Framework of Evolutionary Algorithm for Investigation of Influential Nodes in Complex Networks

Yang Liu, Xi Wang, and Jürgen Kurths

***Abstract*—There are many target methods that are efficient to tackle the robustness and immunization problem, in particular, to identify the most influential nodes in a certain complex network. Unfortunately, owing to the diversity of networks, none of them could be accounted as a universal approach that works well in a wide variety of networks. Hence, in this paper, from a percolation perspective, we connect the immunization and robustness problem with an evolutionary algorithm, i.e., a framework of an evolutionary algorithm for investigation of influential nodes in complex networks, in which we have developed procedures of selection, mutation, and initialization of population as well as maintaining the diversity of population. To validate the performance of the proposed framework, we conduct intensive experiments on a large number of networks and compare it to several state-of-the-art strategies. The results demonstrate that the proposed method has significant advantages over others, especially on empirical networks in most of which our method has over 10% advantages of both optimal immunization threshold and average giant fraction, even against the most excellent existing strategies. Additionally, our discussion reveals that there might be better solutions with various initial methods.**

***Index Terms*—Complex networks, evolutionary algorithm, immunization, percolation transition, robustness.**

I. Introduction

WHILE systems are modeled by a graph whose nodes represent the dynamic units and edges capture their interactions [1], problems, like investigating efficient strategies to prevent the outbreak of viruses [2], [3] or designing more reliable communication networks, could be mapped onto finding a set of nodes. After the removal (immunized or protected) of that set, the network is more effective to suppress the propagation of information or keep the function of the system [4]–[7]. This problem, as one of the fundamental structural problems in network science, is also called immunization or robustness problem [8] and has recently attracted increasing attention in a large range of domains [1], [9]–[12].

Indeed, due to the heterogeneous nature of real networks, the roles that different nodes play might be far from each other in a network [9]. In other words, protecting one node may be more significant for the network function than protecting another one. Thus, if a certain state is given (for example, to collapse the network to a given size of the giant component), the immunization or robustness problem could be addressed by minimizing the size of removal set for the purpose of saving resources. To identify this set, numerous methods have been developed in the past few years, from the demand of local information (e.g., counting the number of neighbors) to the capture of global structure (such as counting the number of shortest paths that a node is located on [13]), and further to the study of dynamic process (like heuristic strategies [8], [14], [15]).

Generally, the more topological information a method considers, the better the performance which the method will achieve. For instance, because of the knowledge of neighbors' information, the acquaintance strategy [16], [17] is usually much better than the random immunization [18] which randomly selects immune nodes without any preference, but the acquaintance strategy tries to immunize one of the neighbors of those selected nodes (with higher probability to immunize the hub nodes). In addition, in contrast to the high degree centrality strategy (HD) immunizing the hub nodes directly [19], the high adaptive degree centrality strategy (HAD) repeatedly removes the highest-degree node from the remained network after the removal of the previous highest-degree node. Thus, HAD is also much better than HD in most situation. Further, a similar strategy (to HAD) is adopted by the collective influence (CI) method [14] which considers several layers of neighbors' information, so that it is more effective than HAD. Moreover, the betweenness centrality strategy takes account of the shortest-path flows among each pairwise nodes and identifies the influential nodes which frequently appear on those

Manuscript received April 9, 2018; revised October 4, 2018 and January 3, 2019; accepted February 14, 2019. Date of publication February 22, 2019; date of current version November 27, 2019. This work was supported in part by the High Performance Computer System at the Potsdam Institute for Climate Impact Research through the European Regional Development Fund, the German Federal Ministry of Education and Research and the Land Brandenburg, in part by the China Scholarship Council Scholarship, and in part by Deutsche Forschungsgemeinschaft/Fundação de Amparo à Pesquisa do Estado de São Paulo (DFG/FAPESP) under Grant IRTG 1740/TRP 2015/50122-0. *(Corresponding author: Yang Liu.)*

Y. Liu is with the Department of Complexity Science, Potsdam Institute for Climate Impact Research, 14473 Potsdam, Germany, and also with the Department of Computer Science, Technische Universität Berlin, 10587 Berlin, Germany (e-mail: yangliuyh@gmail.com).

X. Wang is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.

J. Kurths is with the Department of Complexity Science, Potsdam Institute for Climate Impact Research, 14473 Potsdam, Germany, also with the Department of Physics, Humboldt University Berlin, 12489 Berlin, Germany, and also with Saratov State University, 410012 Saratov, Russia.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2019.2901012

shortest paths. And thus the high adaptive betweenness centrality strategy (HAB) is quite effective in most situations, but it is limited due to its high time complexity. Other methods, like the belief propagation-guided decimation strategy viewing networks as conditional random fields [15], are continuously developed to handle the robustness and immunization problem [8], [14], [20]–[26]; however, none of them could be accounted as a universal approach that works well in a wide variety of networks, especially in real-world networks.

In fact, the immunization and robustness problem is actually an NP-hard problem [14]. Undeniably, the evolutionary algorithm is actually a prevalent solution to address the NP-hard problem, which has been proved effective to solve a great number of such problems over the past decades, like the traveling salesman problem [27], the many-objective knapsack problem [28], [29], or the community and controlling regions detection problems in complex networks [30]–[33]. But unfortunately, there is so far no effective way which can connect an evolutionary algorithm with the immunization and robustness problem. Besides, in [34], we have found that the occupation of a node would, directly or indirectly, affect the occupation of other nodes when studying on a given network. In this way, we consider not only the information from the nearest nodes [14], [23], but also the information from other nodes. Thus, we have the basic methods [34] which could notably inhibit the average giant (susceptible) fraction in most networks but only make an acceptable immunization threshold. The fact is that there is a conflict between the optimal average giant fraction and the immunized threshold.

Hence, in this paper, for obtaining the optimal immunized threshold, we develop a framework of evolutionary algorithm for investigating influential nodes. The main contributions are listed below.

1) A new goal function is proposed to cope with the conflict between the optimal threshold and average giant fraction. On the one hand, the threshold can benefit from minimizing the average giant fraction at the beginning (i.e., converge more quickly) and sometimes yield better results compared to the direct optimization of the threshold. On the other hand, with the increase of iterations, a smaller average giant fraction might induce a worse threshold. In this regard, we develop a new goal function which captures the positive feedback from the average giant fraction and deals with the conflict in the meantime.
2) To further handle the conflict, local and global mutation operators as well as a special selection strategy are developed. While studying the conflict in detail, we find that a larger giant fraction might promote a better threshold. Therefore, we design a special selection method to choose the offspring with larger giant fraction by higher probability, which is numerically demonstrated quite effective.
3) In addition, a similar idea is also generalized to optimize the average giant fraction.
4) Besides, an initialization strategy of population is also designed.
5) Lastly, we validate the efficacy of the proposed method on both real-world and synthetic networks compared to the state-of-the-art strategies [14], [21], [23], [34] using percolation metric and the susceptible-infectious-recovery (SIR) epidemic spreading model [35]. Explicitly, experimental results manifest that our method overwhelms others significantly, especially demonstrates excellent performance on empirical networks.

The rest of this paper is organized as follows. In Sections II and III, the problem, related methods, and information of networks are briefly introduced. Afterward, Section IV in detail describes the proposed strategy including several temporary results. Following that, applications and the regarding performance are conducted and evaluated in Section V. Finally, a discussion and the conclusions are given in Sections VI and VII, respectively.

II. Preliminaries

A. Problem

We consider a network composed of $n = |\mathcal{N}|$[1] nodes tied by $m = |\mathcal{M}|$ edges, where $\mathcal{N}$ and $\mathcal{M}$ are the node set and the edge set, respectively. Assuming that $\mathcal{C}$ is the connected component set of the network, then

$$\bigcap_{i=1}^{|\mathcal{C}|} c_i = \emptyset, \ \bigcup_{i=1}^{|\mathcal{C}|} c_i = \mathcal{N} \tag{1}$$

where $c_i \in \mathcal{C}$ is a subset of $\mathcal{N}$ and $|\mathcal{C}|$ represents the size of $\mathcal{C}$. Further, for two nodes $v_\hbar$ and v_z in $\mathcal{N}$, we use $v_\hbar \leftrightarrow v_z$ to denote that there is a path between them, i.e., $v_\hbar$ can reach v_z through the edges in $\mathcal{M}$. Thus, we have

$$\begin{aligned} &v_\hbar \leftrightarrow v_z, \ \forall v_\hbar, v_z \in c_i, \hbar \neq z \\ &v_\hbar \nleftrightarrow v_z, \ \forall v_\hbar \in c_i, \forall v_z \in c_j, i \neq j \end{aligned} \tag{2}$$

in which $v_\hbar \nleftrightarrow v_z$ means $v_\hbar$ cannot reach v_z. With those, the giant connected component of the network is defined to be

$$c_{\max} := \arg\max_{c_i} \mathcal{G}_{c_i}, \forall c_i \in \mathcal{C} \tag{3}$$

where $\mathcal{G}_{c_i} = |c_i|$ is the size of the component c_i. Further, let S_a be an arbitrary configuration (sequence) of $\mathcal{N}$, namely $\{S_a(i)|i \in [1, n]\} \equiv \mathcal{N}$, where $S_a(i)$ corresponds to a unique node in $\mathcal{N}$. We define the immunization threshold $q_c(S_a)$ as a function of S_a

$$q_c(S_a) := \min q, G(S_a; q) \leq \theta, \forall q \in [0, 1] \tag{4}$$

where θ is a given value and $G(S_a; q) = \mathcal{G}_{c_{\max}}/n$ is the fraction that a node is part of the giant connected component after the removal of all nodes in $\{S_a(i)|i \in [1, \lfloor n \times q \rfloor]\}$, including the incidental edges. Moreover, denoting with $F(S_a)$

$$F(S_a) := \frac{1}{n} \sum_{q=1/n}^{1} G(S_a; q) \tag{5}$$

the related average fraction of giant components of S_a (the average susceptible fraction), then the solution to the immunization (robustness) problem is to search for the optimal

[1]A main nomenclature list is referred to the Appendix.

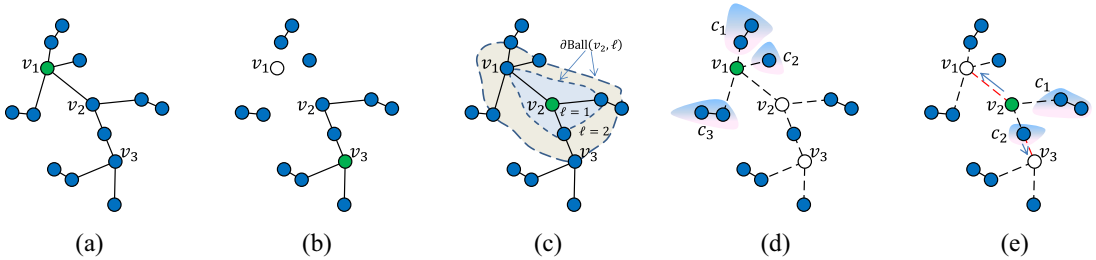


Fig. 1. Illustration of HAD, CI, EI, and PR. (a) and (b) HAD method immunizes a network by repeatedly removing the highest-degree node (node v_3) from the remaining network after the removal of the highest-degree node (node v_1). (c) $\partial\text{Ball}(v_2, 1)$ and $\partial\text{Ball}(v_2, 2)$ can be identified by the two types of dotted line. (d) In this example, the occupied nodes are colored with blue and others are the nonoccupied nodes (candidates). Considering node v_1, the node in component c_2 is a leaf, $\mathcal{G}_{c_1} = 2$ and $\mathcal{C}(v_1) = \{c_1, c_2, c_3\}$. If $K = 3$ and the calculation of v_2 is in front of v_1, then only node v_1 is a hub. (e) $\xi(v_2) = 4/14$.

sequence S_{opt} satisfying

$$S_{\text{opt}} := \begin{cases} \arg\min_{S_a} q_c(S_a), \ \forall a \in [1, n!], & \text{if } \theta \text{ is given} \\ \arg\min_{S_a} F(S_a), \ \forall a \in [1, n!], & \text{otherwise} \end{cases} \tag{6}$$

where $\{S_a | a \in [1, n!]\}$ means all the permutation configurations of $\mathcal{N}$. Note that in this paper, θ is fixed to 0.01 except for a special explanation.

Now, we inverse this process (4) and (5) as a percolation transition [change the process from the removal of the most influential node S_a to the occupation of the least important node S_a^ϵ, which is implemented by inversing S_a, scilicet, $S_a^\epsilon(1) = S_a(n), S_a^\epsilon(2) = S_a(n-1)$, and so forth]. Moreover, let $\mathcal{N}_\epsilon(S_a^\epsilon; t) = \{S_a^\epsilon(i) | i \in [1, t]\}$ be the node set of the temporary network consisting of $t = |\mathcal{N}_\epsilon(S_a^\epsilon; t)|$ occupied nodes and the related edges. In particular, $\mathcal{N}_\epsilon(S_a^\epsilon; t = 0)$ represents an empty network. Then, (5) can be rewritten as

$$F(S_a^\epsilon) \equiv \frac{1}{n} \sum_{t=1}^{n} G_\epsilon(S_a^\epsilon; t) \tag{7}$$

where $G_\epsilon(S_a^\epsilon; t)$ is the corresponding giant fraction after the occupation of $\mathcal{N}_\epsilon(S_a^\epsilon; t)$. Note in the following sections, the symbol ϵ indicates that the process is in percolation transition (7), otherwise, in normal regime (5).

