Letter

Long-loop feedback vertex set and dismantling on bipartite factor graphs

Tianyi Li[1,2,*] Pan Zhang,[1,3,4,†] and Hai-Jun Zhou[1,5,6,‡]

[1]*CAS Key Laboratory for Theoretical Physics, Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing 100190, China*
[2]*System Dynamics Group, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02142, USA*
[3]*School of Fundamental Physics and Mathematical Sciences, Hangzhou Institute for Advanced Study, UCAS, Hangzhou 310024, China*
[4]*International Centre for Theoretical Physics Asia-Pacific, Beijing/Hangzhou, China*
[5]*School of Physical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China*
[6]*MinJiang Collaborative Center for Theoretical Physics, MinJiang University, Fuzhou 350108, China*

(Received 1 March 2021; revised 6 April 2021; accepted 29 May 2021; published 14 June 2021)

Network dismantling aims at breaking a network into disconnected components and attacking vertices that intersect with many loops has proven to be a most efficient strategy. Yet existing loop-focusing methods do not distinguish the short loops within densely connected local clusters (e.g., cliques) from the long loops connecting different clusters, leading to lowered performance of these algorithms. Here we propose a new solution framework for network dismantling based on a two-scale bipartite factor-graph representation, in which long loops are maintained while local dense clusters are simplistically represented as individual factor nodes. A mean-field spin-glass theory is developed for the corresponding long-loop feedback vertex set problem. The framework allows for the advancement of various existing dismantling algorithms; we developed the new version of two benchmark algorithms BPD (which uses the message-passing equations of the spin-glass theory as the solver) and CoreHD (which is fastest among well-performing algorithms). New solvers outperform current state-of-the-art algorithms by a considerable margin on networks of various sorts. Further improvement in dismantling performance is achievable by opting flexibly the choice of local clusters.

DOI: 10.1103/PhysRevE.103.L061302

Network dismantling is the optimization version of the celebrated site percolation problem. It aims at breaking a network into many disconnected components with the minimum number of vertex deletion [1–8]. Dismantling a network essentially means demolishing its long-range loops, and it is deeply connected to the concept of feedback vertex set (FVS, a set of vertices whose removal breaks all the loops in the network [1]). The dismantling problem is rooted in many structural and dynamical issues of network science, such as contagion spreading and vaccination [9,10], vital vertex identification [11,12], control of complex systems [13,14], and network robustness enhancement [15]. Many different heuristic algorithms were proposed over the years to solve this important problem [16–21]. Among them some best-performing algorithms (BPD [4,7], Min-Sum [5], and CoreHD [16,17]) are FVS-based iterative processes that delete those vertices considered most vital for loop integrity. The dismantling and the FVS problem are equivalent for synthetic random networks in which most loops are long range [1]; therefore the FVS strategy is optimal for dismantling such networks [4,5].

Triangles and other short-range loops are, however, abundant in real-world networks, leading to densely connected local clusters and communities [22–28], which are often preferably not to be split into parts. Deleting vertices that are highly involved in short-range loops within dense local regions also increases the dismantling cost [10,21]. It should be more advantageous to delete those articulation points linking different local clusters or communities [21,29], yet existing FVS-based methods so far do not differentiate short-range and long-range loops.

In this Letter, we design a long-loop FVS system and present an improved solution framework for network dismantling. Our model represents local dense clusters as *factors* in a two-scale bipartite factor-graph, intentionally overlooking loops within the specified clusters. Many short-range loops in the original network are thus absent in the constructed factor-graph, while long-range loops are preserved. By this means, solutions of message-passing techniques (e.g., BPD) would be near optimal after original clustered graphs are transferred to (quasi-) trees, an idea investigated similarly in recent attempts at other tasks [30–32]. In this work, we address the similar idea on the network dismantling problem. We derive coarse-grained (two-state) message-passing equations for this model to evaluate the impact of node removal to the factor graph, and use this equation set as a solver for network dismantling, which is the advanced version of BPD (termed as FBPD) with BPD reducing to it as a special case. Besides BPD, the new model allows for the advancement of various existing algorithms, and we further develop the corresponding version

