Tafreshi, N., Gómez-Gardenes, J. & Dorogovtsev, S. (2014) k- core percolation on multiplex networks. *Phys. Rev. E*, **90**, 032816.

87. Chalupa, J., Leath, P. L. & Reich, G. R. (1979) Bootstrap percolation on a Bethe lattice. *J. Phys. C*, **12**, L31.

88. Cellai, D., Lawlor, A., Dawson, K. A. & Gleeson, J. P. (2011) Tricritical point in heterogeneous k-core percolation. *Phys. Rev. Lett.*, **107**, 175703.

89. Baxter, G. J., Dorogovtsev, S. N., Goltsev, A. V. & Mendes, J. F. (2011) Heterogeneous k-core versus bootstrap percolation on complex networks. *Phys. Rev. E*, **83**, 051134.

90. Cellai, D., Lawlor, A., Dawson, K. A. & Gleeson, J. P. (2013) Critical phenomena in heterogeneous k-core percolation. *Phys. Rev. E*, **87**, 022134.

91. Dodds, P. S. & Watts, D. J. (2004) Universal behavior in a generalized model of contagion. *Phys. Rev. Lett.*, **92**, 218701.

92. Morone, F., Burleson-Lesser, K., Vinutha, H. A., Sastry, S. & Makse, H. A. (2019) The jamming transition is a k-core percolation transition. *Physica A*, **516**, 172-177.

93. Azimi-Tafreshi, N., Osat, S. & Dorogovtsev, S. (2019) Generalization of core percolation on complex networks. *Phys. Rev. E*, **99**, 022312.

94. Buldyrev, S. V., Parshani, R., Paul, G., Stanley, H. E. & Havlin, S. (2010) Catastrophic cascade of failures in interdependent networks. *Nature*, **464**, 1025.

95. Yang, Y., Nishikawa, T. & Motter, A. E. (2017) Small vulnerable sets determine large network cascades in power grids. *Science*, **358**, eaan3184.

96. Dorogovtsev, S., Goltsev, A. & Mendes, J. (2006) k-Core architecture and k-core percolation on complex networks. *Physica D*, **224**, 7–19.

97. Schwarz, J., Liu, A. J. & Chayes, L. (2006) The onset of jamming as the sudden emergence of an infinite k-core cluster. *Europhys. Lett.*, **73**, 560.

98. Parshani, R., Buldyrev, S. V. & Havlin, S. (2010) Interdependent networks: reducing the coupling strength leads to a change from a first to second order percolation transition. *Phys. Rev. Lett.*, **105**, 048701.

99. Gao, J., Buldyrev, S. V., Havlin, S. & Stanley, H. E. (2011) Robustness of a network of networks. *Phys. Rev. Lett.*, **107**, 195701.

100. Guggiola, A. & Semerjian, G. (2015) Minimal contagious sets in random regular graphs. *J. Stat. Phys.*, **158**, 300–358.

101. Ackerman, E., Ben-Zwi, O. & Wolfovitz, G. (2010) Combinatorial model and bounds for target set selection. *Theor. Comput. Sci.*, **411**, 4017–4022.

102. Dreyer Jr, P. A. & Roberts, F. S. (2009) Irreversible k-threshold processes: graph-theoretical threshold models of the spread of disease and of opinion. *Discrete Appl. Math.*, **157**, 1615–1627.

103. Reichman, D. (2012) New bounds for contagious sets. *Discrete Math.*, **312**, 1812–1814.

104. Feige, U., Krivelevich, M. & Reichman, D. (2017) Contagious sets in random graphs. *Ann. Appl. Probab.*, **27**, 2675–2697.

105. Angel, O. & Kolesnik, B. (2017) Large deviations for subcritical bootstrap percolation on the random graph. *arXiv preprint arXiv:1705.06815*.

106. Angel, O. & Kolesnik, B. (2018) Sharp thresholds for contagious sets in random graphs. *Ann. Appl. Probab.*, **28**, 1052–1098.