B. Related Methods

It is possible to tackle this problem by approaches based on main properties of networks, like the centrality of networks [36] including the degree centrality (remove or immunize the part of nodes with the highest degree, called the HD) and the betweenness centrality [the high betweenness centrality strategy (HB)], pagerank [37] or k-shell decomposition [38] etc. [8], [39]. Meanwhile, we can also think about heuristic algorithms developed from these basic strategies, e.g., the HAD. Furthermore, we could handle it by means of percolation-based methods, e.g., the inverse targeting (IVS) approach [21], the CI strategy [14], or the explosive immunization (EI) method [23]. Here, we mainly introduce three of them, namely HAD, CI, and EI.

Denoting the degree of the node $v_i \in \mathcal{N}$ with k_{v_i}, the HAD method immunizes a network by repeatedly removing the highest-degree node from the remained network after the removal of the previous highest-degree node [Fig. 1(a) and (b)].

Let ℓ be the length of the shortest path between two nodes and $\partial\text{Ball}(v_i, \ell)$ be the node set consisting of all the nodes with ℓ-length shortest path to node v_i [Fig. 1(c)]. The CI strategy (for large networks) first identifies each node through $\text{CI}_\ell(v_i)$ (the CI strength of node v_i at level ℓ [14])

$$\text{CI}_\ell(v_i) = (k_{v_i} - 1) \sum_{v_j \in \partial\text{Ball}(v_i, \ell)} (k_{v_j} - 1). \tag{8}$$

Then, analogous to HAD, each time CI chooses the node with highest CI_ℓ to be immunized from the remained network after the top CI_ℓ node was removed.

Letting $\mathcal{C}(v_i)$ denote the occupied component set which comprises all the components that node v_i would connect to; then the EI method (based on percolation transition) selects the node with minimal σ to be occupied from a random set of candidates [randomly select part of nodes from node set $\mathcal{N}^{\text{rc}} = \mathcal{N} \backslash \mathcal{N}_\epsilon(t)$]

$$\sigma_{v_i} = k_{v_i}^{(\text{eff})} + \sum_{c_j \in \mathcal{C}(v_i)} (\sqrt{\mathcal{G}_{c_j}} - 1) \tag{9}$$

where $k_{v_i}^{(\text{eff})}$ is used to quantify the potential danger of node v_i in the epidemic spread [23], i.e., a node with large $k_{v_i}^{(\text{eff})}$ indicates that it is more likely to spread the epidemic out

$$k_{v_i}^{(\text{eff})} = k_{v_i} - L_{v_i} - M_{v_i}(\{k_{v_j}^{(\text{eff})} | v_j \in \Gamma(v_i)\}) \tag{10}$$

in which L_{v_i} is the number of leaves (nodes with degree 1), $\Gamma(v_i)$ denotes all of v_i's nearest neighbors, and $M_{v_i}(\{k_{v_j}^{(\text{eff})}\})$ is the number of strong hubs (nodes with $k_{v_j}^{(\text{eff})} \geq K$, K is an undetermined parameter, usually $K = 6$ [23]). An example is shown in Fig. 1(d). Note that in this paper we only consider $\sigma_{v_i}^{(1)}$ of [23] because it has better results of both q_c and F in most real-world networks compared to $\sigma_{v_i}^{(2)}$.

C. Union-Find Algorithm

The union-find (UF) algorithm [40] is designed for the UF set which is a data structure that can track disjoint subsets when the elements are partitioned into several groups. One of its main application is to identify connected components in networks. Let $\mathcal{N}_p$ be the parent set associated with the node set $\mathcal{N}$ and $\mathcal{G}(v_i)$ be the size of the component that node v_i belongs to, respectively. The UF algorithm can help us, within nearly constant time (for ER-type networks), to merge components (union) or to determine whether any two nodes are in the same

Algorithm 1 UF Algorithm

function ROOT(v_i, $\mathcal{N}_p$)
  while $v_i \neq \mathcal{N}_p(v_i)$ **do**
    $\mathcal{N}_p(v_i) \leftarrow \mathcal{N}_p(\mathcal{N}_p(v_i))$
    $v_i \leftarrow \mathcal{N}_p(v_i)$
  end while
  return v_i
end function

function FIND(v_i, v_j, $\mathcal{N}_p$)
  return ROOT(v_i, $\mathcal{N}_p$) = ROOT(v_j, $\mathcal{N}_p$)
end function

function UNION(v_i, v_j, $\mathcal{N}_p$, $\mathcal{G}$)
  $v_\hbar \leftarrow$ ROOT(v_i, $\mathcal{N}_p$)
  $v_z \leftarrow$ ROOT(v_j, $\mathcal{N}_p$)
  if $\mathcal{G}(v_\hbar) < \mathcal{G}(v_z)$ **then**
    $\mathcal{N}_p(v_\hbar) \leftarrow v_z$
    $\mathcal{G}(v_z) \leftarrow \mathcal{G}(v_z) + \mathcal{G}(v_\hbar)$
    return $\mathcal{G}(v_z)$
  else
    $\mathcal{N}_p(v_z) \leftarrow v_\hbar$
    $\mathcal{G}(v_\hbar) \leftarrow \mathcal{G}(v_\hbar) + \mathcal{G}(v_z)$
    return $\mathcal{G}(v_\hbar)$
  end if
end function

component (find). The details of this algorithm are shown in Algorithm 1.

D. Basic Relationship Related Method

The relationship related (RR) method is mainly based on the assumption of the existence of a mutual influence among nodes [34]. Therefore, RR also starts with an empty and arbitrary sequence S_{old} (or a certain strategy, e.g., HD). Each time RR selects the node v_s which would minimize the corresponding rule function $\xi(\cdot)$ [the node satisfies $v_s = \arg\min_{v_\hbar} \xi(v_\hbar)$] from a random set of candidates $\mathcal{N}^{\text{rc}}(t)$ to be occupied, where $\xi(v_\hbar)$ is defined as the following two cases (corresponding to the sum rule and the product rule [41], respectively):

$$\xi(v_\hbar) = \begin{cases} 1 + \sum_{c_i \in \mathcal{C}(v_\hbar)} \mathcal{G}_{c_i} \\ 1 + \prod_{c_i \in \mathcal{C}(v_\hbar)} \mathcal{G}_{c_i}. \end{cases} \tag{11}$$

The random set $\mathcal{N}^{\text{rc}}(t)$ consists of the nodes which are from $\{S^{\epsilon}_{\text{old}}(i) | i \in [t+1, \min(\lfloor t + r \times n \rfloor, n)]\}$ through τ times random selection, where $r = r_s/(T \times \delta_r + 1)$, $\tau = \tau_s + \lfloor T \times \delta_\tau + 0.5 \rfloor$, and $T \leq \hat{T}$. r_s, $\hat{T}$, δ_r, τ_s, and δ_τ are undetermined parameters (more details refer to [34]). Then, update $S^{\epsilon}_{\text{old}}$ and repeat the occupation until all nodes are occupied. Note that here S_{old} keeps unchanged while $S^{\epsilon}_{\text{old}}$ is updated as the increase of t, and we use S_{new} to denote the inverse of $S^{\epsilon}_{\text{old}}$ for convenient description. Finally, compare S_{new} with S_{old} through (6), and choose the better one as S_{old} for another S_{new} (Algorithm 2). An example of RR is shown in Fig. 2.

E. Basic Prediction Relationship Method

The prediction relationship (PR) method [34] is specially designed for model networks. The main idea of this method is trying to keep the occupied components away from as many high degree nodes as possible, since there are hardly any nodes with low degree that are more important than those with high degree in a synthetic network. PR first identifies each node

Algorithm 2 RR Method

Input: *Network*, $S^{\epsilon}_{\text{old}}$, r_s, $\hat{T}$, δ_r, τ_s, δ_τ
Output: $S^{\epsilon}_{\text{new}}$
$T \leftarrow 1$
while $T \leq \hat{T}$ **do**
  $S^{\epsilon}_{\text{new}} \leftarrow S^{\epsilon}_{\text{old}}$
  $\mathcal{N}_p(v_i) \leftarrow v_i$, $\mathcal{G}(v_i) \leftarrow 1$ // Initialize for UF algorithm
  $t \leftarrow 1$
  $r \leftarrow r_s/(T \times \delta_r + 1)$
  $\tau \leftarrow \tau_s + \lfloor T \times \delta_\tau + 0.5 \rfloor$
  while $t \leq n$ **do**
    $\mathcal{N}^{\text{rc}}(t) \leftarrow$ based on t, r and τ
    $v_s \leftarrow \arg\min_{v_\hbar} \xi(v_\hbar)$, $v_\hbar \in \mathcal{N}^{\text{rc}}(t)$ // UF algorithm
    $S^{\epsilon}_{\text{new}} \leftarrow$ update itself
    $t \leftarrow t + 1$
  end while
  if S_{new} is better than S_{old} **then** // according to Eq. (6)
    $S^{\epsilon}_{\text{old}} \leftarrow S^{\epsilon}_{\text{new}}$
  end if
  $T \leftarrow T + 1$
end while

based on the distribution of node degree

$$\mathcal{H}_{v_\hbar} = 1 - \sum_{k_{v_j} < k_{v_\hbar}} p(k_{v_j}) = \sum_{k_{v_j} \geq k_{v_\hbar}} p(k_{v_j}) \tag{12}$$

where $p(k_{v_j})$ is the probability of nodes with degree k_{v_j}. Then, similar to RR, we construct the rule function $\xi(\cdot)$ and choose the node v_s which would minimize $\xi(\cdot)$ to be occupied

$$\xi(v_\hbar) = \sum_{c_i \in \mathcal{C}(v_\hbar)} \sum_{v_j \in c_i} \sum_{v_z \in \Gamma^u(v_j)} \mathcal{H}_{v_z} + \sum_{v_z \in \Gamma^u(v_\hbar)} \mathcal{H}_{v_z} \tag{13}$$

in which $\Gamma^u(v_j)$ denotes all of the v_j's nearest-unoccupied neighbors (here view $v_\hbar$ as occupied node). In this way, those components would manage to be connected by just a few high-degree nodes [Fig. 1(e)].

III. NETWORKS

The network associated with the robustness and immunization problem mainly corresponds to three categories: 1) infrastructure networks [4], [6]; 2) population or computer related networks [2], [3]; and 3) information transition networks [5]. For example, the failure of a transformer in the power grid network or an accident in a crossroad in the road network might devastate the whole system like blackouts or traffic jam. In this regard, we choose the real-world networks including a power grid network [42], [43] (power) and two road networks [44] (including roadNet-PA and roadNet-TX) belong to infrastructure networks; the Scottish cattle movements network [23], autonomous systems graphs [45] (including as-733 and as-Skitter) and two Internet peer-to-peer networks [46], [47] (including p2p-Gnutella08 and p2p-Gnutella31) are population or computer related networks; and three collaboration networks [46] (including ca-GrQc, ca-AstroPh, and ca-CondMat), two citation networks [45], [48] (including hep-th and cit-HepTh), two communication networks (including Email-Enron [44], [49] and Email-EuAll [46]), one location-based online social network [50] (loc-Gowalla), the Amazon product co-purchasing network [51] (com-Amazon), and the Google Web graph [44] (Web-Google) represent information

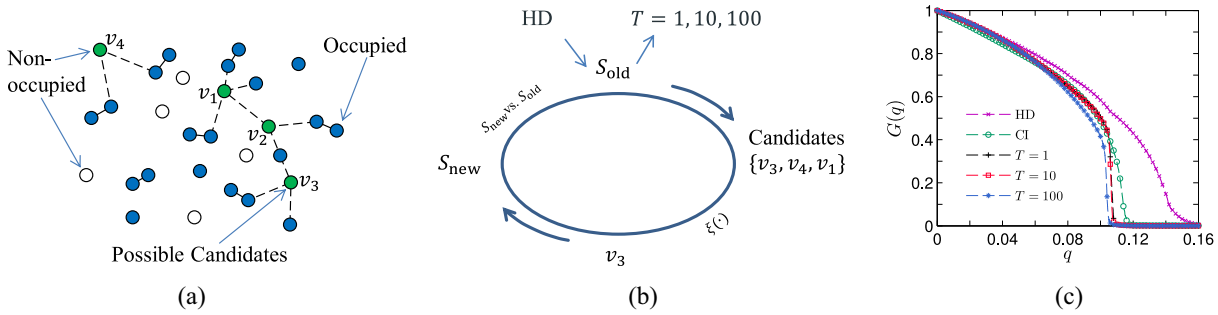


Fig. 2. Illustration of the RR method. (a) and (b) Example under product rule (11), where the occupied nodes are colored with blue and the others are the nonoccupied nodes (possible candidates colored with green). The new occupied node v_3 is chosen from the random candidate set $\mathcal{N}^{\text{rc}}(20) = \{v_3, v_4, v_1\}$ because $\xi(v_3) = 3$ compared to $\xi(v_1) = 5$ and $\xi(v_4) = 5$. In (c), we show the performance of RR based on HD (namely, start with $S_{\text{old}} = \text{HD}$) for different repeated times T on SF networks with the exponential parameter $\gamma = 3.0$, average degree $\langle k \rangle = 4.0$ and $n = 10^5$ (over 50 implementations). RR and CI are with $r_s = F(\text{HD})$, $\tau_s = 10$, $\delta_r = 0.01$, $\delta_\tau = 0.01$, and $\ell = 4$, respectively.