*now at Department of Decision Sciences and Managerial Economics, CUHK Business School, Hong Kong, China; tianyi.li@cuhk.edu.hk
†panzhang@itp.ac.cn
‡zhouhj@itp.ac.cn

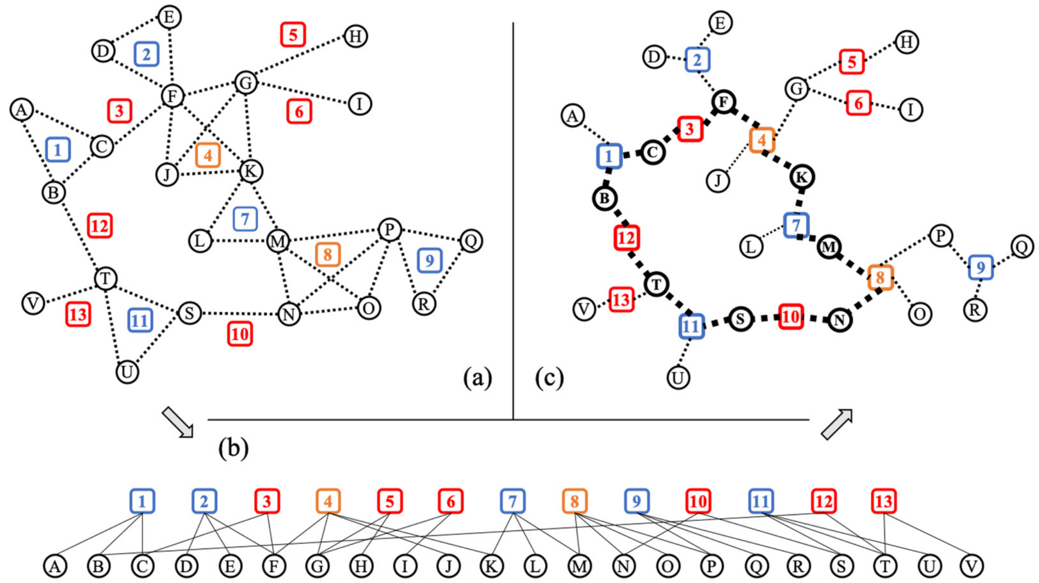

FIG. 1. Factor-graph construction for the long-loop feedback vertex set problem. This example network (a) contains 22 vertices (circles). We identify all maximal cliques as local clusters and represent them as factors (squares). The resulting bipartite factor-graph (b) has 13 factors and 22 vertices; a link between a vertex and a factor indicates that the vertex is a member of the cluster represented by the factor. Thick dotted links in (c) highlight the only intercluster long loop in the factor-graph.

of the CoreHD algorithm (termed as FCoreHD). We find that even when only the shortest loops in cliques of three and four vertices (i.e., triangles and tetrahedra) are overlooked, the factor-graph model can already lead to remarkable and consistent improvements in dismantling performance as compared with the conventional network model. Nevertheless, this solution framework is flexible and leaves room for further opting the design of the factor graph.

Factor-graph representation. Given a simple network G of N vertices and M undirected edges, which contains many densely connected local clusters or communities as well as long-range loops, our objective is to delete a minimum number of vertices so as to break long loops while preserving local loopy structures as much as possible. To minimize the distraction of short loops to long-loop breaking, we introduce a set of factors and expand the original network into a bipartite factor-graph, $\mathcal{G}$ (Fig. 1). Each factor (square in Fig. 1) a of $\mathcal{G}$ represents a cluster of G which is a densely connected local region with a high proportion of short loops, and a link (i, a) is drawn between every vertex (circle in Fig. 1) i of this cluster and the factor a to indicate that i is a member of cluster a. There is a huge literature on discovering clusters or communities in a complex network [28], and our theoretical framework is flexible to different criteria that might be adopted in determining clusters. A particularly straightforward recipe is to specify cliques of the graph as the clusters [25]. Notably, if an edge of G is not assigned to any such multivertex clusters, then the two incident vertices are regarded as forming a minimum-size cluster (i.e., a 2-clique). Clusters may partially overlap at some vertices; thus a vertex is generally connected to many factors in $\mathcal{G}$ (Fig. 1). Notice that all edges and loops within the identified clusters completely disappear in the factor-graph $\mathcal{G}$. All loops in $\mathcal{G}$, necessarily alternating between vertices and factors, correspond to intercluster (and often long) loops of the original graph G.

