

**Optimal stabilization of Boolean networks through collective influence**Jiannan Wang,<sup>1,2,\*</sup> Sen Pei,<sup>3</sup> Wei Wei,<sup>1,2</sup> Xiangnan Feng,<sup>1,2</sup> and Zhiming Zheng<sup>1,2</sup><sup>1</sup>*School of Mathematics and Systems Science, Beihang University, Beijing, China*<sup>2</sup>*Key Laboratory of Mathematics Informatics Behavioral Semantics, Ministry of Education, China*<sup>3</sup>*Department of Environmental Health Sciences, Mailman School of Public Health, Columbia University, New York, New York, USA*

(Received 27 July 2017; revised manuscript received 14 February 2018; published 13 March 2018)

Boolean networks have attracted much attention due to their wide applications in describing dynamics of biological systems. During past decades, much effort has been invested in unveiling how network structure and update rules affect the stability of Boolean networks. In this paper, we aim to identify and control a minimal set of influential nodes that is capable of stabilizing an unstable Boolean network. For locally treelike Boolean networks with biased truth tables, we propose a greedy algorithm to identify influential nodes in Boolean networks by minimizing the largest eigenvalue of a modified nonbacktracking matrix. We test the performance of the proposed collective influence algorithm on four different networks. Results show that the collective influence algorithm can stabilize each network with a smaller set of nodes compared with other heuristic algorithms. Our work provides a new insight into the mechanism that determines the stability of Boolean networks, which may find applications in identifying virulence genes that lead to serious diseases.

DOI: [10.1103/PhysRevE.97.032305](https://doi.org/10.1103/PhysRevE.97.032305)**I. INTRODUCTION**

Boolean networks were proposed as a model of genetic regulatory network by Kauffman in 1969 [1]. Unlike other mathematical models using differential equations, a Boolean network models the dynamics of a gene as a binary (on or off) switch, while interactions among genes are represented by Boolean functions. The large number of parameters describing details of the regulatory dynamics, which often appear in other models, are simply neglected in Boolean networks. Despite this simplification, the Boolean network model is still able to provide deep insights into the dynamics of various biochemical systems [2–6], social networks [7,8], and economic systems [9]. Due to their wide applications, Boolean network models with various topology and update functions have attracted much attention during past decades [10–13].

As the state space of a Boolean network is finite, there must be a time when the system gets back to one of its previous states. Given the deterministic dynamics, a Boolean network will finally evolve along a cycled orbit, called an attractor. The attractor is vital in the study of Boolean networks because of its close relationship with numerous phenomena in real-world complex systems. For example, in intracellular regulation dynamics, it was found that attractors of the genetic network that controls the yeast cell cycle match the actual biological dynamics [14]. In multicellular organisms, such as humans, attractors were believed to correspond to the varieties of cells generated from cell differentiation [1]. One of the key problems about attractors is the stability, i.e., the ability to eliminate small perturbations as the system evolves. In an unstable network, the whole system can be influenced by a very small perturbation. Under this situation, the network dynamics

is affected dramatically, usually leading to a system failure. The stability of Boolean networks is crucial to understanding the regulation of gene networks, as previous research indicated that real-world genetic regulatory networks usually lie on the critical region between stable and unstable [1], so that creatures can not only survive most genetic mutations but also maintain their diversity. Besides, the problem of stability is also relevant to certain kinds of cancer [15], since cells in cancer tissue exhibit much higher heterogeneity than normal ones [16].

In previous works, researchers have made great progress in studying the stability of Boolean networks. In 1985, Derrida *et al.* proposed the annealed approximation to predict the stability of their random Boolean network model [17]. Since then, other researchers have applied the annealed approximation to networks with various degree distributions and update functions [18–20]. However, there are far more topological features that cannot be described solely by degree distributions, such as community structure, degree assortativity, and reciprocity. It was not until 2009 when Pomerance *et al.* proposed the semiannealed approximation that the stability analysis of Boolean network models with arbitrary topology becomes possible. With this method, researchers are able to broaden their research to problems in a variety of Boolean network models constructed from real biological systems, such as the stability of Boolean multilevel networks [21] and the joint effects of topology and update rules on the stability of Boolean networks [22].

Previous research mainly studied the stability of Boolean networks on a macroscopic scale. However, due to the vast heterogeneity in topological and dynamical properties among nodes in the network, it is more than likely that the influence of each node on the network dynamics is quite different and a small fraction of nodes have a disproportionate effect on the stability of the whole system. Identifying influential nodes is important in a series of problems, such as the detection of

\*wangjiannan@buaa.edu.cn

virulence genes and the optimal immunization of epidemic spread [23–32].

Much progress has been made in identifying vital nodes in the dynamics of Boolean networks. According to previous research, nodes in Boolean networks can be classified into three categories: frozen, irrelevant, and relevant [33–35]. Frozen nodes, whose corresponding update functions output constant values, are naturally immune to any possible perturbations. Irrelevant nodes, which can be identified through repeated pruning steps of removing nodes that do not influence others, have their dynamics completely determined by others. Nodes in these two categories are considered not important, while remaining nodes, also referred to as the relevant core, decide the attractor of Boolean networks. Canalized Boolean functions, whose outputs are fixed when one of inputs takes a specific value, also play an important role in Boolean networks. Researches found that some frozen nodes, combining with nodes that have canalized update functions, can form the structure of a self-freezing loop [35,36]. Although nonfrozen nodes are included in a self-freezing loop, all nodes in the loop become frozen after sufficient iterations. Therefore, nodes in self-freezing loops have no influence on the dynamics of Boolean networks either.

In this paper, our objective is to identify and control a minimum set of nodes through which an unstable network can be stabilized. First we construct a theoretical framework to give a mathematical description of this problem. Then we map this problem to the minimization of the largest eigenvalue of a modified nonbacktracking matrix. Following the collective influence theory designed for optimal percolation [23,37–39], we propose the method of collective influence to identify the minimum set of influential nodes. Different from other methods such as PageRank [40] and k-core [41], the collective influence takes into account not only topological features, but also dynamical properties of nodes with respect to stability. Simulation results on four different Boolean network models show that our method outperforms traditional benchmark algorithms. Besides, our method also has superior computational efficiency, making it applicable to large-scale Boolean networks.