TABLE I
BASIC INFORMATION OF THE 18 REAL-WORLD NETWORKS

Network	n	m	Network	n	m
Power	4941	6594	Email-Enron	36692	183831
CA-GrQc	5242	14490	p2p-Gnutella31	62586	147892
p2p-Gnutella08	6301	20777	loc-Gowalla	196591	950327
as-733	6474	12572	Email-EuAll	265214	364481
Scottish	7228	24784	com-Amazon	334863	925872
CA-AstroPh	18771	198050	web-Google	875713	4322051
CA-CondMat	23133	93439	PAroad	1088092	1541898
hep-th	27240	341923	Txroad	1379917	1921660
Cit-HepPh	34546	420877	as-Skitter	1696415	11095298

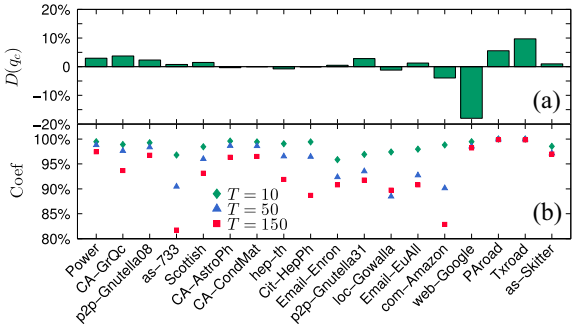


Fig. 3. Different percentage $D(q_c) = (q_c^1 - q_c^2)/[(q_c^1 + q_c^2)/2]$ of q_c^1 and q_c^2, and the Pearson correlation coefficient (Coef) between q_c^1 and F versus T on the 18 real-world networks. q_c^1 and q_c^2 are obtained through the optimization of F and q_c, respectively. Twenty independent realizations are conducted on each network.

transition networks. Most of them can also be classified into other categories, such as peer-to-peer network can be viewed as information transition networks. Some basic information regarding these networks can be found in Table I.[2] Note that for all networks considered here, the directed edges are simply replaced with undirected ones, and also all self-loops and isolated nodes are entirely removed. In addition, without loss of generality, we also conduct our experiments on paradigmatic model networks [including Erdös–Rényi (ER) [52] and scale-free (SF) [43] networks].

IV. METHOD

These two basic methods, RR and PR, are always more effective than CI, EI, and also the approaches in [15], [21], and [39], especially RR in real-world networks [34]. Besides, simulation results show that the optimization of the average giant fraction F oftentimes obtains similar and sometimes much smaller immunized threshold q_c compared to the direct optimization of q_c (6) through RR [Fig. 3(a)]. Moreover, while studying the Pearson correlation coefficient (Coef) between F and q_c by optimizing F [Fig. 3(b)], we find that Coef decreases as the rise of repeated times T. These facts are twofold. On the one hand, the optimization of F might help us find a better q_c, i.e., there may be a positive feedback between F and q_c. On the other hand, undoubtedly there is also a conflict between the optimal q_c and F. Therefore, in the following part of the method, we first focus on the optimization of the immunized threshold q_c by capturing the positive feedback from the average giant fraction F and coping with the conflict between them at the same time, so that the framework of evolutionary algorithm for investigation of the influential nodes is developed, including the goal function, selection, mutation, and initialization of population. Then, a similar idea is also used to optimize F.

A. Goal Function

For a certain network, it is obvious that if the occupied node set is certain, the size of the giant component would be certain too, no matter what configuration the node set has, namely $G_\epsilon^i(t) = G_\epsilon^j(t)$ if $\mathcal{N}_\epsilon^i(t) = \mathcal{N}_\epsilon^j(t)$. Thus, if we partition the nodes of a network into several groups and make the calculation (RR) happen only on their own group (Fig. 4); then the threshold q_c would, to some extend, only directly depend on the group where the critical node belongs to. In other words, we here try to use the other groups to catch the positive feedback from F and deal with the conflict on the group where the critical node is part of.

In detail, given an arbitrary sequence S_a of the node set $\mathcal{N}$, we denote the corresponding occupied node set at $t + \delta t$ by

$$\mathcal{N}_\epsilon(S_a^\epsilon; t + \delta t) = \{S_a^\epsilon(i) | i \in [1, t + \delta t]\}, t + \delta t \le n \quad (14)$$

where $\delta t \in \mathbb{N}$ is the size of each group. Then, it is easy to verify that the configuration of the node set $\mathcal{N}_\epsilon(S_a^\epsilon; t +$

[2]The source data of these networks is either from http://snap.stanford.edu/data or http://konect.uni-koblenz.de/networks/opsahl-powergrid.

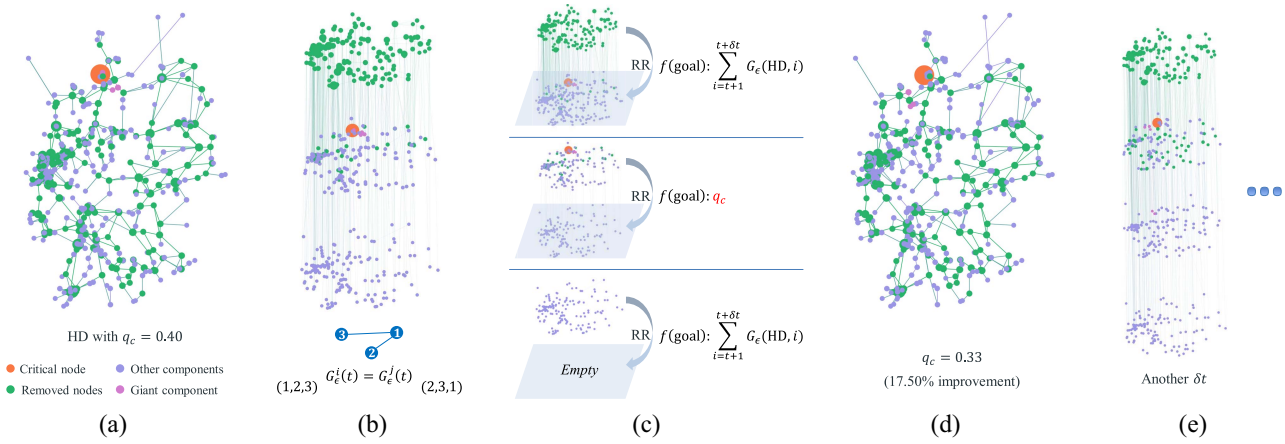


Fig. 4. Illustration of Evol$_1$ on the German Power Grid network. (a) q_c(HD) = 0.40. (b) and (c) Divide the nodes into three groups ($\delta t = \lfloor n/3 \rfloor$) and make the calculation happen only on their own group, then the threshold q_c only directly depends on the group where the critical node belongs to. (d) New sequence is obtained and has an improvement of 17.50% of q_c compared to HD. (e) Other δt for a better performance. Note that RR is under sum rule with $\hat{\tau} = 50$, $\hat{r} = 0.9$, and $\hat{T} = 20$.

$\delta t)\backslash\mathcal{N}_\epsilon(S_a^\epsilon; t)$ is independent of the occupied order of nodes in the sets $\mathcal{N}_\epsilon(S_a^\epsilon; t)$ and $\mathcal{N}\backslash\mathcal{N}_\epsilon(S_a^\epsilon; t+\delta t)$. Namely, the change of $F_2(S_a)$ would not bring about any influence to $F_1(S_a)$ and $F_3(S_a)$ if we let $F_1(S_a)$, $F_2(S_a)$, and $F_3(S_a)$

$$F_1(S_a) = \sum_{i=1}^{t} G_\epsilon(S_a^\epsilon; i)$$
$$F_2(S_a) = \sum_{i=t+1}^{t+\delta t} G_\epsilon(S_a^\epsilon; i)$$
$$F_3(S_a) = \sum_{i=t+\delta t+1}^{n} G_\epsilon(S_a^\epsilon; i)$$

be associated with $\mathcal{N}_\epsilon(S_a^\epsilon; t)$, $\mathcal{N}_\epsilon(S_a^\epsilon; t+\delta t)\backslash\mathcal{N}_\epsilon(S_a^\epsilon; t)$, and $\mathcal{N}\backslash\mathcal{N}_\epsilon(S_a^\epsilon; t+\delta t)$, respectively. Denoting with v_c the critical node corresponding to the last removed node (4) after which $G(S_a; q) \le \theta$, then our goal function is defined as

$$f(\text{goal}) = \begin{cases} q_c, & \text{if } v_c \in \mathcal{N}_\epsilon(S_a^\epsilon; t+\delta t)\backslash\mathcal{N}_\epsilon(S_a^\epsilon; t) \\ \sum_{i=t+1}^{t+\delta t} G_\epsilon(S_a^\epsilon; i), & \text{otherwise.} \end{cases} \quad (15)$$

Consequently, starting with $\mathcal{N}_\epsilon(S_a^{\epsilon 0}; t = 0)$ and a certain $\delta t \in [1, \hat{\delta t}]$, we orderly occupy the nodes in the node set $\mathcal{N}_\epsilon(S_a^\epsilon; t+\delta t)\backslash\mathcal{N}_\epsilon(S_a^\epsilon; t)$ based on RR, where $\hat{\delta t}$ is an undetermined parameter; following that we further handle the nodes in $\mathcal{N}_\epsilon(S_a^\epsilon; t+2\delta t)\backslash\mathcal{N}_\epsilon(S_a^\epsilon; t+\delta t)$, in $\mathcal{N}_\epsilon(S_a^\epsilon; t+3\delta t)\backslash\mathcal{N}_\epsilon(S_a^\epsilon; t+2\delta t)$ etc. Note that these procedures can be processed in parallel. Then, we will get a new sequence $S_a^{\epsilon 1}$, as well as $S_a^{\epsilon 2}, \ldots, S_a^{\epsilon \hat{T}_g}$ with other δt.

In addition, since usually $\delta t \ll n$ (note that δt is the main factor to influence the parallel computation), the adaptive strategy [i.e., $r = r_s/(T \times \delta_r + 1)$ and $\tau = \tau_s + \lfloor T \times \delta_\tau + 0.5 \rfloor$] in RR [34] is not valid any more. So we here adjust them to be randomly selected from $r \in (0, \hat{r}]$ and $\tau \in [1, \hat{\tau}]$ with same probability, where $\hat{r} \le 1$ and $\hat{\tau} \in \mathbb{Z}^+$ are undetermined parameters accordingly for the upper bounds of the proportion of possible candidates and the times of selection (Evol$_1$). An example is detailed in Fig. 4 and the related processes are shown in Algorithm 3. In Fig. 5, we also show

Algorithm 3 Evol$_1$ Method

Input: *Network*, $\hat{r}$, $\hat{\tau}$, $\hat{T}$, $\hat{\delta t}$, $S_a^{\epsilon 0}$, $\hat{T}_g$
Output: $S_a^{\epsilon \hat{T}_g}$
1: $T_g \leftarrow 0$
2: $S_a^{\epsilon T_g} \leftarrow S_a^{\epsilon 0}$
3: **while** $T_g < \hat{T}_g$ **do**
4: $\delta t \in [1, \hat{\delta t}]$ // δt is a random number
5: Divide $\mathcal{N}$ into several group based on $S_a^{\epsilon T_g}$ and δt: $\mathcal{N}_\epsilon(S_a^\epsilon; \delta t)\backslash\mathcal{N}_\epsilon(S_a^\epsilon; 0)$, $\mathcal{N}_\epsilon(S_a^\epsilon; 2\delta t)\backslash\mathcal{N}_\epsilon(S_a^\epsilon; \delta t)$, …, $\mathcal{N}_\epsilon(S_a^\epsilon; n)\backslash\mathcal{N}_\epsilon(S_a^\epsilon; (\lfloor \frac{n}{\delta t} \rfloor - 1)\delta t)$
6: Initialize $\mathcal{N}_p(v_i)$ and $\mathcal{G}(v_i)$ for each group according to $S_a^{\epsilon T_g}$
7: At each group call RR (Algorithm 2) with different goal function (Eq. (15))
//Parallel, and here RR with $r \in (0, \hat{r}]$ and $\tau \in [1, \hat{\tau}]$
8: $T_g \leftarrow T_g + 1$
9: **end while**
10: $S_a^{\epsilon \hat{T}_g} \leftarrow S_a^{\epsilon T_g}$

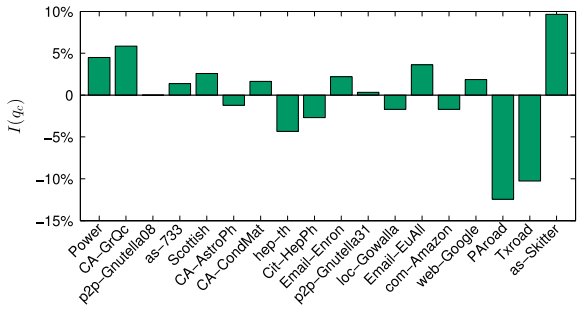


Fig. 5. Improvement percentage $I(q_c) = [q_c(\text{RR}) - q_c(\text{Evol}_1)]/q_c(\text{RR})$ of q_c for Evol$_1$ compared to RR on the 18 real-world networks.

the performance of Evol$_1$ on the 18 real-world networks compared to RR, where RR is with the same settings of parameters as [34] and Evol$_1$ is based on HD with $\hat{\tau} = 50$, $\hat{r} = 1$, $\hat{T} = 20$, $\hat{\delta t} = \lfloor 0.1 \times n \rfloor$, and $\hat{T}_g = 5000$ for networks with $n \le 10^5$, $\hat{T}_g = 2500$ for $10^5 < n \le 10^6$, and $\hat{T}_g = 500$ for $n > 10^6$, respectively. These settings are also conducted in the following part of this paper if there is no special explanation.