Model with local constraints. For a factor-graph $\mathcal{G}$ of vertices and factors, we define a subset Γ of its *vertices* as a feedback vertex set if the set intersects with every loop of $\mathcal{G}$ [1]. A minimum FVS (minFVS) is then a FVS of global minimum cardinality, which offers an optimal way of breaking all the intercluster loops and dismantling $\mathcal{G}$ into a collection of tree components [1]. However, the minFVS problem is an NP-hard combinatorial optimization problem and an exact solution is practically impossible [33]. Here we derive an approximate but efficient message-passing algorithm by extending an earlier spin-glass model [2]. In the following discussions we use symbols $i, j, k, \ldots$ to indicate vertices and $a, b, c, \ldots$ to indicate factors, and denote by ∂i (∂a) the set of nearest-neighboring factors (vertices) of vertex i (factor a).

First, a state c_i is introduced to each vertex i, which is either zero ($c_i = 0$, indicating i being inactive) or is equal to the index a of a neighboring factor (i being active, $c_i = a \in \partial i$). A configuration of the whole system is denoted as $\underline{c} \equiv (c_1, \ldots, c_N)$, and its energy is simply the total number of zero-state (inactive) vertices. Second, a local constraint χ_a is introduced for every factor a as

$$\chi_a = \prod_{i\in\partial a}\left(\delta_{c_i}^a + \delta_{c_i}^0\right) + \sum_{i\in\partial a}\left(1 - \delta_{c_i}^a - \delta_{c_i}^0\right)\prod_{j\in\partial a\backslash i}\left(\delta_{c_j}^a + \delta_{c_j}^0\right), \tag{1}$$

where $\delta_m^n = 1$ if $m = n$ and $\delta_m^n = 0$ otherwise, and $\partial a\backslash i$ means vertex i is excluded from set ∂a. Notice that $\chi_a = 1$ only if at most one of the neighboring vertices of factor a takes state other than 0 and a; otherwise, $\chi_a = 0$. We require $\boldsymbol{c}$ to satisfy $\chi_a = 1$ for all the factors a. Given such a valid configuration $\underline{c}$, each connected subgraph of $\mathcal{G}$ formed by the active vertices and the attached factors is then most often a tree (free of any loop) or occasionally a cycle-tree (containing exactly one loop) [2]. If cycle-trees do exist, then we can easily break each of the associated loops by inactivating a single vertex, and then all the inactivated vertices of $\underline{c}$ form a FVS. Conversely, each FVS Γ of $\mathcal{G}$ can be mapped to a set of valid configurations $\underline{c}$ after setting $c_i = 0$ for every vertex $i \in \Gamma$ [2].

We define the partition function $Z(\beta)$ of the system as

$$Z(\beta) = \sum_{\underline{c}} \prod_i \left[1 + (e^{-\beta} - 1)\delta^0_{c_i}\right] \prod_a \chi_a, \tag{2}$$

where β is the inverse temperature parameter. At large values of β, the partition function is predominantly contributed by minFVS configurations and their low-energy excitations.