## II. STABILITY OF BOOLEAN NETWORKS

A Boolean network consists of  $N$  nodes connected with  $L$  directed edges. For each node  $i$ , its state, often denoted as  $x_i$ , can only be one of the following two states: *on* (1) or *off* (0). Interactions among  $N$  nodes are defined by  $N$  Boolean functions  $\{f_1, f_2, \dots, f_N\}$ , which are often referred to as update functions. At time  $t$ , the state of the network can be represented by an  $N$ -dimensional vector  $X(t) = (x_1(t), x_2(t), \dots, x_N(t))$ . At time  $t + 1$ , its state  $X(t + 1)$  is determined by  $X(t)$  and update functions. Take node  $i$  for an example, given that it has  $k_i$  inputs  $\{i_1, i_2, \dots, i_{k_i}\}$ , its state at time  $t + 1$  is then determined by

$$x_i(t + 1) = f_i(x_{i_1}(t), x_{i_2}(t), \dots, x_{i_{k_i}}(t)). \quad (1)$$

If we denote the set of nodes that input into  $i$  as  $X_i = (x_{i_1}, x_{i_2}, \dots, x_{i_{k_i}})$ , the dynamics of the Boolean network can be represented as

$$x_i(t + 1) = f_i(X_i(t)). \quad (2)$$

The topology of the Boolean network is represented by the adjacent matrix  $A$ , whose elements  $A_{ij} = 1$  if there exists an edge from node  $i$  to  $j$ ; otherwise,  $A_{ij} = 0$ . To define the stability of Boolean networks, we consider two initial states of the network  $X(t_0)$  and  $\tilde{X}(t_0)$ . The Hamming distance between these two states is defined as

$$H(t_0) = \frac{1}{N} \sum_{i=1}^N |x_i(t_0) - \tilde{x}_i(t_0)|. \quad (3)$$

Let us suppose that  $N$  is large enough and the initial Hamming distance between  $X(t_0)$  and  $\tilde{X}(t_0)$  is close, i.e.,  $H(X(t_0), \tilde{X}(t_0)) \ll 1$ . The stability of Boolean networks mainly concerns the behavior of  $H(t)$  as  $t \rightarrow \infty$ . The network is stable if  $H(t)$  is negligible as  $t \rightarrow \infty$ , which indicates the system will eventually go back to normal despite small perturbations. Otherwise, the network is regarded unstable. An unstable network usually fails to recover from a small perturbation spontaneously, therefore an external input can affect its dynamics dramatically.

The Hamming distance has its drawback as a measure of the stability of Boolean networks. As illustrated in Refs. [33,34], for a Boolean network with canalized update functions, its stability depends on the fraction of active nodes in the stationary state. If the stationary state is an attractor in which nodes go through a sequence of states with a period larger than 1, the key parameter that decides stability is not constant. Furthermore, different network copies can be in different phases of the attractor. In this case, the normalized Hamming distance would exhibit a periodic behavior. However, the measure of Hamming distance provides at least a sufficient condition for stability, so it remains proper in this paper.

## III. OPTIMAL STABILIZATION THROUGH COLLECTIVE INFLUENCE

In this section, we discuss the optimal stabilization problem of Boolean networks. Previous research on the stability of Boolean networks mainly focused on the criterion between stable networks and unstable ones [18,19,21]. In a stable Boolean network, small perturbations tend to vanish spontaneously as time evolves. Whereas, in an unstable network, it can cause remarkable disturbance on the dynamics of the whole system. To avoid this unfavorable property of unstable Boolean networks, it is important to know whether we can stabilize an unstable Boolean network by making some nodes immune to any possible perturbations. If we were able to immunize all nodes in the network, it is quite obvious that every Boolean network could be stabilized. But what if we could only immunize a fraction of nodes? How do we stabilize a Boolean network by immunizing a set of nodes as small as possible? Considering the great heterogeneity in topological and dynamical properties among nodes in real biological systems, we certainly do not expect the best strategy is to select targets randomly. In this paper, our purpose is to identify the minimum set of nodes whose immunization can stabilize an unstable Boolean network. In the following paragraphs, this problem is referred to as the optimal stabilization of Boolean networks. The selected targets that are immune to perturbations are defined as controllers.

We start by proposing a mathematical description of the optimal stabilization problem. We use  $\mu_i$  to represent whether a node  $i$  is a controller:  $\mu_i = 0$  if node  $i$  is controlled; otherwise,  $\mu_i = 1$ . Therefore, the vector  $\mu = (\mu_1, \mu_2, \dots, \mu_N)$  contains the information of selected controllers and we call it a configuration of the network. The fraction of controllers in the network is represented by

$$q = 1 - \frac{1}{N} \sum_{i=1}^N \mu_i = 1 - \langle \mu \rangle. \quad (4)$$

Regarding the stability of Boolean networks, we represent the Hamming distance  $H$  as a function of  $q$ ,

$$H(q) = \frac{1}{\tilde{t}} \sum_{t=t^*}^{t^*+\tilde{t}} H(q, t) = \frac{1}{\tilde{t}N} \sum_{t=t^*}^{t^*+\tilde{t}} \sum_{i=1}^N [x_i(t) - \bar{x}_i(t)], \quad (5)$$

where  $t^*$  and  $\tilde{t}$  are large enough to pass any transient or periodic behaviors. In a stable Boolean network, we can always expect that  $\langle H(q) \rangle = 0$ , where  $\langle \cdot \rangle$  stands for the average over all initial values. In an unstable network, however, it always holds  $\langle H(q) \rangle > 0$ . The optimal stabilization problem is to find the minimum fraction  $q_c$  of nodes and the corresponding optimal configuration such that  $\langle H(q_c) \rangle = 0$ :

$$q_c = \min\{q \in [0, 1] : \langle H(q) \rangle = 0\}. \quad (6)$$

For  $q \geq q_c$ , there exist a variety of configurations which are able to stabilize the Boolean network. In contrast, for  $q < q_c$ , the configuration that can stabilize the whole system does not exist. As  $q$  decreasing from 1 to 0, the number of configurations that satisfy  $\langle H(q) \rangle = 0$  also decreases and eventually vanishes at  $q_c$ .

Considering a single node in the network, it can be perturbed only when it is not a controller and at least one of its inputs has been perturbed. Therefore, the parameter  $\mu_i$  itself fails to measure the influence of perturbation. So we need another variable encoding the information about whether a node is perturbed or not. This information is stored in another variable  $v_i$ :  $v_i = 1$  if node  $i$  is perturbed; otherwise,  $v_i = 0$ . The influence of perturbation in the whole system is then represented by the fraction of perturbed nodes, when  $q$  fraction of the nodes in the configuration  $\mu$  are controlled:

$$H(q, \mu) = \frac{1}{N} \sum_{i=1}^N v_i. \quad (7)$$

For a given  $q$ , the optimal stabilization of Boolean networks requires to minimize the influence of perturbation over all possible configurations. However, because an explicit function of  $H(q, \mu)$  is not available, it is difficult to select important nodes in a Boolean network by minimizing  $H(q, \mu)$  directly. Instead, we transform this problem into minimizing the largest eigenvalue of a modified non-backtracking matrix, which can be represented analytically.