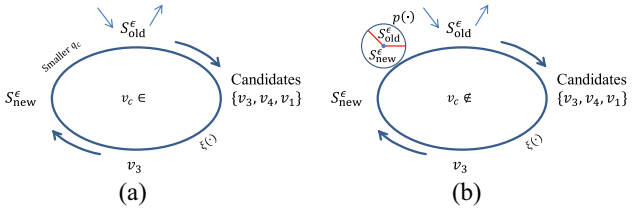


Fig. 6. Illustrations of Selection-I. (a) In the group where the critical node v_c belongs to, the sequence with smaller q_c is chosen directly. (b) In the other groups the selection tends to a smaller average giant fraction (16).

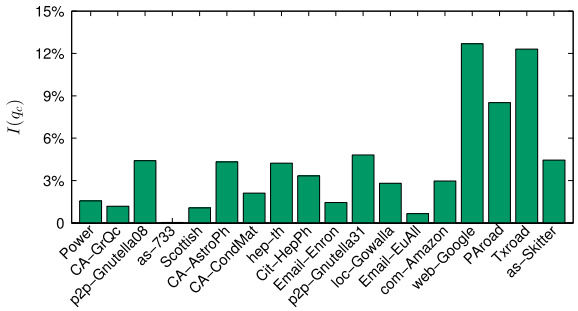


Fig. 7. Improvement percentage $I(q_c)$ of q_c for Evol_2 compared to Evol_1 on the 18 real-world networks where $I(q_c) = [q_c(\text{Evol}_1) - q_c(\text{Evol}_2)]/q_c(\text{Evol}_1)$.

B. Selection-I

As illustrated in Fig. 5, Evol_1 fails in some cases, but still outperforms RR on more than half of all networks with significant improvements. This indicates that q_c can benefit from the treatment of the conflict between it and F. Thus, another question is: do the perturbations of F in other groups influence q_c? To study this, we add a selection probability $p(S^{\epsilon}_{\text{new}})$ in all the groups except for the one that the critical node belongs to

$$p(S^{\epsilon}_{\text{new}}) = \frac{\sum_{i=t+1}^{t+\delta t} G_{\epsilon}(S^{\epsilon}_{\text{old}}; i)}{\sum_{i=t+1}^{t+\delta t} [G_{\epsilon}(S^{\epsilon}_{\text{new}}; i) + G_{\epsilon}(S^{\epsilon}_{\text{old}}; i)]}. \tag{16}$$

In other words, the algorithm would rather choose the smaller average giant fraction F with a higher probability than directly select it (Evol_2). An example and the corresponding performance are demonstrated in Figs. 6 and 7, respectively.

C. Local and Global Mutation

Furthermore, we introduce mutation operators to disorder F, including local and global mutation (Evol_3). Both of them, at each time, equally choose one from the following six mutation operators to produce the corresponding sequence.

1) The displacement mutation (DM) operator [53] usually randomly selects a fragment that would be moved from the sequence and eventually inserted in a random place.
2) The exchange mutation (EM) operator [54] aims at choosing two nodes in the sequence at random and then exchanging them (similar strategy could be found in [4]).
3) As for the insertion mutation operator [53], [55], one random node is moved out the sequence and placed at a random position afterward.

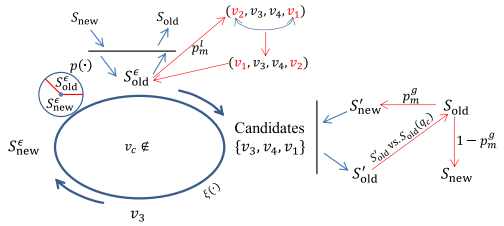


Fig. 8. Illustration of mutations. The local mutation happens with the probability p^l_m on the groups where the critical node is absent. For the entire sequence S_{old}, it may either transfer to S_{new} directly (with the probability $1 - p^g_m$), or mutates into S'_{new} (with the probability p^g_m). If mutation achieves, then the sequence with smaller q_c would be chosen between S_{old} and S'_{old} which is obtained based on S'_{new}.

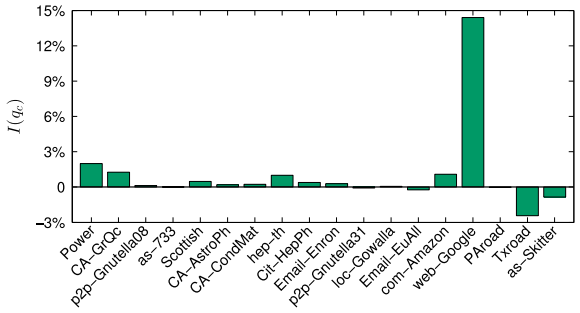


Fig. 9. Improvement percentage $I(q_c)$ of q_c for Evol_3 compared to Evol_2 on the 18 real-world networks, where Evol_3 is with $p^l_m = 0.1$ and $p^g_m = 0.3$, and $I(q_c) = [q_c(\text{Evol}_2) - q_c(\text{Evol}_3)]/q_c(\text{Evol}_2)$.

4) The simple inversion mutation (SIM) operator [56] selects randomly two cut points in the sequence, and then reverses the fragment between these two cut points.
5) On the basis of SIM, we slightly change it by narrowing the cut points (S-SIM), namely the random selection happens in a narrow range.
6) The inversion mutation [57] operator works similarly to the DM. It also randomly selects a fragment, removes it from the sequence and then inserts it in a randomly selected position, however, in the reversed order.

We show a simple illustration of the two types of mutations in Fig. 8 and the performance of Evol_3 in Fig. 9, where the local mutation probability p^l_m and the global mutation probability p^g_m are fixed to 0.1 and 0.3, respectively.

D. Selection-II

The results of Evol_2 and Evol_3 (see Figs. 7 and 9) manifest that: on the one hand, the perturbations of F in other groups truly affect the threshold q_c; on the other hand, large disturbance (mutation) may slow down the convergence rate of the algorithm (see the results in the Txroad and as-Skitter networks). Besides, when investigating $G(q)$ of q under different $\hat{\tau}$ (Fig. 10), we find that a small q_c always corresponds to a large $G(q)$ at the early stage of removal, i.e., when q is small. This indicates that a better q_c may be acquired by increasing $G(q)$ in some places. Therefore, we finally make the selection tend to a larger average giant fraction and

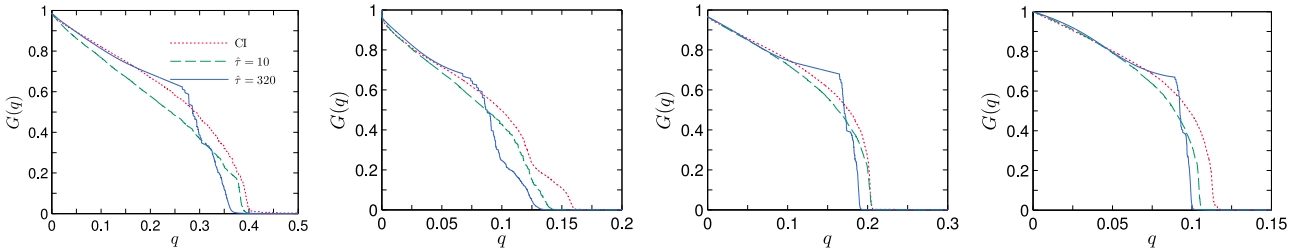


Fig. 10. Giant fraction $G(q)$ versus the removed fraction of nodes q for CI and RR (under sum rule with $\hat{\tau} = 10$ and $\hat{\tau} = 320$ respectively) on (a) Cit-HepPh network, (b) loc-Gowalla network, (c) ER network with $n = 10^5$ and $\langle k \rangle = 3.5$, and (d) SF network with $\gamma = 3.0$, $n = 10^5$, and $\langle k \rangle = 4.0$.

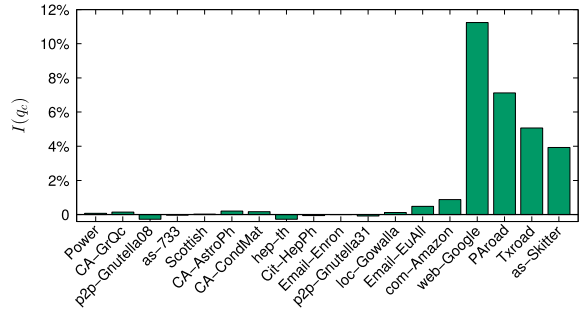


Fig. 11. Improvement percentage $I(q_c)$ of q_c for Evol_4 compared to Evol_3 on 18 real-world networks, where $I(q_c) = [q_c(\text{Evol}_3) - q_c(\text{Evol}_4)]/q_c(\text{Evol}_3)$.

rewrite (16) as (Evol_4)

$$p(S^{\epsilon}_{\text{new}}) = \frac{\sum_{i=t+1}^{t+\delta t} G_{\epsilon}(S^{\epsilon}_{\text{new}}; i)}{\sum_{i=t+1}^{t+\delta t} [G_{\epsilon}(S^{\epsilon}_{\text{new}}; i) + G_{\epsilon}(S^{\epsilon}_{\text{old}}; i)]}. \tag{17}$$

The corresponding results are shown in Fig. 11.

E. Evol_q, Evol_F, Reinitialization of $S_a^{\epsilon 0}$, and Summary

For a convenient description, we use Evol_q and Evol_F to be associated with the optimization of the immunized threshold q_c and the average giant fraction F, respectively. They are

$$\text{Evol}_q := \text{Evol}_4(\text{to stabilize}) + \text{Evol}_1(\text{to stabilize})$$

where $\text{Evol}_4 + \text{Evol}_1$ means we further use Evol_1 to optimize F of other groups that the critical node does not belong to, after the optimization of q_c under Evol_4

$$\text{Evol}_F := \text{Evol}_1 \text{but with goal function}$$
$$f(\text{goal}) = \sum_{i=t+1}^{t+\delta t} G_{\epsilon}(S^{\epsilon}_a; i), \forall t. \tag{18}$$

The corresponding algorithms of Evol_q are detailed in Algorithms 4 and 5. Note that it is also possible to obtain a better F through $\text{Evol}_4(\text{to stabilize}) + \text{Evol}_F(\text{to stabilize})$.

Moreover, another strategy is to use RR (Algorithm 2) to quickly reinitialize the input $S_a^{\epsilon 0}$, and then based on it to further optimize q_c or F. In detail, we employ RR to independently evolve $S_a^{\epsilon 0}$ a number of times (population size, e.g., 100 in this paper), and then choose the one with smallest F as the input of Evol_q or Evol_F. Note here $\delta_r = 0.1$, $\delta_\tau = 0.05$, $\tau_s = 5$, and $\hat{T} = 200$ are conducted for the purpose of quick computation, and this procedure is parallel processed too. Besides, in order to differentiate Evol_q and Evol_F, we mark them with I, namely Evol_q^I and Evol_F^I correspond to Evol_q and Evol_F but with reinitialization of $S_a^{\epsilon 0}$, respectively.

With respect to the computational complexity of Evol_q, it is very hard to analyze it in detail. Actually, the UF (Algorithm 1) can achieve its function within near-constant time. That means that the time complexity of Evol_q is approximately $O(\tau^2 T T_g \langle k \rangle n)$ (ignore the mutation operators) for ER-type networks, since the computational complexity of searching the neighbors of a certain node is $O(\langle k \rangle)$. However, in SF-type networks (most of real-world networks), the heterogeneity of degree makes this approximation invalid. In other words, there are a few nodes whose degrees are proportional to $n^{(1/[\gamma-1])}$ [1]. This problem also exists in CI and EI [see (8)

Algorithm 4 RR Method-Update

function RR-U($S^{\epsilon}_{\text{old}}$, t, δt, n, $\hat{r}$, $\hat{\tau}$, p_m^I, $\hat{T}$)
  $T \leftarrow 1$
  $\hat{t} \leftarrow \min(t + \delta t, n)$
  $S^p_{\text{old}} \leftarrow S^{\epsilon}_{\text{old}}(i), \forall i \in [t+1, \hat{t}]$ // S^p_{old} is a fragment of $S^{\epsilon}_{\text{old}}$
  $S^p_{\text{new}} \leftarrow S^p_{\text{old}}$
  while $T \le \hat{T}$ **do**
    $j \leftarrow 1$
    $r \in (0, \hat{r}]$, $\tau \in [1, \hat{\tau}]$ // r and τ are random numbers
    if $v_c \notin S^p_{\text{new}}$ **then**
      Mutate S^p_{new} with probability p_m^I
    end if
    while $j \le \hat{t}$ **do**
      $\mathcal{N}^{\text{TC}}(t+j) \leftarrow$ based on r and τ
      $v_s \leftarrow \arg\min_{v_\hbar} \xi(v_\hbar), v_\hbar \in \mathcal{N}^{\text{TC}}(t+j)$ // UF algorithm
      $S^p_{\text{new}} \leftarrow$ update itself
      $j \leftarrow j + 1$
    end while
    if $v_c \in S^p_{\text{new}}$ **then**
      if $q_c(S^p_{\text{old}}) < q_c(S^p_{\text{new}})$ **then**
        $S^p_{\text{new}} \leftarrow S^p_{\text{old}}$
      else
        $S^p_{\text{old}} \leftarrow S^p_{\text{new}}$
      end if
    else
      if $p(S^{\epsilon}_{\text{new}})$ **then** // according to Eq. (17)
        $S^p_{\text{old}} \leftarrow S^p_{\text{new}}$
      else
        $S^p_{\text{new}} \leftarrow S^p_{\text{old}}$
      end if
    end if
    $T \leftarrow T + 1$
  end while
  $S^{\epsilon}_{\text{old}}(i), \forall i \in [t+1, \hat{t}] \leftarrow S^p_{\text{old}}$
  return $S^{\epsilon}_{\text{old}}$
end function