Belief-propagation (BP). We solve the spin glass model (2) by the now standard replica-symmetric cavity method [34,35]. First, the probability of vertex i taking state c_i in the absence of the constraint from neighboring factor a, denoted as $q^{c_i}_{i\to a}$, is estimated via the following self-consistent BP equations

$$q^0_{i\to a} \propto e^{-\beta} \prod_{b\in\partial i\backslash a} \left[1 + \sum_{j\in\partial b\backslash i} \left(\frac{1}{q^b_{j\to b} + q^0_{j\to b}} - 1\right)\right], \tag{3a}$$

$$q^b_{i\to a} \propto 1 + \sum_{j\in\partial b\backslash i} \left(\frac{1}{q^b_{j\to b} + q^0_{j\to b}} - 1\right), \quad (b \in \partial i\backslash a), \tag{3b}$$

$$q^a_{i\to a} \propto 1, \tag{3c}$$

with normalization $q^0_{i\to a} + q^a_{i\to a} + \sum_{b\in\partial i\backslash a} q^b_{i\to a} = 1$, where $\partial i\backslash a$ means excluding factor a from set ∂i. In the numerical implementation we need only to iterate the coarse-grained probability $q^{a+0}_{i\to a} \equiv q^a_{i\to a} + q^0_{i\to a}$, rendering the minFVS problem essentially an Ising-type spin glass system. Second, the marginal probability of vertex i taking state $c_i = 0$ (i.e., being a feedback vertex) under the constraints of all its neighboring factors a, denoted as q^0_i, is estimated via

$$q^0_i = \frac{e^{-\beta} \prod_{a\in\partial i} \left[1 + \sum_{j\in\partial a\backslash i} \left(\frac{1}{q^a_{j\to a} + q^0_{j\to a}} - 1\right)\right]}{e^{-\beta} \prod_{a\in\partial i} \left[1 + \sum_{j\in\partial a\backslash i} \left(\frac{1}{q^a_{j\to a} + q^0_{j\to a}} - 1\right)\right] + \sum_{a\in\partial i} \left[1 + \sum_{j\in\partial a\backslash i} \left(\frac{1}{q^a_{j\to a} + q^0_{j\to a}} - 1\right)\right]}. \tag{4}$$

The mean fraction ρ of feedback vertices at inverse temperature β is then

$$\rho = \frac{1}{N} \sum_i q^0_i. \tag{5}$$

Explicit expressions for the free energy density and the entropy density can be derived and be evaluated at a BP fixed point [36]. We are then able to estimate the minFVS relative size by taking the $\beta \to \infty$ of ρ (if the entropy density is non-negative at this limit) or by taking the value of ρ at the maximal value of β at which the entropy density reaches zero from above [2].

Similarly to the iterative process of the BPD algorithm, we can try to construct a quasiminimum FVS for the factor-graph $\mathcal{G}$ by deleting at each BP iteration a tiny fraction of vertices i of $\mathcal{G}$ whose estimated inactive probability q^0_i are the highest among remaining vertices, until all the loops in $\mathcal{G}$ are broken. We refer to this factor-graph decimation process as FBPD.

Results on random and real networks. We test the performance of FBPD on random networks G formed by local n-cliques (i.e., fully connected n-vertex subnetworks). Each vertex participates in exactly K randomly chosen n-cliques and the graph is otherwise completely random [37–41]. The total number of n-cliques and edges is NK/n and $M = NK(n-1)/2$. All vertices are involved in both intraclique short loops and interclique long loops. Naturally we represent each n-clique as a factor, getting a factor-graph $\mathcal{G}$ that retains only long loops and is locally treelike. Quantitative results obtained on the cases of $n = 3$ (triangle clusters) and $n = 4$ (tetrahedron clusters) are shown in Fig. 2. We find that, at a given value of K, the fraction ρ of long-loop feedback vertices achieved by FBPD on individual network instances is only slightly exceeding the theoretical minFVS relative size $\rho_{\min}$ predicted by the replica-symmetric mean-field theory. For example at $K = 10$ and $n = 3$, $\rho_{\min} = 0.6932$ while the FBPD empirical values are $\rho \approx 0.7080$ for graphs of size $N = 10^5$ ($\beta = 7$ for FBPD, and algorithmic results are insensitive to this parameter).