To derive the relation between  $v = (v_1, v_2, \dots, v_N)$  and  $\mu = (\mu_1, \mu_2, \dots, \mu_N)$ , we consider a directed edge from node  $i$  to  $j$ . Suppose that node  $j$  is temporarily removed from the network and we concern whether node  $i$  is perturbed. This information is stored in a variable  $v_{i \rightarrow j}$ , which represents the probability that node  $i$  is perturbed in the absence of  $j$ . Clearly, we can

conclude that  $v_{i \rightarrow j} = 0$  if  $\mu_i = 0$ . So we only consider the case when  $\mu_i = 1$ . Given that  $j$  is temporarily removed from the network, node  $i$  is perturbed only because of the event ‘‘at least one of the nodes pointing to node  $i$  other than  $j$  is perturbed.’’ Assuming that the network is locally treelike, the variables  $v_{i \rightarrow j}$  then satisfy the following message-passing equations:

$$v_{i \rightarrow j} = \rho_i \mu_i \left[ 1 - \prod_{k \in \partial i \setminus j} (1 - v_{k \rightarrow i}) \right], \quad (8)$$

where  $\rho_i$  represents the sensitivity of node  $i$  [42] and  $\partial i$  is the set of nodes that input into  $i$ . Obviously, the system in Eq. (8) admits the solution  $\{v_{i \rightarrow j} = 0\}$  for all  $i, j$ . As a result, the impact of perturbation in the whole network  $H(q, \mu) = 0$ . By linearizing these equations and neglecting terms with higher orders, we conclude that the stability of the solution  $\{v_{i \rightarrow j} = 0\}$  depends on the largest eigenvalue of the linear operator of the  $L \times L$  matrix  $\hat{M}$ , whose elements can be represented in terms of a nonbacktracking matrix  $\hat{B}$  [43,44]:

$$\hat{M}_{k \rightarrow l, i \rightarrow j} = \mu_i \rho_i \hat{B}_{k \rightarrow l, i \rightarrow j}, \quad (9)$$

where

$$\hat{B}_{k \rightarrow l, i \rightarrow j} = \begin{cases} 1 & \text{if } l = i \text{ and } j \neq k \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

We use  $\lambda(q, \mu)$  to represent the largest eigenvalue of  $\hat{M}$ , which depends on the fraction of controllers  $q$  and the configuration of the network  $\mu$ . According to the Frobenius theorem [45],  $\lambda(q, \mu)$  is real and positive. The stability of the solution  $H(q, \mu) = 0$  is determined by the critical condition  $\lambda(q_c, \mu^*) = 1$ , where  $\mu^*$  is the optimal configuration. When  $q < q_c$ , for each configuration  $\mu$   $\lambda(q, \mu) > 1$ , hence it is impossible to find a set of controllers such that  $H(q, \mu) = 0$ . On the contrary, when  $q > q_c$ , there are two different possibilities. On one hand, for some nonoptimal configurations we have  $\lambda(q, \mu) > 1$ , which are unable to stabilize the whole network; on the other hand, there exist configurations that satisfy  $\lambda(q, \mu) < 1$ , which correspond to a stable solution of  $H(q, \mu) = 0$ . As we approach from above,  $q \mapsto q_c^+$ , the number of configurations satisfying  $\lambda(q, \mu) < 1$  gradually decreases and eventually vanishes at  $q_c$ .

Due to the complexity of Boolean network dynamics, it is difficult to give an analytical description of  $\lambda(q, \mu)$ . Our approach is to approximate  $\lambda(q, \mu)$  with power method, which converges to its exact solution after sufficient steps of iterations. For a given configuration, we can use  $\lambda(\mu)$  to represent the largest eigenvalue of  $\hat{M}$  and the parameter  $q$  can be omitted. Let us consider an arbitrary nonzero vector  $\omega_0$ . For convenience, we suppose  $\omega_0 = (1, 1, \dots, 1)^t$  and use  $\omega_l$  to represent the result of  $\omega_0$  after  $l$  iterations:

$$\omega_l = \hat{M}^l \omega_0. \quad (11)$$

According to power method, the largest eigenvalue of  $\hat{M}$  decides the growth rate of  $\omega_0$ ; therefore, we can approximate  $\lambda(\mu)$  by calculating the growth rate of  $\omega_l$  as  $l \rightarrow \infty$ :

$$\lambda(\mu) = \lim_{l \rightarrow \infty} \lambda_l(\mu) = \lim_{l \rightarrow \infty} \left( \frac{|w_l(\mu)|}{|w_0|} \right)^{\frac{1}{l}}. \quad (12)$$

For a finite  $l$ , we map this problem to a many-body interaction through which we can give an approximation of

the norm of  $w_l(\mu)$  as (see Appendix A)

$$|w_l(\mu)|^2 = \sum_{i=1}^N K_i^{\text{in}} \sum_{j \in \partial \text{Ball}(i,l)} \left( \prod_{k \in P_l(i,j)} \mu_k \rho_k \right) K_j^{\text{out}}, \quad (13)$$

where  $\text{Ball}(i,l)$  consists of the nodes within a ball of radius  $l$  from node  $i$  (defined as the shortest path),  $\partial \text{Ball}(i,l)$  is the surface of the ball, and  $P_l(i,j)$  is the shortest directed path of length  $l$  from node  $i$  to  $j$ . Here, we define the collective influence (CI) score of node  $i$  as

$$S_i^{\text{CI}} = K_i^{\text{in}} \sum_{j \in \partial \text{Ball}(i,l)} \left( \prod_{k \in P_l(i,j)} \mu_k \rho_k \right) K_j^{\text{out}}. \quad (14)$$

Collective influence measures the contribution of each node to the largest eigenvalue of  $\hat{M}$ , similar to the idea in Ref. [52] developed based on the adjacent matrix. With this approximation,  $|w_l(n)|^2$  is the sum of collective influence of all nodes:

$$|w_l(\mu)|^2 = \sum_{i=1}^N S_i^{\text{CI}}. \quad (15)$$

To minimize  $\lambda(\mu)$ , our main idea is to take advantage of the greedy algorithm and select one controller at a time. At each step, we select the node with the highest collective influence as a controller, which results in the biggest drop in the value of  $|w_l(n)|^2$ . The selected controller is virtually removed from the network before the next selection is made. Then we continue this procedure until the network is finally stabilized. One advantage of using the greedy algorithm is that each time a controller is selected, the factors  $\mu_i$  in the formula of collective influence can be just ignored. Since previously selected controllers have been virtually removed,  $\mu_i = 1$  for every remaining node in the network. One problem of collective influence is that a proper radius  $l$  still needs to be specified. Intuitively, the performance of collective influence is better with a larger radius  $l$ , but in the mean time the computation complexity increases dramatically. When we consider the case  $l = 0$ , we get  $S_0^{\text{CI}}(i) = K_i^{\text{in}} q_i$ , which is similar to the high degree strategy. Previous works have proved this strategy to be less than satisfying [27]. Thus, we go further to consider the case  $l = 1$ , where the collective influence is given by

$$S_1^{\text{CI}}(i) = K_i^{\text{in}} \rho_i \sum_{j \in \Gamma_i} \rho_j K_j^{\text{out}}, \quad (16)$$

where  $\Gamma_i$  is the set of nodes that node  $i$  points to.  $S_1^{\text{CI}}(i)$  consists of not only the in-degree of node  $i$  itself, but also the topological information of its nearest neighbors. We can expect a better performance when considering larger radii, but the computation will be more time-consuming.