107. Hashimoto, K.-i. (1989) Zeta functions of finite graphs and representations of p-adic groups. *Adv. Stud. Pure Math.*, **15**, 211–280.

108. Saad, Y. (2011) *Numerical Methods for Large Eigenvalue Problems: Revised Edition*, vol. 66. Philadelphia: SIAM.

109. Morone, F., Min, B., Bo, L., Mari, R. & Makse, H. A. (2016) Collective influence algorithm to find influencers via optimal percolation in massively large social media. *Sci. Rep.*, **6**, 30062.

110. Bau, S., Wormald, N. C. & Zhou, S. (2002) Decycling numbers of random regular graphs. *Random Struct. Alg.*, **21**, 397–413.

111. Kobayashi, T. & Masuda, N. (2016) Fragmenting networks by targeting collective influencers at a mesoscopic level. *Sci. Rep.*, **6**, 37778.

112. Osat, S., Faqeeh, A. & Radicchi, F. (2017) Optimal percolation on multiplex networks. *Nat. Commun.*, **8**, 1540.

113. Teng, X., Pei, S., Morone, F. & Makse, H. A. (2016) Collective influence of multiple spreaders evaluated by tracing real information flow in large-scale social networks. *Sci. Rep.*, **6**, 36043.

114. Bovet, A., Morone, F. & Makse, H. A. (2018) Validation of Twitter opinion trends with national polling aggregates: Hillary Clinton vs Donald Trump. *Sci. Rep.*, **8**, 8673.

115. Bovet, A. & Makse, H. A. (2019) Influence of fake news in Twitter during the 2016 US presidential election. *Nat. Commun.*, **10**, 7.

116. Morone, F., Roth, K., Min, B., Stanley, H. E. & Makse, H. A. (2017) Model of brain activation predicts the neural collective influence map of the brain. *Proc. Natl. Acad. Sci. USA*, **114**, 3849–3854.

117. Luo, S., Morone, F., Sarraute, C., Travizano, M. & Makse, H. A. (2017) Inferring personal economic status from social network location. *Nat. Commun.*, **8**, 15227.

118. Szolnoki, A. & Perc, M. (2016) Collective influence in evolutionary social dilemmas. *EPL*, **113**, 58004.

119. Zhang, R. & Pei, S. (2018) Dynamic range maximization in excitable networks. *Chaos*, **28**, 013103.

120. Wang, J., Pei, S., Wei, W., Feng, X. & Zheng, Z. (2018) Optimal stabilization of Boolean networks through collective influence. *Phys. Rev. E*, **97**, 032305.

121. Wang, J., Zhang, R., Wei, W., Pei, S. & Zheng, Z. (2019) On the stability of multilayer Boolean networks under targeted immunization. *Chaos*, **29**, 013133.

122. Pei, S., Tang, S., Yan, S., Jiang, S., Zhang, X. & Zheng, Z. (2012) How to enhance the dynamic range of excitatory-inhibitory excitable networks. *Phys. Rev. E*, **86**, 021909.

123. Karp, R. M. (1972) Reducibility among combinatorial problems. *Complexity of Computer Computations*. (R. Miller ed.). New York: Springer, pp. 85–103.

124. Marinari, E. & Monasson, R. (2004) Circuits in random graphs: from local trees to global loops. *J. Stat. Mech.*, **2004**, P09004.

125. Marinari, E., Monasson, R. & Semerjian, G. (2005) An algorithm for counting circuits: application to real-world and random graphs. *EPL*, **73**, 8.

126. Bianconi, G. & Marsili, M. (2005) Loops of any size and Hamilton cycles in random scale-free networks. *J. Stat. Mech.*, **2005**, P06005.