TABLE II
IMMUNIZATION THRESHOLD q_c OF HD, HAD, CI, EI, EVOL$_q$, EVOL$_F$, AND EVOL$_q^I$ FOR THE 18 REAL-WORLD NETWORKS. ℓ OF CI IS FIXED TO 4 EXCEPT FOR THE EMAIL-EUALL NETWORK (WITH $\ell = 3$) AND THE AS-SKITTER NETWORK (WITH $\ell = 2$). THE BOLD NUMBERS ARE THE MINIMAL q_c AMONG THESE STRATEGIES FOR A SAME NETWORK

Network	HD	HAD	CI	EI	Evol$_q$	Evol$_F$	Evol$_q^I$
Power	1.9732E-1‡‡	1.5421E-1‡‡	1.1536E-1‡‡	6.8225E-2(4.6384E-4)‡†	**5.2934E-2(6.6145E-4)**	5.7407E-2(9.7011E-4)	5.3309E-2(8.8682E-4)
CA-GrQc	1.7397E-1‡‡	1.5642E-1‡‡	3.3574E-1‡‡	8.1695E-2(7.1612E-4)‡	**6.6072E-2(4.6840E-4)**	6.9973E-2(6.6832E-4)	6.6386E-2(5.5011E-4)
p2p-Gnutella08	3.2455E-1‡‡	2.5138E-1‡	2.2916E-1†	2.3947E-1(2.8279E-3)‡	**2.0901E-1(5.3262E-4)**	2.2321E-1(1.2330E-3)	2.0930E-1(6.1417E-4)
as-733	3.7534E-2‡‡	3.8307E-2‡‡	2.9657E-2‡†	2.6158E-2(2.3652E-4)‡	**2.3324E-2(0.0000E-0)**	2.3609E-2(1.6064E-4)	2.3409E-2(9.3420E-5)
Scottish	1.2133E-1‡‡	8.3425E-2‡†	2.8168E-1‡‡	6.5170E-2(5.7915E-4)†	**5.8854E-2(2.0273E-4)**	6.2078E-2(4.7312E-4)	5.9144E-2(2.8916E-4)
CA-AstroPh	4.5517E-1‡‡	3.3423E-1‡‡	2.5917E-1‡†	2.3017E-1(1.0847E-3)‡	**2.0651E-1(5.8575E-4)**	2.1915E-1(1.5347E-3)	2.0683E-1(7.0643E-4)
CA-CondMat	2.4752E-1‡‡	1.9452E-1‡‡	1.3906E-1‡†	1.1675E-1(5.4347E-4)†	**1.0525E-1(3.3510E-4)**	1.1076E-1(6.9904E-4)	1.0530E-1(4.1045E-4)
hep-th	6.6435E-1‡‡	4.4390E-1‡†	4.1057E-1‡	4.0392E-1(1.1668E-3)‡	3.5474E-1(2.6301E-3)	3.7681E-1(1.6501E-3)	**3.4941E-1(1.6354E-3)**
Cit-HepPh	6.5226E-1‡‡	4.4280E-1‡	4.1000E-1†	4.1969E-1(1.0695E-3)‡	3.7226E-1(2.7828E-3)	3.8978E-1(2.0858E-3)	**3.6734E-1(2.1711E-3)**
Email-Enron	1.1165E-1‡‡	1.1615E-1‡‡	8.3778E-2‡	7.5339E-2(3.7045E-4)‡	**6.7526E-2(3.0542E-4)**	7.0154E-2(4.9555E-4)	6.7550E-2(2.8754E-4)
p2p-Gnutella31	2.2546E-1‡‡	1.9851E-1‡†	1.7567E-1‡	1.6181E-1(1.9531E-4)	**1.5480E-1(3.6307E-4)**	1.6508E-1(1.7062E-3)	1.5521E-1(4.4571E-4)
loc-Gowalla	2.7380E-1‡‡	2.0432E-1‡‡	1.5965E-1‡†	1.3691E-1(3.8989E-4)†	1.2558E-1(8.1679E-4)	1.3027E-1(1.0315E-3)	**1.2352E-1(7.9836E-4)**
Email-EuAll	5.3956E-3‡†	4.8338E-3‡	4.4982E-3‡	2.6340E-2(1.5483E-3)‡‡	**3.8940E-3(9.2257E-6)**	3.9892E-3(2.3371E-5)	3.9256E-3(1.4333E-5)
com-Amazon	2.3385E-1‡‡	2.0464E-1‡‡	1.2574E-1‡‡	8.2036E-2(1.5478E-4)†	7.6161E-2(7.3255E-4)	8.1055E-2(8.0609E-4)	**7.5734E-2(5.5457E-4)**
web-Google	2.8902E-1‡‡	1.9589E-1‡‡	9.4237E-2‡‡	4.7902E-2(2.9879E-4)‡‡	**2.4833E-2(1.8758E-3)**	4.0611E-2(1.6168E-3)	2.4847E-2(1.9779E-3)
PAroad	2.5172E-1‡‡	2.2633E-1‡‡	6.5374E-2‡‡	1.5811E-2(4.0486E-4)‡‡	8.8932E-3(3.6267E-4)	1.1085E-2(3.0214E-4)	**8.0742E-3(3.7822E-4)**
Txroad	2.2277E-1‡‡	2.3261E-1‡‡	5.9963E-2‡‡	1.2174E-2(2.4996E-4)‡‡	6.3769E-3(3.3293E-4)	7.9650E-3(3.3975E-4)	**5.6730E-3(3.1205E-4)**
as-Skitter	1.8988E-1‡‡	1.1888E-1‡‡	8.9509E-2‡‡	4.1794E-2(1.9965E-4)‡†	**3.1546E-2(6.6333E-4)**	3.6313E-2(3.7537E-4)	3.2100E-2(4.8640E-4)

$\dagger$, $\ddagger$, $\ddagger\dagger$ and $\ddagger\ddagger$ indicate Evol$_q$ has an improvement of over 5%, 10%, 20% and 30% compared to the corresponding strategies, respectively.

Algorithm 5 Evol$_q$ Method

Input: *Network*, $\hat{r}$, $\hat{\tau}$, $\hat{T}$, $\hat{\delta t}$, $S_a^{\epsilon 0}$, $\hat{T}_g$, p_m^l, p_m^g
Output: $S_a^{\epsilon \hat{T}_g}$

$T_g \leftarrow 1$
$S_a^{\epsilon T_g} \leftarrow S_a^{\epsilon 0}$
$S_b^{\epsilon T_g} \leftarrow S_a^{\epsilon T_g}$
while $T_g \leq \hat{T}_g$ **do**
  $t \leftarrow 0$
  Mutate $S_a^{\epsilon T_g}$ with probability p_m^g
  $\delta t \in [1, \hat{\delta t}]$ // δt is a random integer
  $\mathcal{N}_\epsilon(S_a^\epsilon; \delta t) \backslash \mathcal{N}_\epsilon(S_a^\epsilon; 0)$, $\mathcal{N}_\epsilon(S_a^\epsilon; 2\delta t) \backslash \mathcal{N}_\epsilon(S_a^\epsilon; \delta t)$, ..., $\mathcal{N}_\epsilon(S_a^\epsilon; n) \backslash \mathcal{N}_\epsilon(S_a^\epsilon; (\lfloor \frac{n}{\delta t} \rfloor - 1)\delta t)$ // divide $\mathcal{N}$ into several group based on $S_a^{\epsilon T_g}$ and δt
  Initialize $\mathcal{N}_p(v_i)$ and $\mathcal{G}(v_i)$ for each group according to $S_a^{\epsilon T_g}$
  while $t < n$ **do** // parallel
    $S_a^{\epsilon T_g} \leftarrow$ RR-U(S_{old}^ϵ, t, δt, n, $\hat{r}$, $\hat{\tau}$, p_m^l, $\hat{T}$) // call Algorithm 4
    $t \leftarrow t + \delta t$
  end while
  if $q_c(S_b^{\epsilon T_g}) < q_c(S_a^{\epsilon T_g})$ **then**
    $q_c(S_a^{\epsilon T_g}) \leftarrow q_c(S_b^{\epsilon T_g})$
  else
    $q_c(S_b^{\epsilon T_g}) \leftarrow q_c(S_a^{\epsilon T_g})$
  end if
  $T_g \leftarrow T_g + 1$
end while
$S_a^{\epsilon T_g} \leftarrow S_a^{\epsilon T_g}$ // call Algorithm 3 (Evol$_1$) to fix F
$S_a^{\epsilon \hat{T}_g} \leftarrow S_a^{\epsilon T_g}$

and (9)]. On the whole, compared to CI and EI, Evol$_q$ can obtain a better outcome (where T_g is small) with less time in most networks, especially in large networks, e.g., a better result can be obtained within minutes on the as-Skitter network where CI might take more than one week if $\ell = 3$ in the same simulation environment. Besides this, the results would become better and better with increasing T_g.

V. EXPERIMENTS AND RESULTS

To validate the effectiveness of the proposed method, we conduct adequate experiments on Lenovo NeXtScale nx360M5, Xeon E5-2667v3 8C 3.2 GHz, Infiniband FDR14 with eight threads, considering both paradigmatic model networks (including ER [52] and SF [43], [58] networks) and several real-world networks (Table I). In this section, if there is no special explanation, ℓ of CI is fixed to 4, each result of EI is obtained with $K = 6$ and 2000 candidates, the parameters of RR and PR are same as [34], and our methods are based on HD with $\hat{\tau} = 50$, $\hat{r} = 1.0$, $\hat{T} = 20$, $\hat{\delta t} = \lfloor 0.1 \times n \rfloor$, $p_m^l = 0.1$, and $p_m^g = 0.3$, respectively.

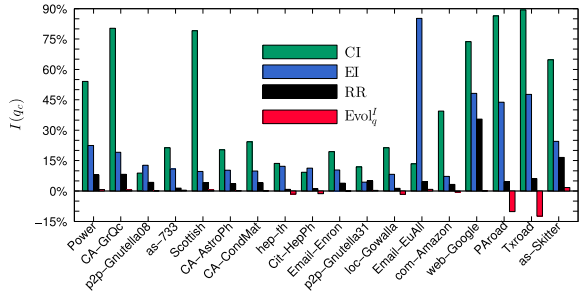


Fig. 12. Improvement percentage $I(q_c) = [q_c(S_a) - q_c(\text{Evol}_q)]/q_c(S_a)$ of q_c for Evol$_q$ compared to CI (green), EI (blue), RR (black), and Evol$_q^I$ (red) on the 18 real-world networks, where S_a accordingly corresponds to CI, EI, RR, and Evol$_q^I$. Besides, EI, RR, Evol$_q^I$, and Evol$_q$ are simulated through 20 independent realizations, respectively.