In general, the collective influence algorithm is scalable for large networks with a computational complexity  $O(N \log N)$ . Computing the collective influence is equivalent to iteratively visiting neighbors of each node layer by layer within a radius of  $l$ . Since  $l$  is finite, it takes  $O(1)$  time to compute the collective influence of each node. Initially, we have to calculate the collective influence for all nodes in the network. However, during later steps, we only need to recalculate for nodes within a  $l + 1$

radius from the selected controllers, which scales as  $O(N)$ . When it comes to selecting the node with the highest collective influence, we can make use of the data structure of heap that takes  $O(\log N)$  time. Therefore, the overall complexity of the collective influence algorithm is  $O(N \log N)$ . In the following section, we take  $S_1^{\text{CI}}$  as the representative of collective influence and discuss its performance compared with other algorithms.

Similar to the idea proposed in Ref. [53], the contribution of an individual node to the largest eigenvalue  $\lambda(\mu)$  can also be approximated with the left and right eigenvalues of  $\hat{M}$ . With this approximation, the optimal stabilization problem can also be solved using the cavity method. This provides a competitive result with our collective influence algorithm, but at much higher computation expense. Details are exhibited in Appendix B.

#### IV. NUMERICAL SIMULATIONS

In this section, we construct four different Boolean networks on which we can test the performance of collective influence algorithm. First we consider Kauffman's  $N$ - $K$  network model, where all nodes have exactly  $K$  inputs randomly selected from the other  $N - 1$  nodes. The degree distribution of the  $N$ - $K$  network model does not show much heterogeneity, which is quite different from real genetic regulatory networks. Considering that real genetic regulatory networks usually have short tailed in-degree distributions and long-tailed out-degree distributions [46,47], we construct the second network with Poisson distributed in-degrees and scale-free out-degrees. These two networks above are typical toy models of Boolean networks. The following two networks are constructed using data from real-world systems. The third network captures the innovation spread among 241 physicians in four towns of Illinois: Peoria, Bloomington, Quincy, and Galesburg [48]. The last network was created from a survey on the social relationship among adolescents [49]. On each of the above four networks, update functions are given in the form of truth tables. For any inputs of  $f_i$ , its corresponding output is randomly chosen from  $\{0, 1\}$  with probability 0.5. As a result, all nodes in the networks have a common sensitivity  $\rho_i \equiv \rho \equiv 0.5$ .

On each network, we compare the performance of collective influence with the following methods: high degree (HD) [50], eigenvector centrality (EC) [51], Google PageRank (PR) [40], voter rank (VR) [26], and degree product, which have been proved useful in detecting influential nodes in complex networks. High degree strategy, as its name implies, defines the influence of a node  $i$  with its degree. Here, we choose  $K_i^{\text{in}}$  as the degree rank of node  $i$ . To get a better performance, we adopt the adaptive version of high degree method (HDA). After each selection, the degree of each node is recalculated. However, as mentioned in various studies, nodes with high degree do not necessarily possess high influence. The eigenvector centrality fixes this problem by considering not only the number of a node's neighbors, but also the influence of its neighbors, known as the mutual enhancement effect. For each node  $i$ , its score of eigenvector centrality  $S^{\text{EC}}(i)$  is given by

$$S^{\text{EC}}(i) = \sum_j A_{ij} S^{\text{EC}}(j). \quad (17)$$

PageRank is a famous algorithm that is used to rank websites in google search engines and other commercial scenarios. According to the Page rank algorithm, the influence of a webpage is determined by random walking on the network of web pages. Mathematically, the PageRank score of node  $i$  is

$$S^{\text{PR}}(i) = \sum_j A_{ij} \frac{S^{\text{PR}}(j)}{k_j^{\text{out}}}. \quad (18)$$

Due to the ubiquitous existence of community structure in complex networks, influential nodes in complex networks are more likely to connect with each other, which result in the fact that their sphere of influence tend to overlap. To avoid such condition, in the voter rank algorithm, each node is granted an initial voting ability  $\theta_i$  and the score of voter rank is calculated as

$$S^{\text{VR}}(i) = \sum_j A_{ij} \theta_j. \quad (19)$$

At each step, the voter rank selects one single node with the highest voting score. Then the vote abilities of its neighbors spontaneously decrease. Therefore, the nodes nearby are less likely to be chosen in the following process and the selected nodes are less likely to be close to each other. Degree product is constructed by minimizing the largest eigenvalue  $\lambda_Q$  proposed in Ref. [15] with the cavity method. At each step, the score of each node is calculated as  $S^{\text{DP}}(i) = \rho_i K_i^{\text{in}} K_i^{\text{out}}$ . Since  $\rho_i \equiv \rho$ , this can simply be replace by  $S^{\text{DP}}(i) = K_i^{\text{in}} K_i^{\text{out}}$ .

For each algorithm, we start from  $q = 0$  and pick the controllers one after another until  $q = 0.2$ . Each time a controller is chosen, we calculate the average Hamming distance  $\langle H \rangle$  to see whether or not the network has been stabilized.

As for the calculation of Hamming distance  $\langle H \rangle$ , we take it as the average of 100 initial values. For each initial value,  $H$  is calculated through the following procedure. First, we randomly generate an initial value  $X(t)$  and evolve it according to update functions until  $t_0 = 100$ , where we expect it to have completed any transient behaviors. Next, we chose a small fraction ( $\varepsilon = 0.01$ ) of its components and flip their states to create a perturbed value  $\tilde{X}(t_0)$ . In other words,  $\tilde{x}_i(t_0) = 1 - x_i(t_0)$  if node  $i$  is perturbed, otherwise  $\tilde{x}_i(t_0) = x_i(t_0)$ . The initial Hamming distance is  $H(X(t_0), \tilde{X}(t_0)) = 0.01$ . Finally, we take  $X(t_0)$  and  $\tilde{X}(t_0)$  as the initial values and evolve both of them in parallel. Here we stress that for nodes that have been chosen as controllers, their states in both orbits are always the same, since controllers are immune to any perturbations. Our main interest is the long-time behavior of  $H$ , which is calculated by averaging  $H(X(t), \tilde{X}(t))$  from  $t = 400$  to  $t = 500$ . This whole procedure is repeated and  $\langle H \rangle$  is the average of  $H$  over all initial values.

From Fig. 1 we can see the performance of the six algorithms mentioned above in Kauffman's  $N$ - $K$  network model. The network consists of  $N = 20\,000$  nodes and its average degree  $K = 3$ . For each algorithm, the average Hamming distance  $\langle H \rangle$  decreases with the increase of the fraction of controllers  $q$ . During the process, collective influence outperforms the other five algorithms and stabilizes the network with the minimal fraction of controllers  $q_c^{\text{CI}} \approx 0.13$ , followed by high degree and degree product that stabilize the network at  $q_c^{\text{HD}} \approx q_c^{\text{DP}} \approx 0.17$ .