127. Zhou, H.-J. (2013) Spin glass approach to the feedback vertex set problem. *Eur. Phys. J. B*, **86**, 455.

128. Im, Y. S. & Kahng, B. (2018) Dismantling efficiency and network fractality. *Phys. Rev. E*, **98**, 012316.

129. Mezard, M. & Montanari, A. (2009) *Information, Physics, and Computation*. Oxford: Oxford University Press.

130. Mézard, M. & Parisi, G. (2001) The Bethe lattice spin glass revisited. *Eur. Phys. J. B*, **20**, 217–233.

131. Bayati, M., Borgs, C., Braunstein, A., Chayes, J., Ramezanpour, A. & Zecchina, R. (2008) Statistical mechanics of steiner trees. *Phys. Rev. Lett.*, **101**, 037208.

132. Zdeborová, L., Zhang, P. & Zhou, H.-J. (2016) Fast and simple decycling and dismantling of networks. *Sci. Rep.*, **6**, 37954.

133. Schmidt, C., Pfister, H. D. & Zdeborová, L. (2019) Minimal sets to destroy the k-core in random networks. *Phys. Rev. E*, **99**, 022310.

134. Achlioptas, D., D'souza, R. M. & Spencer, J. (2009) Explosive percolation in random networks. *Science*, **323**, 1453–1455.

135. da Costa, R. A., Dorogovtsev, S. N., Goltsev, A. V. & Mendes, J. F. F. (2010) Explosive percolation transition is actually continuous. *Phys. Rev. Lett.*, **105**, 255701.

136. Riordan, O. & Warnke, L. (2011) Explosive percolation is continuous. *Science*, **333**, 322–324.

137. Grassberger, P., Christensen, C., Bizhani, G., Son, S.-W. & Paczuski, M. (2011) Explosive percolation is continuous, but with unusual finite size behavior. *Phys. Rev. Lett.*, **106**, 225701.

138. Friedman, E. J. & Landsberg, A. S. (2009) Construction and analysis of random networks with explosive percolation. *Phys. Rev. Lett.*, **103**, 255701.

139. Newman, M. E. J. & Ziff, R. M. (2001) Fast Monte Carlo algorithm for site or bond percolation. *Phys. Rev. E*, **64**, 016706.

140. Lipton, R. J., Rose, D. J. & Tarjan, R. E. (1979) Generalized nested dissection. *SIAM J. Numer. Anal.*, **16**, 346–358.

141. Ren, X.-L., Gleinig, N., Helbing, D. & Antulov-Fantulin, N. (2019) Generalized network dismantling. *Proc. Natl. Acad. Sci. USA*, **116**, 6554–6559.

142. Bar-Yehuda, R. & Even, S. (1981) A linear-time approximation algorithm for the weighted vertex cover problem. *J. Algorithms*, **2**, 198–203.

143. Domingos, P. & Richardson, M. (2001) Mining the network value of customers. *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, CA, USA*. New York, NY, USA: ACM, pp. 57–66.

144. Cornuejols, G., Fisher, M. L. & Nemhauser, G. L. (1977) Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms. *Manag. Sci.*, **23**, 789–810.

145. Nemhauser, G. L., Wolsey, L. A. & Fisher, M. L. (1978) An analysis of approximations for maximizing submodular set functions—I. *Math. Program.*, **14**, 265–294.

146. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J. & Glance, N. (2007) Cost-effective outbreak detection in networks. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Jose, CA, USA*. New York, NY, USA: ACM, pp. 420–429.

147. Goyal, A., Lu, W. & Lakshmanan, L. V. (2011) SIMPATH: an efficient algorithm for influence maximization under the linear threshold model. *2011 IEEE 11th International Conference on Data Mining (ICDM)*, Washington, DC, USA: IEEE Computer Society,

148. Chen, W., Wang, Y. & Yang, S. (2009) Efficient influence maximization in social networks. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris, France*. New York, NY, USA: ACM, pp. 199–208.