Besides the message-passing FBPD, the factor-graph representation allows for the advancement of other long-loop breaking heuristics. All loops of a network G are contained in its 2-core, the maximum subnetwork in which every vertex is connected to at least two other vertices of this subnetwork. Similarly, the 2-core of the corresponding factor-graph $\mathcal{G}$ contains all the long loops of G. A simple and fast algorithm to destroy the 2-core of G is CoreHD, which recursively removes a highest-degree vertex of the extant 2-core [16,17]. We tested the factor-graph version of CoreHD (termed as FCoreHD)

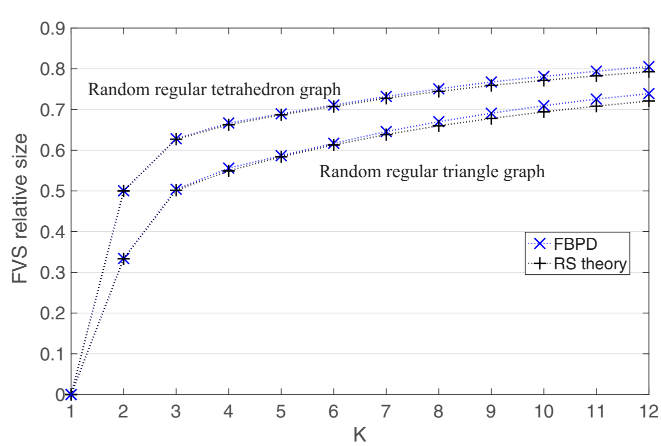

FIG. 2. The minimum fraction of long-loop feedback vertices for random regular triangle networks (lower) and tetrahedron networks (upper). K is the number of triangles or tetrahedra participated by each vertex. The FBPD algorithmic results obtained on single network instances of $N = 100\,002$ (crosses) are only slightly higher than predictions from the replica-symmetric mean-field theory (pluses).

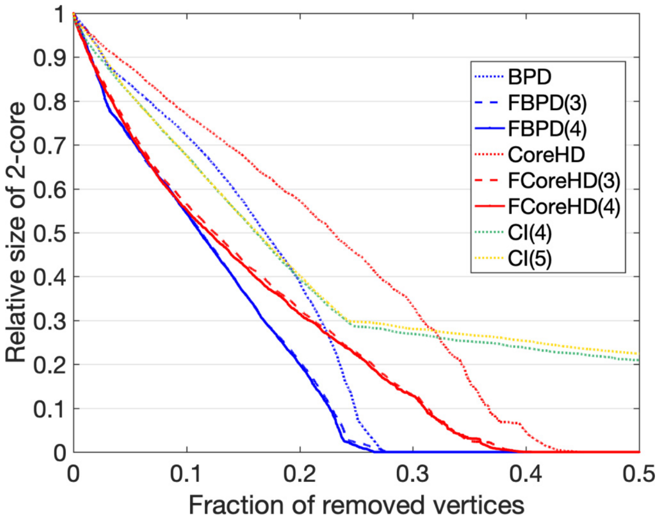

FIG. 3. Shrinkage of the 2-core of the LastFM network (7624 vertices, 27 806 edges) along the sequential removal of vertices. The FBPD(n) and FCoreHD(n) trajectories are obtained on the factor-graph $\mathcal{G}_n$ considering cliques up to size n (3 or 4). Trajectories from BPD, CoreHD, and CI (level 4 and 5) algorithms are compared against.

TABLE I. Dismantling results of real-world networks. N, network size; n, number of vertices deleted by the algorithm. Superscripts B, FB, C, FC indicate BPD, FBPD, CoreHD, FCoreHD; subscripts 3 and 4 indicate the maximum cliques considered (triangles and tetrahedra, respectively).