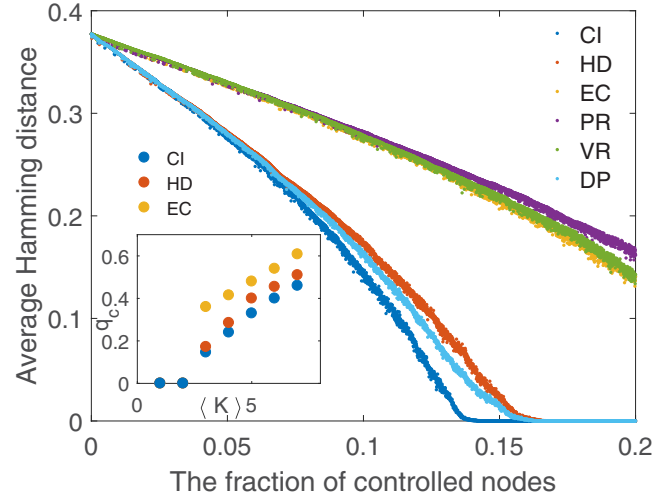


FIG. 1. Normalized average Hamming distance  $\langle H \rangle$  plotted against the fraction of controllers  $q$  in  $N$ - $K$  network. The network consists of 20 000 nodes and its average degree is 3. The performances of collective influence, high degree, eigenvector centrality, PageRank, voter rank, and degree product are represented in different colors. The small panel shows the results of  $q_c$  with the increase of average degree.

As for the other three algorithms, their performances are rather close. The average Hamming distance remains  $\langle H \rangle \approx 0.15$  even when  $q = 0.2$  of nodes in the network are under control. One interesting phenomenon shown in Fig. 1 is that the performance of high degree beats those more complex algorithms like eigenvector centrality, PageRank, and voter rank, which usually perform quite well. One possible reason could lie in the difference between optimal stabilization of Boolean networks and those rank problems for which these algorithms are designed. In these three algorithms, a node usually exhibits higher importance if it is pointed by more nodes with higher importance themselves. However, in the problem of optimal stabilization, it is much the opposite. In this problem, one node enjoys higher influence by pointing to more nodes with higher influence. The small panel shows that with the average degree of the  $N$ - $K$  increases, the minimal fraction of controllers  $q_c$  increases as well. During the process, collective influence still outperforms other algorithms.

In Fig. 2, we show the performances of the six algorithms on a heterogeneous network. The networks is constructed using the configuration model, which consists of 1000 nodes with average degree 3. Figure 2 shows that with the increase of the fraction of controllers  $q$ , the network reaches stable region first at  $q_c^{\text{CI}} \approx 0.03$  when collective influence is applied. The degree product performs competitively with  $q_c^{\text{DP}} \approx 0.04$ . Again, the high degree strategy outperforms other three algorithms and stabilizes the network at  $q_c^{\text{HD}} \approx 0.07$ . The performances of eigenvector centrality, PageRank and voter rank are similar, which can not stabilize the network until  $q = 0.14$ . When we compare Figs. 1 and 2, it is interesting that although these two networks have the same average degree, an algorithm can behave quite differently on the two networks. In general, an algorithm can achieve stabilization with a much smaller fraction of controllers in the heterogeneous network, since it is easier to control the dynamics of the whole system

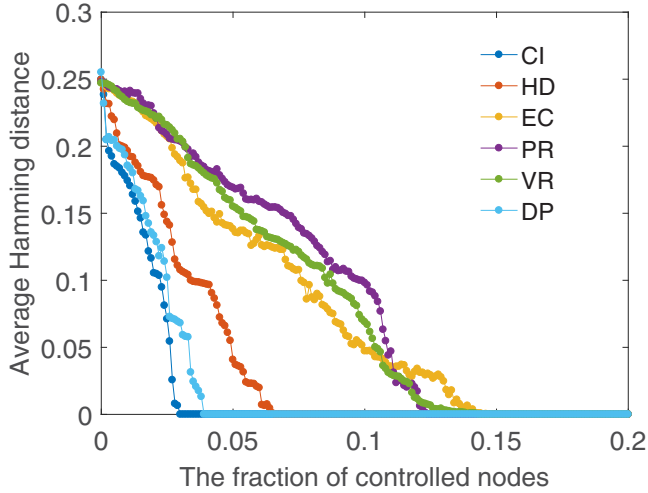


FIG. 2. Normalized average Hamming distance  $\langle H \rangle$  plotted against the fraction of controllers  $q$  in a heterogeneous network. The network is constructed using the configuration model. The in-degrees of nodes follow a Poisson distribution and the out-degrees are scale-free. The network consists of 1000 nodes and its average degree is 3. The performances of collective influence, high degree, eigenvector centrality, PageRank, voter rank, and degree product are represented in different colors.

if the network exhibits more topological heterogeneity. In Kauffman’s  $N$ - $K$  network model, however, the importance of nodes is quite similar to each other, thus it is unlikely to achieve stabilization by simply controlling a small fraction of them.

Figures 3 and 4 show the performances of the six algorithms when applied to real-world networks [48,49]. In Fig. 3, each node in the network represents a physician and each directed

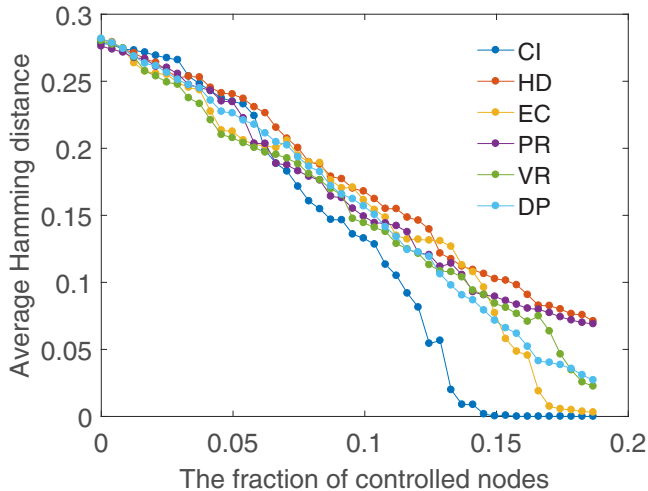


FIG. 3. Normalized average Hamming distance  $\langle H \rangle$  plotted against the fraction of controllers  $q$  in the physician network. The network consists of 241 nodes and 1098 edges. The performances of collective influence, high degree, eigenvector centrality, PageRank, voter rank, and degree product are represented in different colors. Simulations are performed for an initial Hamming distance  $H(t_0) = 0.01$  and the results of  $\langle H \rangle$  are averaged over 100 random initial values.