149. Chen, W., Wang, C. & Wang, Y. (2010) Scalable influence maximization for prevalent viral marketing in large-scale social networks. *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, DC, USA*. New York, NY, USA: ACM, pp. 1029–1038.

150. Dijkstra, E. W. (1959) A note on two problems in connexion with graphs. *Numerische Mathematik*, **1**, 269–271.

151. Cormen, T. H., Leiserson, C. E., Rivest, R. L. & Stein, C. (2009) *Introduction to Algorithms*. Cambridge: MIT Press.

152. Wang, Y., Cong, G., Song, G. & Xie, K. (2010) Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, DC, USA*. New York, NY, USA: ACM, pp. 1039–1048.

153. Nematzadeh, A., Ferrara, E., Flammini, A. & Ahn, Y.-Y. (2014) Optimal network modularity for information diffusion. *Phys. Rev. Lett.*, **113**, 088701.

154. Curato, G. & Lillo, F. (2016) Optimal information diffusion in stochastic block models. *Phys. Rev. E*, **94**, 032310.

155. Yan, S., Tang, S., Fang, W., Pei, S. & Zheng, Z. (2015) Global and local targeted immunization in networks with community structure. *J. Stat. Mech.*, **2015**, P08010.

156. Hu, Y., Ji, S., Jin, Y., Feng, L., Stanley, H. E. & Havlin, S. (2018) Local structure can identify and quantify influential global spreaders in large scale social networks. *Proc. Natl. Acad. Sci. USA*, **115**, 7468–7472.

157. Da Silva, R. A. P., Viana, M. P. & da Fontoura Costa, L. (2012) Predicting epidemic outbreak from individual features of the spreaders. *J. Stat. Mech.*, **2012**, P07005.

158. Pei, S., Muchnik, L., Andrade Jr, J. S., Zheng, Z. & Makse, H. A. (2014) Searching for superspreaders of information in real-world social media. *Sci. Rep.*, **4**, 5547.

159. Carmi, S., Havlin, S., Kirkpatrick, S., Shavitt, Y. & Shir, E. (2007) A model of Internet topology using k-shell decomposition. *Proc. Natl. Acad. Sci. USA*, **104**, 11150–11154.

160. Zeng, A. & Zhang, C.-J. (2013) Ranking spreaders by decomposing complex networks. *Phys. Lett. A*, **377**, 1031–1035.

161. Tang, S., Teng, X., Pei, S., Yan, S. & Zheng, Z. (2015) Identification of highly susceptible individuals in complex networks. *Physica A*, **432**, 363–372.

162. Malliaros, F. D., Rossi, M.-E. G. & Vazirgiannis, M. (2016) Locating influential nodes in complex networks. *Sci. Rep.*, **6**, 19307.

163. Sabidussi, G. (1966) The centrality index of a graph. *Psychometrika*, **31**, 581–603.

164. Freeman, L. C. (1977) A set of measures of centrality based on betweenness. *Sociometry*, pp. 35–41.

165. Friedkin, N. E. (1991) Theoretical foundations for centrality measures. *Am. J. Sociol.*, **96**, 1478–1504.

166. Dangalchev, C. (2006) Residual closeness in networks. *Physica A*, **365**, 556–564.

167. Brin, S. & Page, L. (1998) The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, **30**, 107–117.

168. Lü, L., Zhang, Y.-C., Yeung, C. H. & Zhou, T. (2011) Leaders in social networks, the delicious case. *PLoS One*, **6**, e21202.

169. Travençolo, B. A. N. & Costa, L. d. F. (2008) Accessibility in complex networks. *Phys. Lett. A*, **373**, 89–95.

170. Bonacich, P. (1972) Factoring and weighting approaches to status scores and clique identification. *J. Math. Sociol.*, **2**, 113–120.

171. Radicchi, F. & Castellano, C. (2016) Leveraging percolation theory to single out influential spreaders in networks. *Phys. Rev. E*, **93**, 062314.