Network	N	n^B	n_3^{FB}	n_4^{FB}	n^C	n_3^{FC}	n_4^{FC}
Email1	986	456	454	455	**452**	458	**452**
Road	1174	148	149	149	149	**147**	**147**
Yeast	2284	357	351	354	353	**350**	355
FBK-a	4039	1907	1921	**1872**	1914	1892	**1872**
Grid	4941	312	**301**	304	317	302	305
Wiki	5201	1148	1138	**1134**	1150	1149	1145
IntNet	6474	160	159	159	160	161	**156**
LastFM	7624	1285	1262	**1258**	1296	1296	1276
Twitch	9498	3013	3011	**2992**	3036	3037	3021
FBK-b	13 866	3311	3295	**3293**	3356	3332	3302
Cite	34 546	13 433	**13 393**	13 399	13 547	13 484	13 475
Email2	36 692	**2616**	2626	2647	2621	2648	2647
Github	37 700	6468	**6422**	6423	6554	6520	6495
Deezer	54 573	23 805	23 751	**23 686**	24 180	24 098	24 112
BK	58 228	6225	6140	**6130**	6257	6213	6207
Amazon	262 111	45 038	44 862	**44 438**	45 612	44 915	44 803

on random clustered networks. FCoreHD's performance is inferior to FBPD but clearly outperforms CoreHD. Similar results are observed on real networks. The 2-core of a real network will generally shrink much faster if its vertices are sequentially deleted according to the factor-graph algorithms (FBPD, FCoreHD) than according to the graph algorithms (BPD, CoreHD). Figure 3 offers a concrete demonstration, where the corresponding trajectories from the CI algorithm (level 4 and 5) are also compared against.

We now compare the performance of factor-graph (FBPD, FCoreHD) and graph (BPD, CoreHD) solvers on network dismantling. Algorithms first delete vertices to solve the minFVS problem and then reinsert some vertices to the network under the constraint that any remaining connected component contains at most $0.01N$ vertices [3–5,16,42]. Real-world networks of miscellaneous sorts are used in experiments, including road network [27], power grid [22], internet (IntNet) [43], email networks (Email1/2) [26,44], protein interaction network (Yeast) [45], product co-purchasing network (Amazon) [46], Wikipedia network (Wiki) [47], citation network (Cite) [43], and social networks on various platforms (LastFM, Github, Twitch, Facebook networks FBK-a/b, Deezer, Brightkite BK) [47–51]. For each network G, we adopt two factor-graph versions ($\mathcal{G}_3$ and $\mathcal{G}_4$). $\mathcal{G}_3$ is obtained by repeatedly picking a 3-clique at random from G as a factor and then deleting all the edges of this 3-clique; when there is no more 3-cliques then we continue with 2-cliques (the conventional edges). $\mathcal{G}_4$ is similarly constructed while starting with 4-cliques, and therefore more loops are treated as intracluster loops. Results listed in Table I confirm the improvement of factor-graph algorithms over graph algorithms. It is observed that FBPD in general claims a substantial margin over the other three algorithms. In many cases, the performance of FBPD improves as more short loops are replaced by factors (e.g., using $\mathcal{G}_n$ with larger values of n). We also tested the MinSum algorithm on these networks and results confirm previous studies [4,16], suggesting that MinSum's overall performance is comparable to BPD but is much slower in speed than BPD/FBPD, while all of them are slower than CoreHD/FCoreHD by a few orders of magnitude.

Conclusion. The factor-graph representation overlooks short-range structures in a complex network and allows us to rank vertices according to their contribution to long-range loops. As one important application, we demonstrated that factor-graph FVS algorithms (FBPD, FCoreHD) considerably outperform corresponding graph FVS algorithms (BPD, CoreHD) in dismantling real networks, even when only the shortest loops in 3-cliques and 4-cliques were overlooked. This representation also enables the computation of the ensemble-averaged size of long-loop minimum feedback vertex sets for random clustered networks.