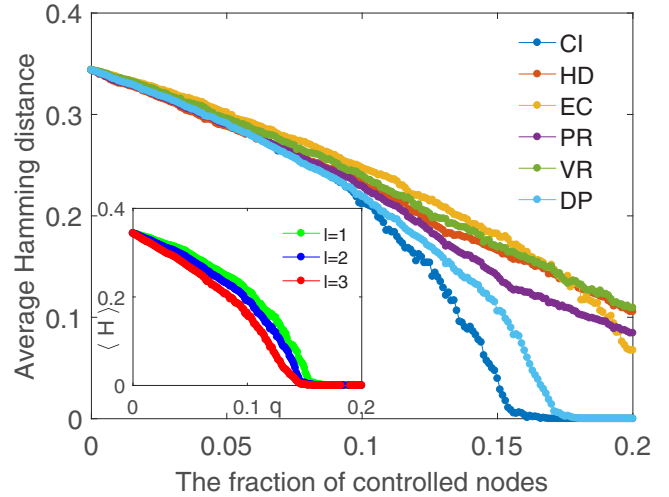


FIG. 4. Normalized average Hamming distance  $\langle H \rangle$  plotted against the fraction of controllers  $q$  in student social network. The network consists of 1000 nodes and 4175 edges, which is part of the adolescent social network constructed from the survey. The performances of collective influence, high degree, eigenvector centrality, PageRank, voter rank, and degree product are represented in different colors. The small panel shows the performance of the collective influence algorithm when larger radii are applied.

edge from node  $i$  to node  $j$  shows that physician  $i$  regards physician  $j$  as his friend or he turns to  $j$  if he needs advice or is interested in a discussion. There always only exists one edge between two nodes even if more than one of the listed conditions are true. The network in Fig. 4 is created from a survey including 2539 students. In the survey, each student was asked to list his five best female and five male friends. Each node represents a student and an edge from node  $i$  to  $j$  means that student  $i$  regards student  $j$  as a friend. In the simulation, we choose part of the network that contains 1000 students among them. As is shown in Figs. 3 and 4, collective influence outperforms the other algorithms in both networks. In Fig. 3, the physician network becomes stable at  $q_c^{CI} \approx 0.15$  when collective influence is applied, followed by eigenvector centrality which achieves stable region at  $q_c^{EC} \approx 0.18$ . According to Fig. 4, collective influence and degree product are the only algorithms that are able to achieve stabilization of adolescent social network, while for the other four algorithms, the network will not be stabilized even at  $q = 0.2$ . However, collective influence still outperforms degree product with  $q_c^{CI} \approx 0.16$  and  $q_c^{DP} \approx 0.18$ . We continue to compare the performances of collective influence when  $l = 1$ ,  $l = 2$ , and  $l = 3$ , the results in the inset show that the performance of collective influence algorithm improves with a larger radius  $l$ , but the improvement is rather limited.

### V. CONCLUSION

In this paper, we study the optimal stabilization problem of Boolean networks, which aims to identify the minimal set of influential nodes whose immunization is capable of stabilizing an unstable Boolean network. Since it is difficult to represent the average Hamming distance as a function of network

configuration, we transform this problem into minimizing the largest eigenvalue of a modified nonbacktracking matrix that determines the stability of Boolean networks. We propose collective influence that enables us to identify influential nodes with respect to the stability of Boolean networks. To test the performance of collective influence, we construct two toy networks and two real-world networks on which we compare the performance of collective influence with other five algorithms: high degree, eigenvector centrality, PageRank, voter rank, and degree product. The results show that in all four networks, our collective influence algorithm outperforms others by stabilizing the networks with a smaller fraction of controllers. We also find that it is easier to stabilize a Boolean network with more heterogeneity. Our work may contribute to the identification of virulence genes that cause serious inherited diseases. Besides, it also provides a new insight into the mechanism that determines the stability of Boolean networks, which is useful to control dynamics in a series of real biological systems.

### APPENDIX A

In this Appendix we present the details of approximations of  $\lambda_l(\mu)$ , which eventually converges to  $\lambda(\mu)$  as  $l \rightarrow \infty$ . The indices of  $\hat{M}$  correspond to directed edges of the network and its elements contain the information of connections between these edges. For computation convenience, we embed it into an  $N \times N \times N \times N$  matrix  $M$ :

$$M_{ijkl} = \mu_k \rho_k A_{ij} A_{kl} \delta_{jk} (1 - \delta_{il}), \quad (\text{A1})$$

where  $i, j, k$ , and  $l$  all vary from 1 to  $N$ .  $M$  encodes the same information with  $\hat{M}$  in a higher dimension, but its indices correspond to nodes in the network, which is similar to the adjacent matrix. As for the  $L$ -dimensional initial vector  $\omega_0$  whose elements are all 1, its projection in the enlarged  $N \times N \times N \times N$  space is an  $N \times N$  vector, whose elements are given by  $|w_0\rangle_{ij} = A_{ij}$ . Given the topology of the Boolean network, we can calculate the right vector  $|w_1(\mu)\rangle$  as

$$|w_1\rangle_{ij} = \sum_{kl} M_{ijkl} |w_0\rangle_{kl}. \quad (\text{A2})$$

Combining Eqs. (A1) and (A2), we conclude that

$$|w_1\rangle_{ij} = \sum_{kl} \mu_k \rho_k A_{ij} A_{kl} \delta_{jk} (1 - \delta_{il}) A_{kl}. \quad (\text{A3})$$

For each term after the summation symbol in Eq. (A3), it is nonzero only when  $j = k$  and  $i \neq l$ . Considering our locally treelike assumption, the right vector  $|w_1(\mu)\rangle$  can be calculated as

$$|w_1\rangle_{ij} = \mu_j \rho_j A_{ij} K_j^{\text{out}}. \quad (\text{A4})$$

Similarly, the left vector  $\langle w_1(\mu)|$  can be calculated as

$${}_{ij}\langle w_1| = \sum_{kl} {}_{kl}\langle w_0| M_{kl ij} = \mu_i \rho_i A_{ij} K_i^{\text{in}}. \quad (\text{A5})$$

Using the left and right vectors, we can write the expression of the norm of  $|w_1(\mu)\rangle$  as

$$\begin{aligned} |w_1(\mu)|^2 &= \sum_{ij} {}_{ij}\langle w_1| w_1\rangle_{ij} \\ &= \sum_{ij} \mu_j \rho_j A_{ij} K_j^{\text{out}} \mu_i \rho_i A_{ij} K_i^{\text{in}} \\ &= \sum_{ij} \mu_i \mu_j \rho_i \rho_j A_{ij} K_i^{\text{in}} K_j^{\text{out}}. \end{aligned} \quad (\text{A6})$$

The norm of  $\omega_0$  can be easily calculated as

$$|w_0(\mu)|^2 = \sum_{ij} {}_{ij}\langle w_0| w_0\rangle_{ij} = L. \quad (\text{A7})$$

According to Eq. (12), we can finally give the mathematical formula of  $\lambda_1(\mu)$ :

$$\lambda_1(\mu) = \left( \frac{1}{L} \sum_{ij} \mu_i \mu_j \rho_i \rho_j A_{ij} K_i^{\text{in}} K_j^{\text{out}} \right)^{\frac{1}{2}}. \quad (\text{A8})$$

Similarly, we can calculate the right vector  $|w_2(\mu)\rangle$  as

$$\begin{aligned} |w_2\rangle_{ij} &= \sum_{kl} M_{ijkl} |w_1\rangle_{kl} \\ &= \mu_j \rho_j A_{ij} \sum_l A_{jl} \mu_l \rho_l K_l^{\text{out}} (1 - \delta_{il}), \end{aligned} \quad (\text{A9})$$

while its left vector  $\langle w_2(\mu)|$  is given by

$$\begin{aligned} {}_{ij}\langle w_2| &= \sum_{kl} {}_{kl}\langle w_1| M_{kl ij} \\ &= \mu_i \rho_i A_{ij} \sum_k \mu_k \rho_k A_{ki} K_k^{\text{in}} (1 - \delta_{jk}). \end{aligned} \quad (\text{A10})$$