172. Martin, T., Zhang, X. & Newman, M. E. J. (2014) Localization and centrality in networks. *Phys. Rev. E*, **90**, 052808.

173. Restrepo, J. G., Ott, E. & Hunt, B. R. (2006) Characterizing the dynamical importance of network nodes and links. *Phys. Rev. Lett.*, **97**, 094102.

174. Katz, L. (1953) A new status index derived from sociometric analysis. *Psychometrika*, **18**, 39–43.

175. Bauer, F. & Lizier, J. T. (2012) Identifying influential spreaders and efficiently estimating infection numbers in epidemic models: a walk counting approach. *EPL*, **99**, 68007.

176. KLEMM, K., SERRANO, M. Á., EGUÍLUZ, V. M. & SAN MIGUEL, M. (2012) A measure of individual role in collective dynamics. *Sci. Rep.*, **2**, 292.

177. LAWYER, G. (2015) Understanding the influence of all nodes in a network. *Sci. Rep.*, **5**, 8665.

178. GU, J., LEE, S., SARAMÄKI, J. & HOLME, P. (2017) Ranking influential spreaders is an ill-defined problem. *EPL*, **118**, 68002.

179. BORGE-HOLTHOEFER, J. & MORENO, Y. (2012) Absence of influential spreaders in rumor dynamics. *Phys. Rev. E*, **85**, 026116.

180. CHAMI, G. F., KONTOLEON, A. A., BULTE, E., FENWICK, A., KABATEREINE, N. B., TUKAHEBWA, E. M. & DUNNE, D. W. (2017) Diffusion of treatment in social networks and mass drug administration. *Nat. Commun.*, **8**, 1929.

181. KARRER, B. & NEWMAN, M. E. J. (2010) Message passing approach for general epidemic models. *Phys. Rev. E*, **82**, 016101.

182. MIN, B. (2018) Identifying an influential spreader from a single seed in complex networks via a message-passing approach. *Eur. Phys. J. B*, **91**, 18.

183. RADICCHI, F. & CASTELLANO, C. (2017) Fundamental difference between superblockers and superspreaders in networks. *Phys. Rev. E*, **95**, 012318.

184. ROGERS, E. M. (2010) *Diffusion of Innovations*. New York: Simon and Schuster.

185. CENTOLA, D. (2010) The spread of behavior in an online social network experiment. *Science*, **329**, 1194–1197.

186. COHEN, E. (1997) Size-estimation framework with applications to transitive closure and reachability. *J. Comput. Syst. Sci.*, **55**, 441–453.

187. MÉZARD, M. & ZECCHINA, R. (2002) Random k-satisfiability problem: From an analytic solution to an efficient algorithm. *Phys. Rev. E*, **66**, 056126.

188. PEI, S., TENG, X., SHAMAN, J., MORONE, F. & MAKSE, H. A. (2017) Efficient collective influence maximization in cascading processes with first-order transitions. *Sci. Rep.*, **7**, 45240.

189. JANSON, S. & THOMASON, A. (2008) Dismantling sparse random graphs. *Comb. Probab. Comput.*, **17**, 259–264.

190. SUN, J., LIU, R., FAN, Z., XIE, J., MA, X. & HU, Y. (2018) Lower bound of network dismantling problem. *Chaos*, **28**, 063128.

191. COJA-OGHLAN, A., FEIGE, U., KRIVELEVICH, M. & REICHMAN, D. (2015) Contagious sets in expanders. *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, pp. 1953–1987.

192. WORMALD, N. C. (1995) Differential equations for random processes and random graphs. *Ann. Appl. Probab.*, **5**, 1217–1235.

193. WORMALD, N. C. (1999) The differential equation method for random graph processes and greedy algorithms. *Lect. Approx. Random Algorithms*, **73**, 155.