In the present work we consider the straightforward way of constructing factor-graphs via cliques. Opting the choice of factors to achieve an optimal factor graph for a given real-wold network instance is an interesting open issue, insightful for future studies especially for domain applications. Another future direction is to extend the factor-graph framework to directed networks and consider the more difficult problem of breaking all long-range directed cycles.

Acknowledgments. We sincerely thank the two anonymous reviewers for the recognition of this study and the valuable comments. T.L. thanks the Institute of Theoretical Physics of the Chinese Academy of Sciences (ITP-CAS) for hospitality. This work was supported by the National Natural Science Foundation of China Grants No. 11975295, No. 11947302, No. 11975294, and No. 12047503, and the Chinese Academy of Sciences Grant No. QYZDJ-SSW-SYS018 and QYZDB-SSW-SYS032. Numerical simulations were carried out at the Tianwen and HPC clusters of ITP-CAS and the Tianhe-2 platform of the National Supercomputer Center in Guangzhou.

[1] P. Haxell, O. Pikhurko, and A. Thomason, Maximum acyclic and fragmented sets in regular graphs, J. Graph Theory **57**, 149 (2008).
[2] H.-J. Zhou, Spin glass approach to the feedback vertex set problem, Eur. Phys. J. B **86**, 455 (2013).
[3] F. Morone and H. Makse, Influence maximization in complex networks through optimal percolation, Nature (Lond.) **524**, 65 (2015).
[4] S. Mugisha and H.-J. Zhou, Identifying optimal targets of network attack by belief propagation, Phys. Rev. E **94**, 012305 (2016).
[5] A. Braunstein, L. Dall'Asta, L. Semerjian, and G. Zdeborová, Network dismantling, Proc. Natl. Acad. Sci. USA **113**, 12368 (2016).
[6] P. Clusella, P. Grassberger, F. J. Pérez-Reche, and A. Politi, Immunization and Targeted Destruction of Networks Using Explosive Percolation, Phys. Rev. Lett. **117**, 208301 (2016).
[7] S.-M. Qin, Spin-glass model for the c-dismantling problem, Phys. Rev. E **98**, 062309 (2018).
[8] X.-L. Ren, N. Gleinig, D. Helbing, and N. Antulov-Fantulin, Generalized network dismantling, Proc. Natl. Acad. Sci. USA **116**, 6554 (2019).
[9] F. Altarelli, A. Braunstein, L. Dall'Asta, and R. Zecchina, Optimizing spread dynamics on graphs by message passing, J. Stat. Mech. (2013) P09011.
[10] Y. Chen, G. Paul, S. Havlin, F. Liljeros, and H. E. Stanley, Finding a Better Immunization Strategy, Phys. Rev. Lett. **101**, 058701 (2008).
[11] L. Lü, D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, and T. Zhou, Vital nodes identification in complex networks, Phys. Rep. **650**, 1 (2016).
[12] Ş. Erkol, C. Castellano, and F. Radicchi, Systematic comparison between methods for the detection of influential spreaders in complex networks, Sci. Rep. **9**, 15095 (2019).
[13] Y.-Y. Liu and A.-L. Barabási, Control principles of complex systems, Rev. Mod. Phys. **88**, 035006 (2016).
[14] A. Y. Lokhov and D. Saad, Optimal deployment of resources for maximizing impact in spreading processes, Proc. Natl. Acad. Sci. USA **114**, E8138 (2017).
[15] M. Chujyo and Y. Hayashi, A loop enhancement strategy for network robustness, Appl. Netw. Sci. **6**, 3 (2021).
[16] L. Zdeborová, P. Zhang, and H.-J. Zhou, Fast and simple decycling and dismantling of networks, Sci. Rep. **6**, 37954 (2016).
[17] J.-H. Zhao, Y. Habibulla, and H.-J. Zhou, Statistical mechanics of the minimum dominating set problem, J. Stat. Phys. **159**, 1154 (2015).