Thus, we can calculate the norm of  $|w_2(\mu)\rangle$ :

$$\begin{aligned} |w_2(\mu)|^2 &= \sum_{ij} {}_{ij}\langle w_2| w_2\rangle_{ij} \\ &= \sum_{ijkl} \mu_i \rho_i \mu_j \rho_j \mu_k \rho_k \mu_l \rho_l A_{ij} A_{jk} A_{kl} \\ &\quad \times K_i^{\text{in}} K_l^{\text{out}} (1 - \delta_{ik}) (1 - \delta_{jl}). \end{aligned} \quad (\text{A11})$$

We are able to represent  $\lambda_2(\mu)$  as

$$\begin{aligned} \lambda_2(\mu) &= \left[ \frac{1}{L} \sum_{ijkl} \mu_i \rho_i \mu_j \rho_j \mu_k \rho_k \mu_l \rho_l A_{ij} A_{jk} A_{kl} \right. \\ &\quad \left. \times K_i^{\text{in}} K_l^{\text{out}} (1 - \delta_{ik}) (1 - \delta_{jl}) \right]^{\frac{1}{4}}. \end{aligned} \quad (\text{A12})$$

From the formulas of  $\lambda_1(\mu)$  and  $\lambda_2(\mu)$ , we can write down higher orders of interactions  $\lambda_n(\mu)$  [23]. The order of interaction is defined as the number of nodes appearing in it. In the case  $l = 1$ ,  $\lambda_1$  corresponds to a two-body problem concerning  $i$  and  $j$ . In the interaction, since node  $i$  points a directed edge to node  $j$ , the variables  $\mu_i \rho_i$  and  $\mu_j \rho_j$  are multiplied by each other. This factor is then multiplied by the in-degree of the initial node and the out-degree of the ending

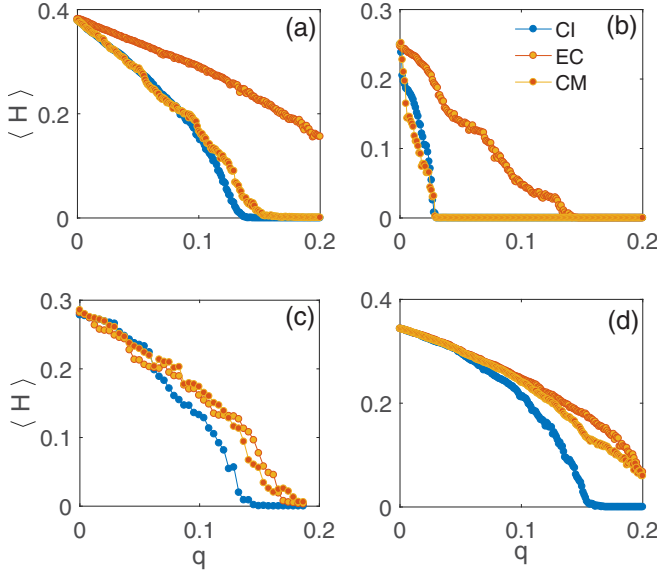


FIG. 5. Normalized average Hamming distance  $\langle H \rangle$  plotted against the fraction of controllers  $q$  in (a) Kauffman's model with  $N = 5000$  and  $K = 3$ ; (b) configuration model with Poisson distributed in-degrees and scale-free out-degrees; (c) physician network consisting of 241 nodes and 1098 edges; (d) student social network with 1000 nodes and 4175 edges.

node of the edge, which equals to  $\mu_i \rho_i \mu_j \rho_j K_i^{\text{in}} K_j^{\text{out}}$ . As for the case  $l = 2$ , we can find similarly that  $\lambda_2$  corresponds to a  $l = 2$  nonbacktracking walk on Boolean networks, which is a four-body problem if  $i \neq l$ , or a three-body problem when  $i = l$ . Here, the series expansion of the maximum eigenvalue can be written in terms of a systematic diagrammatic expansion of increasing levels of multibody interactions. Typically,  $\lambda_l$  corresponds to a nonbacktracking walk of length  $l$ , which can involve as much as  $2l$  nodes in the interaction. We stress that the initial and final nodes of the nonbacktracking walks do not necessarily need to be different, since loops are allowed in nonbacktracking walks. However, due to the fact that most networks in the real world are sparse and locally treelike, we decide that those nonbacktracking walks with loops are negligible and each nonbacktracking walk from node  $i$  to  $j$  is

in fact a shortest path between the two nodes. Thus, the norm of  $|w_l(\mu)|$  can be represented as

$$|w_l(\mu)|^2 = \sum_{i=1}^N K_i^{\text{in}} \sum_{j \in \partial \text{Ball}(i,l)} \left( \prod_{k \in P_l(i,j)} \mu_k \rho_k \right) K_j^{\text{out}}. \quad (\text{A13})$$

## APPENDIX B

In this Appendix we present an alternative approach to minimize the largest eigenvalue  $\lambda(\mu)$ . Following the cavity method (CM) proposed in Ref. [53], the importance of node  $i$  can be measured by the amount  $-\Delta\lambda_i$  by which  $\lambda$  decreases upon removal of the node, normalized by  $\lambda$ :  $S^{\text{CM}}(i) = -\frac{\Delta\lambda_i}{\lambda}$ . Given the left eigenvector  $\phi$  and right eigenvector  $\varphi$  of  $\hat{M}$ , where  $\hat{M}\varphi = \lambda(\mu)\varphi$  and  $\phi^T \hat{M} = \lambda(\mu)\phi^T$ , we consider the perturbation after the removal of a single node  $i$ . Denote the matrix after the removal by  $\hat{M} + \Delta\hat{M}_i$ , the largest eigenvalue of  $\hat{M} + \Delta\hat{M}_i$  by  $\lambda + \Delta\lambda_i$  and its corresponding right eigenvector by  $\varphi + \Delta\varphi_i$ . We obtain that

$$(\hat{M} + \Delta\hat{M}_i)(\varphi + \Delta\varphi_i) = (\lambda + \Delta\lambda_i)(\varphi + \Delta\varphi_i). \quad (\text{B1})$$

For large matrices, it is reasonable to assume that the removal of a link or node as a small effect on the spectral properties of the network. Left multiplying Eq. (B1) by left eigenvector  $\phi^T$  and neglecting second-order terms, we can obtain that

$$\Delta\lambda_i = \frac{\phi^T \Delta\hat{M}_i \varphi}{\phi^T \varphi}. \quad (\text{B2})$$

At each step, we greedily control the node with the highest  $\Delta\lambda_i$ . After that, the eigenvectors  $\lambda$  and  $\varphi$  are recalculated before another node is selected. We test the performance of the cavity method on the four networks mentioned above. Here we use a smaller  $N-K$  network model with  $N = 5000$  instead. Results in Fig. 5 show that in most cases the performance of collective influence algorithm outperforms the cavity method. Specially, in the heterogeneous network with Poisson distributed in-degrees and scale-free out-degrees, the performance of cavity method is competitive with collective influence algorithm. However, the computational complexity of collective influence is much lower.