A. Percolation Metric

The percolation metric includes the immunization threshold q_c [here assume $\theta = 0.01$, i.e., $G(S_a; q_c) \leq 0.01$] and the average giant fraction F [4], [14], [15], [23], [39]. First, we consider the immunization threshold q_c on real-world networks. The related results are shown in Table II where mean and standard deviation values of EI, Evol$_q$, Evol$_F$, and Evol$_q^I$ are given. For almost all of the networks tested here, Evol$_q$ performs significantly better than HD and HAD (over 30%). Further regarding to CI and EI (Fig. 12), Evol$_q$ also has a large improvement of over 20% and 10% in most networks, accordingly. In particular, on the four largest

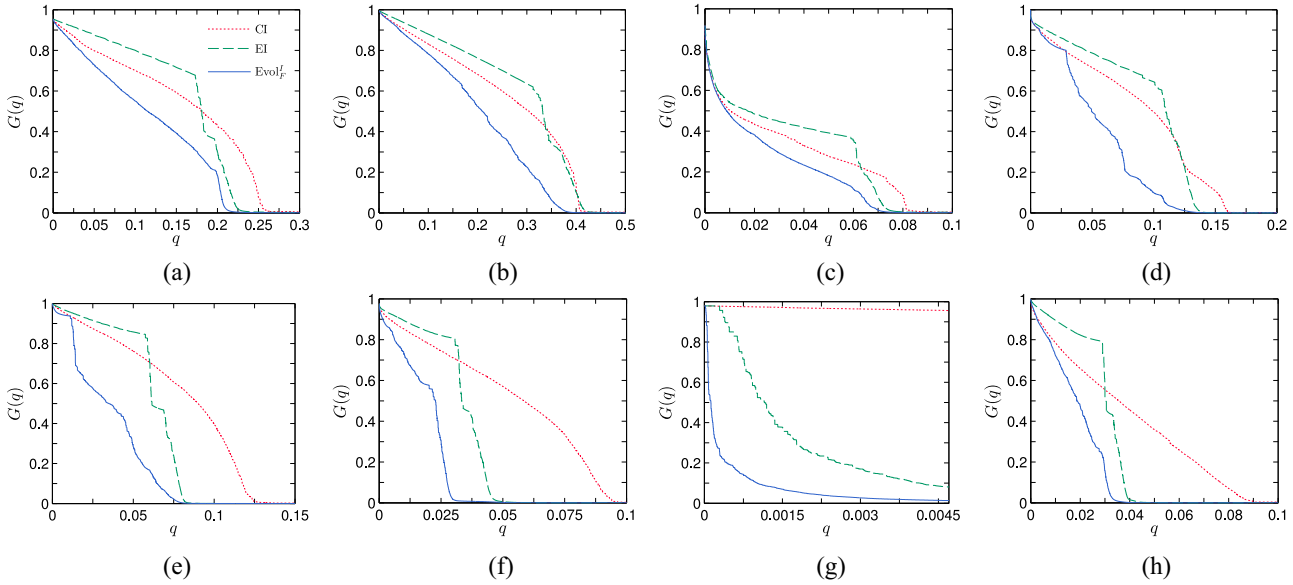


Fig. 13. Giant fraction $G(q)$ versus the removed fraction of nodes q for CI, EI, and Evol$_F^I$ respectively on (a) CA-AstroPh network, (b) Cit-HepPh network, (c) Email-Enron network, (d) loc-Gowalla network, (e) com-Amazon network, (f) Web-Google network, (g) Txroad network, and (h) as-Skitter network.

networks (the Web-Google, PAroad, Txroad, and as-Skitter), Evol$_q$ decreases the corresponding immunized threshold by over 80%, 70%, 60%, and 20% against HD, HAD, CI, and EI, respectively. Besides, Evol$_F$ also obtains much smaller q_c than other methods, even though it is designed for the optimization of the average giant fraction. Moreover, reinitialization of $S_a^{\epsilon 0}$ strategy (Evol$_q^I$) performs extremely better than all of the other methods in the two road networks, but it only acquires smaller q_c in 6/18 networks compared to Evol$_q$. This may further demonstrate the existence of conflict between smaller F and optimal q_c. By contrast, the reinitialization of $S_a^{\epsilon 0}$ strategy promotes the average giant fraction F to an extremely optimal level (see Table III), e.g., improvement of 5.49%–98.60% (average 48.85%) versus CI, 12.32%–75.27% (average 35.60%) against EI and 1.78%–26.88% (average 9.28%) compared to RR. In addition, the result on the Web-Google network indicate that a better F may be obtained through Evol$_q$ + Evol$_F$ in some cases, i.e., first optimize q_c and then based on it to optimize F.

To further estimate the proposed methods, the giant fraction $G(q)$ (susceptible fraction) of removed nodes q is plotted in Fig. 13. The experimental networks include the following.

1) *CA-AstroPh:* A collaboration network regarding Astro Physics in Arxiv [46]. In this network, some scientists (nodes) may be important for the transmission of new methods or models.
2) *Cit-HepPh:* A citation network of Arxiv high energy physics phenomenology (HepPh) [45].
3) *Email-Enron:* The Email-Enron network is a network in which a node represents an email address and an edge denotes that there is at least one mail transferred between the two addresses [44]. Some viruses may spread relying on it.
4) *loc-Gowalla:* A location-based social network.
5) *com-Amazon:* A customers who bought this item also bought feature network collected by crawling Amazon website [51]. In this network, each node corresponds to a product and edges represent that two products are co-purchased together frequently. Thus, it is quite important to find out the most influential products.
6) *Web-Google:* A network of Google Web where nodes represent Web pages and edges indicate hyperlinks among them [44].
7) *Txroad:* A complex road network of Texas [44]. Node is either an intersection or an endpoint, and an edge represents the road among them. In this case, the influential part of nodes may play an important role in monitoring the status of the whole road network.
8) *as-Skitter:* An Internet topology network on autonomous systems by Skitter [45].

The choices of these networks consider both density of edges [9] and assortativity of degrees [10], which are always associated with the robustness of a network. As we can see from Fig. 13, Evol$_F^I$ displays huge superiority that less nodes are immunized (q) for the same extent of immunization ($G(q)$) than both CI and EI in all networks studied here.

Moreover, we perform experiments on model networks, including ER and SF types. Different from real-world networks (the existence of community structure makes some low-degree nodes quite important, the removal of which fragments the network into several independent communities), nodes with higher degree are always more influential in these model networks (Fig. 14). Therefore, a larger $\hat{\tau}$ of Evol$_q$ is used to try to maintain the character of HD. In the meantime, an adaptive $\hat{\delta t}$ is employed to further deal with the conflict between F and q_c. Additionally, the feedback vertex set (FVS) strategy [34] is also conducted here for Evol$_q$ in model networks. The results of $G(q)$ versus q are shown in Fig. 15, where Evol$_q$ outperforms (with smaller q_c) CI and EI on both kinds of networks. With respect to average giant fraction, Evol$_F^I$ can obtain a much smaller F compared to other methods. Further, we exhibit the immunized threshold

TABLE III
AVERAGE GIANT FRACTION F OF HD, HAD, CI, EI, EVOL$_q$, EVOL$_F$, AND EVOL$_F^I$ FOR THE 18 REAL-WORLD NETWORKS. THE BOLD NUMBERS ARE THE MINIMAL F OF THESE STRATEGIES FOR A SAME NETWORK

Network	HD	HAD	CI	EI	Evol$_q$	Evol$_F$	Evol$_F^I$
Power	6.3642E-2‡‡	5.2384E-2‡‡	4.4900E-2‡‡	1.1195E-2(9.4370E-4)‡‡	7.7879E-3(2.5543E-4)	7.5119E-3(2.5272E-4)	**7.0143E-3(1.1146E-4)**
CA-GrQc	8.2540E-2‡‡	6.8512E-2‡‡	5.2699E-2‡‡	3.4655E-2(4.5158E-4)‡	2.8759E-2(3.7922E-4)	2.8584E-2(3.1915E-4)	**2.7483E-2(1.4384E-4)**
p2p-Gnutella08	1.9926E-1‡‡	1.5735E-1‡	1.4149E-1	1.6511E-1(1.2970E-3)‡	1.3629E-1(6.5712E-4)	1.3628E-1(6.0116E-4)	**1.3274E-1(2.9792E-4)**
as-733	1.2752E-2‡‡	1.2484E-2‡‡	1.5000E-2‡‡	9.7419E-3(3.7297E-5)‡	9.0656E-3(1.4590E-4)	8.6605E-3(7.5117E-5)	**8.5420E-3(7.6173E-5)**
Scottish	3.2067E-2‡†	2.7196E-2‡	5.4200E-2‡‡	2.5926E-2(1.4685E-4)‡	2.3008E-2(2.5897E-4)	2.2906E-2(1.4263E-4)	**2.2559E-2(5.9348E-5)**
CA-AstroPh	2.5084E-1‡‡	2.0843E-1‡‡	1.5620E-1‡	1.5788E-1(1.0961E-3)‡	1.2725E-1(2.2500E-3)	1.2854E-1(2.9715E-3)	**1.1302E-1(5.1593E-4)**
CA-CondMat	1.2377E-1‡‡	1.1032E-1‡‡	8.3200E-2‡†	7.7388E-2(3.7784E-4)‡	6.5880E-2(8.7216E-4)	6.5741E-2(1.2293E-3)	**5.9906E-2(2.2204E-4)**
hep-th	3.7507E-1‡‡	3.0476E-1‡†	2.5410E-1	2.7417E-1(1.7074E-3)‡	2.2044E-1(5.9438E-3)	2.4341E-1(3.8714E-3)	**1.8358E-1(1.2201E-3)**
Cit-HepPh	3.6645E-1‡‡	3.0620E-1‡	2.6450E-1†	2.8596E-1(2.9827E-3)‡	2.3654E-1(3.6343E-3)	2.5002E-1(5.0428E-3)	**2.0109E-1(1.4938E-3)**
Email-Enron	3.9252E-2‡‡	3.7965E-2‡‡	2.9200E-2‡†	3.1358E-2(3.4463E-4)‡‡	2.3084E-2(3.8399E-4)	2.1892E-2(3.0673E-4)	**2.0564E-2(9.8980E-5)**
p2p-Gnutella31	1.2866E-1‡†	1.1428E-1‡	1.0150E-1	1.1720E-1(2.0192E-4)‡	9.9900E-2(3.8198E-4)	9.9386E-2(6.6042E-4)	**9.5924E-2(2.8491E-4)**
loc-Gowalla	1.3287E-1‡‡	1.1416E-1‡‡	8.6800E-2‡‡	9.1550E-2(1.9762E-3)‡‡	6.6758E-2(2.4966E-3)	5.8752E-2(1.4089E-3)	**5.3048E-2(9.7933E-4)**
Email-EuAll	8.9831E-4‡	8.5424E-4†	5.6000E-3‡‡	1.8752E-3(4.9563E-5)‡‡	8.2293E-4(6.1861E-6)	8.0425E-4(1.7611E-6)	**8.0209E-4(6.3785E-7)**
com-Amazon	1.2244E-1‡‡	1.1835E-1‡‡	7.9300E-2‡‡	6.1880E-2(6.7827E-4)‡†	4.8843E-2(2.1443E-3)	4.4550E-2(2.2001E-3)	**3.5156E-2(7.1859E-4)**
web-Google	1.1408E-1‡‡	8.8641E-2‡‡	5.2600E-2‡‡	3.2239E-2(4.6141E-4)‡†	**1.6641E-2(1.0884E-3)**	2.3464E-2(8.6606E-4)	1.7796E-2(4.6287E-4)
PAroad	1.1478E-1‡‡	7.1470E-2‡‡	4.1700E-2‡‡	3.4363E-3(3.9929E-4)‡‡	1.3561E-3(1.2141E-4)	1.4127E-3(7.0651E-5)	**9.6908E-4(3.4416E-5)**
Txroad	8.5943E-2‡‡	6.5240E-2‡‡	3.4200E-2‡‡	1.9300E-3(2.8858E-4)‡‡	6.9895E-4(5.3773E-5)	7.0785E-4(3.6916E-5)	**4.7734E-4(2.3365E-5)**
as-Skitter	7.0579E-2‡‡	4.8703E-2‡‡	3.9399E-2‡‡	2.8691E-2(2.2245E-4)‡†	2.2214E-2(9.0711E-4)	2.0447E-2(4.5467E-4)	**1.8006E-2(1.4843E-4)**

†, ‡, ‡† and ‡‡ indicate Evol$_F$ has an improvement of over 5%, 10%, 20% and 30% compared to the corresponding strategies, respectively.

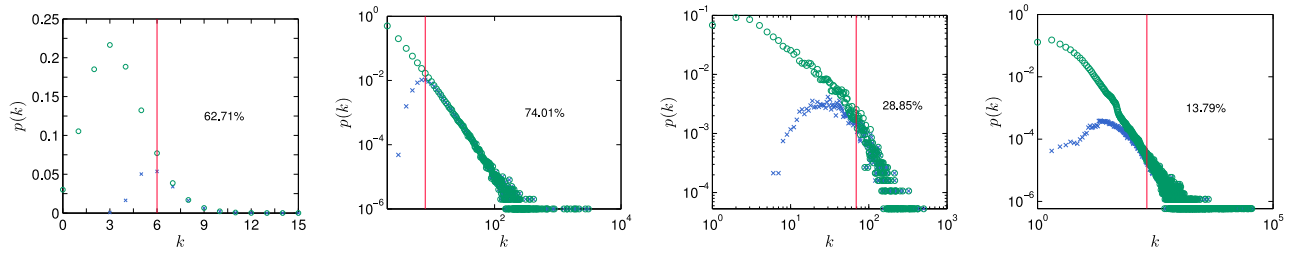


Fig. 14. Node-degree distribution of the entire network (green circle) and the nodes in $\mathcal{N}\backslash\mathcal{N}_\epsilon(S_a; nq_c)$ (blue x-mark). The corresponding method S_a is either PR for (a) ER networks with $n = 10^6$ and $\langle k\rangle = 3.5$, and (b) SF networks with $\gamma = 3.0$, $n = 10^6$, and $\langle k\rangle = 4.0$ or Evol$_q$ for (c) CA-AstroPh network and (d) as-Skitter network. The red line divides each figure into two parts: on the right $p_d \geq 90\%$, where $p_d = n^1_{\geq}(k)/n^2_{\geq}(k)$ is the ratio of $n_1(k)$ and $n_2(k)$, vice versa. $n^1_{\geq}(k)$ and $n^2_{\geq}(k)$ denote the number of nodes with a degree equal or larger than k in $\mathcal{N}\backslash\mathcal{N}_\epsilon(S_a; nq_c)$ and $\mathcal{N}$, respectively. The percentage on the middle of each plot is the percentage of $n^1_{\geq}(k_c)$ and nq_c in which k_c is associated with the degree k where the red line is.

q_c of the average degree $\langle k\rangle$ and the exponential parameter γ [58] in Fig. 16, which shows that Evol$_q$ is still better than other methods. Note that in this paper the SF network is constructed using the Barabási–Albert model [43] when $\gamma = 3.0$, otherwise, by the model in [58].