194. WANDELT, S., SUN, X., FENG, D., ZANIN, M. & HAVLIN, S. (2018) A comparative analysis of approaches to network-dismantling. *Sci. Rep.*, **8**, 13513.

195. HOLME, P. & SARAMÄKI, J. (2012) Temporal networks. *Phys. Rep.*, **519**, 97–125.

196. BOCCALETTI, S., BIANCONI, G., CRIADO, R., DEL GENIO, C. I., GÓMEZ-GARDENES, J., ROMANCE, M., SENDINA-NADAL, I., WANG, Z. & ZANIN, M. (2014) The structure and dynamics of multilayer networks. *Phys. Rep.*, **544**, 1–122.

197. KIVELÄ, M., ARENAS, A., BARTHELEMY, M., GLEESON, J. P., MORENO, Y. & PORTER, M. A. (2014) Multilayer networks. *J Complex Netw.*, **2**, 203–271.

198. ORSINI, C., DANKULOV, M. M., COLOMER-DE SIMÓN, P., JAMAKOVIC, A., MAHADEVAN, P., VAHDAT, A., BASSLER, K. E., TOROCZKAI, Z., BOGUNÁ, M., CALDARELLI, G. & FORTUNATO, S. (2015) Quantifying randomness in real networks. *Nat. Commun.*, **6**, 8627.

199. ERKOL, Ş., FAQEEH, A. & RADICCHI, F. (2018) Influence maximization in noisy networks. *Europhys. Lett.*, **123**, 58007.

200. IRIBARREN, J. L. & MORO, E. (2009) Impact of human activity patterns on the dynamics of information diffusion. *Phys. Rev. Lett.*, **103**, 038702.

201. Gallos, L. K., Rybski, D., Liljeros, F., Havlin, S. & Makse, H. A. (2012) How people interact in evolving online affiliation networks. *Phys. Rev. X*, **2**, 031014.

202. Muchnik, L., Pei, S., Parra, L. C., Reis, S. D., Andrade Jr, J. S., Havlin, S. & Makse, H. A. (2013) Origins of power-law degree distribution in the heterogeneity of human activity in social networks. *Sci. Rep.*, **3**, 1783.

203. Min, B., Liljeros, F. & Makse, H. A. (2015) Finding influential spreaders from human activity beyond network location. *PLoS One*, **10**, e0136831.

204. Aral, S. & Walker, D. (2012) Identifying influential and susceptible members of social networks. *Science*, **337**, 337–341.

205. Aral, S. & Walker, D. (2011) Identifying social influence in networks using randomized experiments. *IEEE Intell. Syst.*, **26**, 91–96.

206. Aral, S. & Dhillon, P. S. (2018) Social influence maximization under empirical influence models. *Nat. Hum. Behav.*, **2**, 375–382.

207. Pei, S., Muchnik, L., Tang, S., Zheng, Z. & Makse, H. A. (2015) Exploring the complex pattern of information spreading in online blog communities. *PLoS One*, **10**, e0126894.

208. Teng, X., Yan, S., Tang, S., Pei, S., Li, W. & Zheng, Z. (2014) Individual behavior and social wealth in the spatial public goods game. *Physica A*, **402**, 141–149.

209. Aral, S., Muchnik, L. & Sundararajan, A. (2009) Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proc. Natl. Acad. Sci. USA*, **106**, 21544–21549.

210. Centola, D. (2011) An experimental study of homophily in the adoption of health behavior. *Science*, **334**, 1269–1272.

211. Aral, S., Muchnik, L. & Sundararajan, A. (2013) Engineering social contagions: optimal network seeding in the presence of homophily. *Netw. Sci.*, **1**, 125–153.

212. Centola, D. & Macy, M. (2007) Complex contagions and the weakness of long ties. *Am. J. Sociol.*, **113**, 702–734.

213. Muchnik, L., Aral, S. & Taylor, S. J. (2013) Social influence bias: a randomized experiment. *Science*, **341**, 647–651.