- 
- [1] S. A. Kauffman, *J. Theor. Biol.* **22**, 437 (1969).  
 [2] R. Serra, M. Villani, A. Barbieri, S. A. Kauffman, and A. Colacci, *J. Theor. Biol.* **265**, 185 (2010).  
 [3] A. L. Bauer, T. L. Jackson, Y. Jiang, and T. Rohlf, *J. Theor. Biol.* **264**, 838 (2010).  
 [4] M. K. Morris, J. Saez-Rodriguez, P. K. Sorger, and D. A. Lauffenburger, *Biochemistry* **49**, 3216 (2010).  
 [5] R. Albert and H. G. Othmer, *J. Theor. Biol.* **223**, 1 (2003).  
 [6] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang, *Proc. Natl. Acad. Sci. USA* **101**, 4781 (2004).  
 [7] M. E. J. Newman, *Phys. Rev. E* **66**, 016128 (2002).  
 [8] P. Rämö, S. Kauffman, J. Kesseli, and O. Yli-Harja, *Physica D* **227**, 100 (2007).  
 [9] M. Paczuski, K. E. Bassler, and A. Corral, *Phys. Rev. Lett.* **84**, 3185 (2000).  
 [10] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, *Bioinformatics* **18**, 261 (2002).  
 [11] J. E. S. Socolar and S. A. Kauffman, *Phys. Rev. Lett.* **90**, 068702 (2003).  
 [12] R. Zhang, H. L. D. de S.Cavalcante, Z. Gao, D. J. Gauthier, J. E. S. Socolar, M. M. Adams, and D. P. Lathrop, *Phys. Rev. E* **80**, 045202(R) (2009).  
 [13] M. De Domenico, A. Sole-Ribalta, E. Cozzo, M. Kivela, Y. Moreno, M. A. Porter, S. Gomez, and A. Arenas, *Phys. Rev. X* **3**, 041022 (2013).  
 [14] S. Bornholdt, *Science* **310**, 449 (2005).  
 [15] A. Pomerance, E. Ott, M. Girvan, and W. Losert, *Proc. Natl. Acad. Sci. USA* **106**, 8209 (2009).  
 [16] I. González-García, R. V. Solé, and J. Costa, *Proc. Natl. Acad. Sci. USA* **99**, 13085 (2002).



- [17] B. Derrida and Y. Pomeau, *Europhys. Lett.* **1**, 45 (1986).
- [18] M. Aldana, *Physica D* **185**, 45 (2003).
- [19] S. A. Kauffman, C. Peterson, B. Samuelsson, and C. Troein, *Proc. Natl. Acad. Sci. USA* **101**, 17102 (2004).
- [20] M. Rybarsch and S. Bornholdt, *Phys. Rev. E* **86**, 026114 (2012).
- [21] E. Cozzo, A. Arenas, and Y. Moreno, *Phys. Rev. E* **86**, 036115 (2012).
- [22] S. Squires, A. Pomerance, M. Girvan, and E. Ott, *Phys. Rev. E* **90**, 022814 (2014).
- [23] F. Morone and H. A. Makse, *Nature* **524**, 65 (2015).
- [24] S. Pei, L. Muchnik, J. S. Andrade, Z. Zheng, and H. A. Makse, *Sci. Rep.* **4**, 5547 (2014).
- [25] S. Pei, F. Morone, and H. A. Makse, *Spreading Dynamics in Social Systems* (Springer Nature, Berlin, 2017).
- [26] J.-X. Zhang, D.-B. Chen, Q. Dong, and Z.-D. Zhao, *Sci. Rep.* **6**, 27823 (2016).
- [27] M. Kitsak, L. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. Stanley, and H. Makse, *Nat. Phys.* **6**, 888 (2010).
- [28] R. Cohen, K. Erez, D. ben-Avraham, and S. Havlin, *Phys. Rev. Lett.* **86**, 3682 (2001).
- [29] S. Eubank, H. Guclu, V. A. Kumar, M. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang, *Nature* **429**, 180 (2004).
- [30] S. Eubank, V. S. Kumar, M. V. Marathe, A. Srinivasan, and N. Wang, in *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2004), pp. 718–727.
- [31] S. Pei and H. A. Makse, *J. Stat. Mech.* (2013) P12002.
- [32] T. Wang, J. Zhang, X. Sun, and S. Wandelt, *Europhys. Lett.* **118**, 68005 (2017).
- [33] B. Drossel, in *Reviews of Nonlinear Dynamics and Complexity*, edited by H. G. Schuster (Wiley, Weinheim, 2008) Vol. 1.
- [34] V. Kaufman, T. Mihaljev, and B. Drossel, *Phys. Rev. E* **72**, 046124 (2005).
- [35] F. Greil, *Front. Plant Sci.* **3**, 178 (2012).
- [36] U. Paul, V. Kaufman, and B. Drossel, *Phys. Rev. E* **73**, 026118 (2006).
- [37] X. Teng, S. Pei, F. Morone, and H. A. Makse, *Sci. Rep.* **6**, 36043 (2016).
- [38] S. Pei, X. Teng, J. Shaman, F. Morone, and H. A. Makse, *Sci. Rep.* **7**, 45240 (2017).
- [39] R. Zhang and S. Pei, *Chaos* **28**, 013103 (2017).
- [40] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web* (Stanford InfoLab, Stanford, CA, 1999).
- [41] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes, *Phys. Rev. Lett.* **96**, 040601 (2006).
- [42] J. Wang, B. Guo, W. Wei, Z. Mi, Z. Yin, and Z. Zheng, *Physica A* **481**, 70 (2017).
- [43] O. Angel, J. Friedman, and S. Hoory, *Trans. Am. Math. Soc.* **367**, 4287 (2015).
- [44] K. I. Hashimoto, in *Automorphic Forms and Geometry of Arithmetic Varieties*, Advanced Studies in Pure Mathematics Vol. 15 (Academic Press, Boston, MA, 1989), pp. 211–280.
- [45] C. R. MacCluer, *SIAM Rev.* **42**, 487 (2000).
- [46] N. Guelzim, S. Bottani, P. Bourguine, and F. Kepes, *Nat. Genet.* **31**, 60 (2002).
- [47] R. Dobrin, Q. K. Beg, A. L. Barabasi, and Z. N. Oltvai, *BMC Bioinform.* **5**, 10 (2004).
- [48] J. Coleman, E. Katz, and H. Menzel, *Sociometry* **20**, 253 (1957).
- [49] J. Moody, *Soc. Networks* **23**, 261 (2001).
- [50] R. Albert, H. Jeong, and A.-L. Barabási, *Nature* **406**, 378 (2000).
- [51] P. F. Bonacich, *J. Math. Sociol.* **2**, 113 (1972).
- [52] E. Ott and A. Pomerance, *Phys. Rev. E* **79**, 056111 (2009).
- [53] J. G. Restrepo, E. Ott, and B. R. Hunt, *Phys. Rev. Lett.* **97**, 094102 (2006).