B. SIR Model Simulation

For a network under the simulation of the SIR epidemic spreading model [35], each node is either in the susceptible (S), infected (I), or recovered (R) state. At the earliest stage, all nodes are initialized to be susceptible, and then the immunization nodes are put on the recovered state (or they are removed from the network, including the incidental edges). Next, we randomly select one node from the susceptible node set to be infected to investigate the transmissibility of this node on the remainder network. The infected nodes, at each time step, infect its susceptible neighbors with the infection rate λ, and then they recover with the rate η. The recovered nodes would not be affected anymore. This process is repeated until there is no infected node in the remaining network.

The simulated results are shown in Figs. 17 and 18 compared to CI and EI on the Email-Enron network, the loc-Gowalla network, the Web-Google network, and the as-Skitter network. Note that EI^b and Evol$_q^b$ are the sequences with minimal q_c of 20 realizations for EI and Evol$_q$, respectively, and Evol$_F^{Ib}$ with minimal F. In each simulation, λ and η are fixed to 0.2 and 0.05, accordingly. For all the networks studied here (Fig. 17), Evol$_q^b$ has a significantly smaller number of recovered individuals (were infected) than EI does under the same immunization fraction q (16–96 times smaller at the end of the infection), especially when the network becomes large.

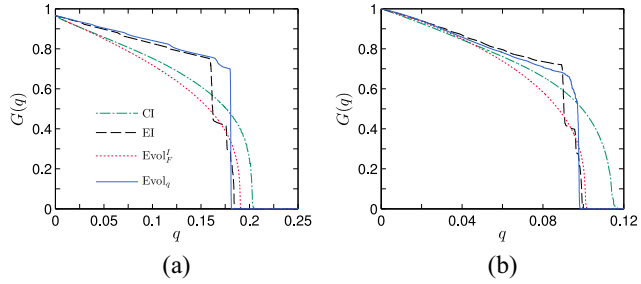


Fig. 15. Fraction $G(q)$ of the size of the giant connected component versus the fraction of immunization nodes q (over 50 implementations) for CI, EI, Evol$_F^I$, and Evol$_q$ for (a) ER networks with $n = 10^6$ and $\langle k\rangle = 3.5$, and (b) SF networks with $\gamma = 3.0$, $n = 10^6$, and $\langle k\rangle = 4.0$. Here, Evol$_q$ optimizes q_c by aid of the optimization of FVS with $\hat{\tau} = 200$, $\hat{T} = 10$, and $\hat{\delta t} = \lfloor F(\text{HD}) \times n\rfloor$.

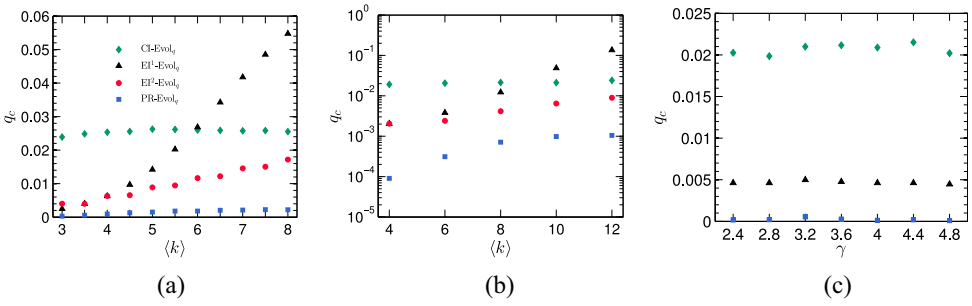


Fig. 16. Immunization threshold q_c of the average degree $\langle k \rangle$ and the exponent γ for CI, EI^1, EI^2, PR, and Evol$_q$ for (a) ER networks with $n = 10^5$, (b) SF networks with $\gamma = 3.0$ and $n = 10^5$, and (c) SF networks with $\langle k \rangle = 4.0$ and $n = 10^5$. CI-Evol$_q$ is the difference of the immunization threshold of PR and the immunization threshold of CI, and also for EI^1-Evol$_q$, EI^2-Evol$_q$, and PR-Evol$_q$. Evol$_q$ is with $\hat{\tau} = 200$, $\hat{T} = 10$, and $\hat{\delta t} = \lfloor F(\text{HD}) \times n \rfloor$ through optimization of FVS. The results of EI^1 are obtained with $K = 6$ and EI^2 is controlled by $K = \langle k \rangle + 2$, respectively.

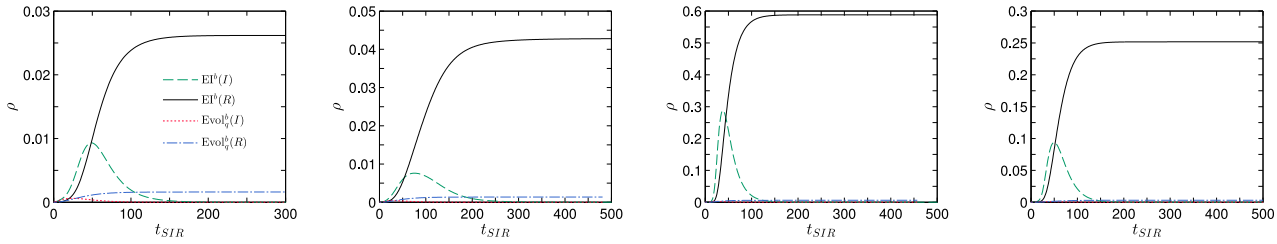


Fig. 17. Rate ρ of infected (I) and recovered (R) individuals versus the spreading time t_{SIR} under the immunization fraction $q = q_c(\text{Evol}_q^b)$ for EI^b and Evol$_q^b$, respectively, on (a) Email-Enron network, (b) loc-Gowalla network, (c) Web-Google network, and (d) as-Skitter network. In each network for each method, 5×10^5 selections are conducted.

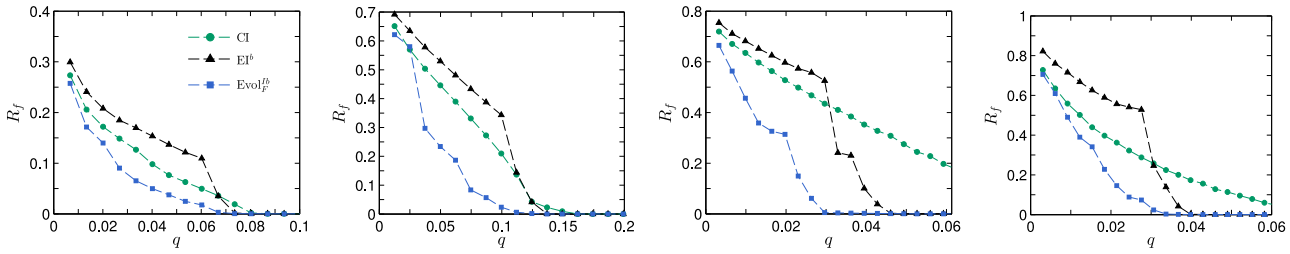


Fig. 18. Final recovered fraction R_f versus the fraction of immunization nodes q for CI, EI^b, and Evol$_F^{Ib}$, respectively, on (a) Email-Enron network, (b) loc-Gowalla network, (c) Web-Google network, and (d) as-Skitter network. Each data point is obtained by 10^4 independent simulations.

Considering the final recovered fraction R_f (Fig. 18), Evol$_F^I$ also outperforms CI and EI in almost all situations.

Besides this, we further consider the global airline network[3] [23], [39] which plays a vital role for the spread of a contagious disease, e.g., SARS or H1N1 [3]. Similar to the other networks, the weighted edges in this network are also simply replaced with the unweighted edges and all of the directed edges and the nodes except for those belonging to the largest connected component are removed, and as a result the experimental network composes of $n = 3146$ nodes (airports) and $m = 18\,146$ edges (airlines). We immunize the top-5% nodes of HD, HB, and Evol$_q$ by a lower infection rate $\lambda' = 0.0001$ (keep the network connected), and then respectively estimate the effectiveness of them using the SIR model with $\lambda = 0.2$ and $\eta = 0.05$ and a unique infected node chosen from the giant component after the immunization of Evol$_q$ (latitude 41.7994 and longitude 12.5949). The simulation results are shown in SFig. 1 where all of the spreads are almost stable at time step $t_{\text{SIR}} = 100$. Compared to HD and HB, Evol$_q$ is considerably better (only a small part of nodes are influenced after the immunization through Evol$_q$).

[3]The source of this data is from https://openflights.org/data.html.

VI. Discussion

A. Evolution Ability and Different Initial Methods

We compare the evolution ability of the proposed method with the method in [39] where an exchange strategy (ES) is developed to optimize the average giant fraction F. Analogous to the EM mutation operator, ES randomly selects two nodes in the sequence, and then exchange them if this operation can make a better (smaller) F. Due to the high time complexity of HAB, we estimate the performance of ES and Evol$_F$ on several small networks. The simulation results are

shown in STable I of the Supplementary, where ES is conducted with 10^6 exchanges for each network. Interestingly, Evol_F(HAB) has better results in all situations except the SF network even though HAB outperforms HD there. Therefore, we further investigate the impact of the initial sequence S_a on the proposed method. As reported in both STables I and II of the Supplementary, an improvement is obtained with a better initial sequence in most situations. Moreover, for all initial methods in all networks studied here, the proposed strategies have smaller q_c and F than both CI and EI, even when starting with a random sequence.

B. Crossover and Parameters

In this paper, neither Evol_q nor Evol_F considers crossover operators, since there are several approaches to maintain the diversity of the sequence (population), e.g., $\hat{\tau}$ and Selection-II. In other words, for the example of Evol_F^I (SFig. 2 in the Supplementary), a larger r may indicate a better $G(q)$ if a large enough fraction of nodes is removed (rapid increase of F_p as the rise of q). However, it cannot always achieve a better F which is usually obtained by a small τ (not too small). The different values of τ, to some degree, correspond to local and global optimization, respectively. Our simulation results manifest that an effective strategy of immunization is to combine them. But a large τ would reduce the efficiency of the algorithm, especially on SF-type networks. Hence, an acceptable way is to make τ and r be random numbers of $\hat{\tau}$ and $\hat{r}$, namely to some extend, a local optimization may be captured by a large τ and a small r, while avoiding by a small τ and a large r.

δt is related to the upper bound of the number of parallel threads. The simulation results (SFig. 3 in the Supplementary) show that, in most of the networks studied here (15/18), $I(q_c)$ is within 2%. This indicates that, on the one hand, better solutions can be obtained by an adjustment of δt. On the other hand, the algorithm may be faster by decreasing δt. Besides, the results of Evol_q under different goal θ are shown in SFig. 4 of the Supplementary. For most situation, Evol_q makes a smaller giant component ($G(q)$) with the same amount of immunized nodes (q) even compared to Evol_F^I. But sometimes [see SFig. 4(c) in the Supplementary], better q_c might be obtained through the optimization of Evol_F^I.

By the large, to effectively maintain the diversity of the sequence, the proposed methods generate offspring using random selection and order occupation considering both local and global optimization. There may be crossovers that could be conducted on the proposed strategy to further improve it since, e.g., a better initial sequence can achieve a better result.

C. Other Possible Applications

As demonstrated in Section V, an effective strategy is more likely to reveal the true robustness of a system that can be modeled as a network. Besides, for the nodes after the removal of which the network would drastically collapse, they always play fundamental roles to keep the system function and more protection should be given [12]. Thus, minimizing the number of that part nodes can save a lot of resource, like monitoring the traffic status of a road network.

1) When studying the robustness of a network, one of the goals is to optimize the network structure, e.g., to enhance the network robustness through the swap of edges but keeping the nodes' degree unchanged [4], [59]. However, an important factor which should be considered in this process is the time consumption because there might be more than million times of swaps to make the network meet our demand. This indicates that it is almost impossible to evolve a network based on the global properties of nodes or heuristic attack methods, like the Betweenness centrality or CI, especially for large networks. Thus, node degree as the most important and easily obtained property naturally becomes the touchstone for the evaluation of network status, even though Zhou and Liu [59] tried to find the Pareto front of different attack methods when enhancing the network robustness. As shown in SFig. 5 of the Supplementary, the network robustness $F(\cdot)$ increases with the rise of F(HD). But there is a critical point of F(HD) after which $F(\cdot)$ no longer increases. Besides, different methods are associated with different critical points but the proposed method makes the transition earlier than other strategies. In other words, a more effective method might correspond to a more reliable critical point.
2) There is evidence that the network assortativity can affect the network robustness [10] and it also plays an important role in a large range of domains related to network science [1], e.g., evolutionary games [60] and synchronization [61] as well as neuroscience [62], [63]. This means the network robustness varies with the change of the network assortativity [SFig. 5(c) and (d)]. Thus, what role does the network robustness truly play in those cases?
3) Optimization of FVS. Similar to [34], we further develop Evol_q to minimize FVS. But it can only obtain smaller FVS than PR in 8/18 networks. Therefore, how to effectively use Evol_q to solve the FVS problem is still an open question.

VII. Conclusion

In this paper, we have developed a framework of an evolutionary algorithm for finding the minimum separator of nodes to fragment a network, which corresponds to the solution to the robustness and immunization problem in network science. In detail, we have designed the goal function, added the mutation operators, proposed a special strategy for the selection and introduced a way for the initialization of population. To test the effectiveness of our method, we conduct extensive experiments on both real-world and synthetic networks using the percolation metric and SIR simulations. The results, especially of the large empirical networks, show that our proposed method considerably outperforms the other representative strategies. Moreover, a better solution might be further obtained with other fundamental approaches.

APPENDIX

Main Nomenclature

$\mathcal{N}$	Node set.
n	Network size.
c_i	Component i.
$\mathcal{G}_{c_i}$	Size of c_i.
$\mathcal{C}$	Component set.
$c_{\max}$	Giant component.
S_a	Arbitrary sequence of $\mathcal{N}$.
q	Proportion of removed nodes.
$G(S_a; q) \coloneqq$	$\mathcal{G}_{\max}/n$.
$q_c(S_a)$	Threshold related to S_a.
$F(S_a)$	Average fraction of giant components.
ϵ	Symbol related to percolation process.
t	Number of occupied nodes.
$\mathcal{N}_\epsilon$	Occupied node set.
ℓ	Undetermined parameter related to CI.
$\mathcal{G}(v_j)$	Component size associated with node v_j.
$\mathcal{C}(v_j)$	Component set related to node v_j.
$\mathcal{N}_p$	Parent set of $\mathcal{N}$ (UF).
τ	Times of selection.
r	Upper bound of possible candidates.
$\xi(\cdot)$	Rule function (RR and PR).
S_{old}	sequence for next-time evolution.
S_{old}	sequence obtained through evolving.
$\hat{r}, \hat{\tau}, \hat{T}, \hat{\delta t}, \hat{T}_g, p_m^l, p_m^g$	Parameters for Evol.
FVS	Feedback vertex set.

REFERENCES

[1] A.-L. Barabási *et al.*, *Network Science*. Cambridge, U.K.: Cambridge Univ. Press, 2016.

[2] P. Wang, M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding the spreading patterns of mobile phone viruses," *Science*, vol. 324, no. 5930, pp. 1071–1076, 2009.

[3] D. Brockmann and D. Helbing, "The hidden geometry of complex, network-driven contagion phenomena," *Science*, vol. 342, no. 6164, pp. 1337–1342, 2013.

[4] C. M. Schneider, A. A. Moreira, J. S. Andrade, S. Havlin, and H. J. Herrmann, "Mitigation of malicious attacks on networks," *Proc. Nat. Acad. Sci. USA*, vol. 108, no. 10, pp. 3838–3841, 2011.

[5] A. D. Kramer, J. E. Guillory, and J. T. Hancock, "Experimental evidence of massive-scale emotional contagion through social networks," *Proc. Nat. Acad. Sci. USA*, vol. 111, no. 24, pp. 8788–8790, 2014.

[6] D. Li *et al.*, "Percolation transition in dynamical traffic network with evolving critical bottlenecks," *Proc. Nat. Acad. Sci. USA*, vol. 112, no. 3, pp. 669–672, 2015.

[7] A. Szolnoki and M. Perc, "Collective influence in evolutionary social dilemmas," *EPL Europhys. Lett.*, vol. 113, no. 5, 2016, Art. no. 58004.

[8] Y. Chen, G. Paul, S. Havlin, F. Liljeros, and H. E. Stanley, "Finding a better immunization strategy," *Phys. Rev. Lett.*, vol. 101, no. 5, 2008, Art. no. 058701.

[9] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[10] M. E. Newman, "Assortative mixing in networks," *Phys. Rev. Lett.*, vol. 89, no. 20, 2002, Art. no. 208701.

[11] M. Perc, "Evolution of cooperation on scale-free networks subject to error and attack," *New J. Phys.*, vol. 11, no. 3, 2009, Art. no. 033027.

[12] L. Lü *et al.*, "Vital nodes identification in complex networks," *Phys. Rep.*, vol. 650, pp. 1–63, Sep. 2016.

[13] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[14] F. Morone and H. A. Makse, "Influence maximization in complex networks through optimal percolation," *Nature*, vol. 524, no. 7563, pp. 65–68, 2015.

[15] S. Mugisha and H.-J. Zhou, "Identifying optimal targets of network attack by belief propagation," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 94, no. 1, 2016, Art. no. 012305.

[16] R. Cohen, S. Havlin, and D. Ben-Avraham, "Efficient immunization strategies for computer networks and populations," *Phys. Rev. Lett.*, vol. 91, no. 24, 2003, Art. no. 247901.

[17] L. K. Gallos, F. Liljeros, P. Argyrakis, A. Bunde, and S. Havlin, "Improving immunization strategies," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 75, no. 4, 2007, Art. no. 045104.

[18] R. Pastor-Satorras and A. Vespignani, "Immunization of complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 65, no. 3, 2002, Art. no. 036104.

[19] R. Cohen, K. Erez, D. Ben-Avraham, and S. Havlin, "Breakdown of the Internet under intentional attack," *Phys. Rev. Lett.*, vol. 86, no. 16, pp. 3682–3685, 2001.

[20] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 65, no. 5, 2002, Art. no. 056109.

[21] C. M. Schneider, T. Mihaljev, and H. J. Herrmann, "Inverse targeting—An effective immunization strategy," *EPL Europhys. Lett.*, vol. 98, no. 4, 2012, Art. no. 46002.

[22] Y. Liu, B. Wei, Z. Wang, and Y. Deng, "Immunization strategy based on the critical node in percolation transition," *Phys. Lett. A*, vol. 379, nos. 43–44, pp. 2795–2801, 2015.

[23] P. Clusella, P. Grassberger, F. J. Pérez-Reche, and A. Politi, "Immunization and targeted destruction of networks using explosive percolation," *Phys. Rev. Lett.*, vol. 117, no. 20, 2016, Art. no. 208301.

[24] K. Gong *et al.*, "An efficient immunization strategy for community networks," *PLoS ONE*, vol. 8, no. 12, 2013, Art. no. e83489.

[25] W. Wang *et al.*, "Epidemic spreading on complex networks with general degree and weight distributions," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 90, no. 4, 2014, Art. no. 042803.

[26] T. Bian and Y. Deng, "Identifying influential nodes in complex networks: A node information dimension approach," *Chaos Interdiscipl. J. Nonlin. Sci.*, vol. 28, no. 6, Art. no. 043109, 2018. doi: 10.1063/1.5030894.

[27] D. Perez, J. Togelius, S. Samothrakis, P. Rohlfshagen, and S. M. Lucas, "Automated map generation for the physical traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 18, no. 5, pp. 708–720, Oct. 2014.

[28] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 264–283, Apr. 2015.

[29] S. Jiang and S. Yang, "A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 329–346, Jun. 2017.

[30] Y. Tang, Z. Wang, H. Gao, S. Swift, and J. Kurths, "A constrained evolutionary computation method for detecting controlling regions of cortical networks," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 9, no. 6, pp. 1569–1581, Nov./Dec. 2012.

[31] C. Pizzuti, "A multiobjective genetic algorithm to find communities in complex networks," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 418–430, Jun. 2012.

[32] F. Folino and C. Pizzuti, "An evolutionary multiobjective approach for community discovery in dynamic networks," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1838–1852, Aug. 2014.

[33] X. Wen *et al.*, "A maximal clique based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 363–377, Jun. 2017.

[34] Y. Liu, X. Wang, and J. Kurths, "Optimization of targeted node set in complex networks under percolation and selection," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 98, no. 1, 2018, Art. no. 012313.

[35] Z. Wang *et al.*, "Statistical physics of vaccination," *Phys. Rep.*, vol. 664, pp. 1–113, Dec. 2016.

[36] L. C. Freeman, "Centrality in social networks conceptual clarification," *Soc. Netw.*, vol. 1, no. 3, pp. 215–239, 1978.

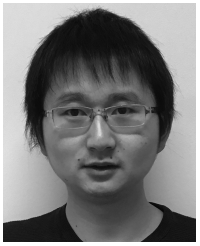
[37] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the Web," Stanford InfoLab, Rep., 1999.

[38] M. Kitsak *et al.*, "Identification of influential spreaders in complex networks," *Nat. Phys.*, vol. 6, no. 11, pp. 888–893, 2010.

[39] C. M. Schneider, T. Mihaljev, S. Havlin, and H. J. Herrmann, "Suppressing epidemics with a limited amount of immunization units," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 84, no. 6, 2011, Art. no. 061911.

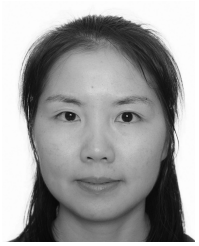
[40] Z. Galil and G. F. Italiano, "Data structures and algorithms for disjoint set union problems," *ACM Comput. Surveys*, vol. 23, no. 3, pp. 319–344, 1991.

[41] D. Achlioptas, R. M. D'souza, and J. Spencer, "Explosive percolation in random networks," *Science*, vol. 323, no. 5920, pp. 1453–1455, 2009.
[42] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
[43] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
[44] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Math.*, vol. 6, no. 1, pp. 29–123, 2009.
[45] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: Densification laws, shrinking diameters and possible explanations," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining*, 2005, pp. 177–187.
[46] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Trans. Knowl. Disc. Data*, vol. 1, no. 1, p. 2, 2007.
[47] R. Matei, A. Iamnitchi, and P. Foster, "Mapping the Gnutella network," *IEEE Internet Comput.*, vol. 6, no. 1, pp. 50–57, Jan./Feb. 2002.
[48] J. Gehrke, P. Ginsparg, and J. Kleinberg, "Overview of the 2003 KDD cup," *ACM SIGKDD Explor. Newslett.*, vol. 5, no. 2, pp. 149–151, 2003.
[49] B. Klimt and Y. Yang, "Introducing the Enron corpus," in *Proc. CEAS*, 2004. [Online]. Available: https://dblp.uni-trier.de/db/conf/ceas/ceas2004.html
[50] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2011, pp. 1082–1090.
[51] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowl. Inf. Syst.*, vol. 42, no. 1, pp. 181–213, 2015.
[52] P. Erdös and A. Rényi, "On random graphs I," *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959. [Online]. Available: http://www.leonidzhukov.net/hse/2016/networks/papers/erdos-1959-11.pdf
[53] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. New York, NY, USA: Springer, 1996.
[54] W. Banzhaf, "The 'molecular' traveling salesman," *Biol. Cybern.*, vol. 64, no. 1, pp. 7–14, 1990.
[55] D. B. Fogel, "An evolutionary approach to the traveling salesman problem," *Bio. Cybern.*, vol. 60, no. 2, pp. 139–144, 1988.
[56] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.
[57] D. B. Fogel, "Applying evolutionary programming to selected traveling salesman problems," *Cybern. Syst.*, vol. 24, no. 1, pp. 27–36, 1993.
[58] K.-I. Goh, B. Kahng, and D. Kim, "Universal behavior of load distribution in scale-free networks," *Phys. Rev. Lett.*, vol. 87, no. 27, 2001, Art. no. 278701.
[59] M. Zhou and J. Liu, "A two-phase multiobjective evolutionary algorithm for enhancing the robustness of scale-free networks against multiple malicious attacks," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 539–552, Feb. 2017.
[60] M. Perc *et al.*, "Statistical physics of human cooperation," *Phys. Rep.*, vol. 687, pp. 1–51, May 2017.
[61] F. A. Rodrigues, T. K. D. M. Peron, P. Ji, and J. Kurths, "The Kuramoto model in complex networks," *Phys. Rep.*, vol. 610, pp. 1–98, Jan. 2016.
[62] E. Bullmore and O. Sporns, "Complex brain networks: Graph theoretical analysis of structural and functional systems," *Nat. Rev. Neurosci.*, vol. 10, no. 3, pp. 186–198, 2009.
[63] G. Del Ferraro *et al.*, "Finding influential nodes for integration in brain networks using optimal percolation theory," *Nat. Commun.*, vol. 9, no. 1, p. 2274, 2018.



Yang Liu received the M.E. degree in computer science from Southwest University, Chongqing, China, in 2016. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Technische Universität Berlin, Berlin, Germany.

He is currently with the Potsdam Institute for Climate Impact Research, Potsdam, Germany, researching on complexity science. His current research interests include complex networks, bio-inspired computing, and graphical model.



Xi Wang received the M.S. degree in computer science from Sichuan University, Chengdu, China, in 2016. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. Her current research interests include medical image analysis, deep learning, and computer-aided classification and detection.

Jürgen Kurths received the Diploma degree in mathematics from the University of Rostock, Rostock, Germany, and the Ph.D. degree from the GDR Academy of Sciences, Berlin, Germany, in 1983.

He was a Full Professor with the University of Potsdam, Potsdam, Germany, from 1994 to 2008. He has been a Professor of Nonlinear Dynamics with the Humboldt University, Berlin, and the Chair of Complexity Science with the Potsdam Institute for Climate Impact Research, Potsdam, since 2008, and a Sixth-Century Chair of Aberdeen University, Aberdeen, U.K., since 2009. He has authored over 650 papers with over 39 000 citations and an H-factor of 88. His current research interests include synchronization, complex networks, and time series analysis and their applications.

Dr. Kurths was a recipient of the Highly Cited Researcher Award in Engineering and Physics, the Alexander von Humboldt Research Award from CSIR, India, in 2005, and the Honorary Doctorate from Lobachevsky University Nizhny Novgorod in 2008 and State University Saratov in 2012. He is an Editor of journals, such as *PLoS ONE*, the *Philosophical Transaction of the Royal Society A*, and the *Journal of Nonlinear Science*, and the Editor-in-Chief of *CHAOS*. He is a fellow of the American Physical Society. He became a member of the Academia Europaea in 2010 and the Macedonian Academy of Sciences and Arts in 2012.