[18] S. Wandelt, X. Sun, D. Feng, M. Zanin, and S. Havlin, A comparative analysis of approaches to network-dismantling, Sci. Rep. **8**, 13513 (2018).
[19] C. Fan, L. Zeng, Y. Sun, and Y.-Y. Liu, Finding key players in complex networks through deep reinforcement learning, Nat. Mach. Intell. **2**, 317 (2020).
[20] D. Zhao, S. Yang, X. Han, S. Zhang, and Z. Wang, Dismantling and vertex cover of network through message passing, IEEE Trans. Circ. Syst. II: Expr. Biref. **67**(11), 2732 (2020).
[21] S. Wandelt, X. Shi, X. Sun, and M. Zanin, Community detection boosts network dismantling on real-world networks, IEEE Access **8**, 111954 (2020).
[22] D. J. Watts and S. H. Strogatz, Collective dynamics of 'small-world' networks, Nature **393**, 440 (1998).
[23] M. Girvan and M. E. J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA **99**, 7821 (2002).
[24] M. E. J. Newman, Properties of highly clustered networks, Phys. Rev. E **68**, 026121 (2003).
[25] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature (Lond.) **435**, 814 (2005).
[26] 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. **6**, 29 (2009).
[27] L. Šubelj and M. Bajec, Robust network community detection using balanced propagation, Eur. Phys. J. B **81**, 353 (2011).
[28] S. Fortunato and D. Hric, Community detection in networks: A user guide, Phys. Rep. **659**, 1 (2016).
[29] L. Tian, A. Bashan, D.-N. Shi, and Y.-Y. Liu, Articulation points in complex networks, Nat. Commun. **8**, 14223 (2017).
[30] M. E. J. Newman, Spectra of networks containing short loops, Phys. Rev. E **100**, 012314 (2019).
[31] G. T Cantwell and M. E. J. Newman, Message passing on networks with loops, Proc. Natl. Acad. Sci. USA **116**, 23398 (2019).
[32] A. Kirkley, G. T. Cantwell, and M. E. J. Newman, Belief propagation for networks with loops, Sci. Adv. **7**, eabf1211 (2021).
[33] F. V. Fomin, S. Gaspers, A. V. Pyatkin, and I. Razgon, On the minimum feedback vertex set problem: Exact and enumeration algorithms, Algorithmica **52**, 293 (2008).
[34] M. Mézard and A. Montanari, *Information, Physics, and Computation* (Oxford University Press, Oxford, UK, 2009).
[35] H.-J. Zhou, *Spin Glass and Message Passing* (Science Press, Beijing, 2015).
[36] T. Li, P. Zhang, and H.-J. Zhou (unpublished).
[37] J. C. Miller, Percolation and epidemics in random clustered networks, Phys. Rev. E **80**, 020901(R) (2009).
[38] M. E. J. Newman, Random Graphs with Clustering, Phys. Rev. Lett. **103**, 058701 (2009).
[39] J. P. Gleeson, Bond percolation on a class of clustered random networks, Phys. Rev. E **80**, 036107 (2009).
[40] S. Yoon, A. V. Goltsev, S. N. Dorogovtsev, and J. F. F. Mendes, Belief-propagation algorithm and the ising model on networks with arbitrary distributions of motifs, Phys. Rev. E **84**, 041144 (2011).
[41] P. Zhang, Spectral estimation of the percolation transition in clustered networks, Phys. Rev. E **96**, 042303 (2017).
[42] Z. Wang, C. Sun, G. Yuan, X. Rui, and X. Yang, A neighborhood link sensitive dismantling method for social networks, J. Comput. Sci. **43**, 101129 (2020).
[43] J. Leskovec, J. Kleinberg, and C. Faloutsos, Graphs over time: Densification laws, shrinking diameters and possible explanations, in *Proceedings of the 11st ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (2005), pp. 177–187.
[44] J. Leskovec, J. Kleinberg, and C. Faloutsos, Graph evolution: Densification and shrinking diameters, ACM Trans. Knowl. Discov. Data **1**, 2–es (2007).
[45] D. Bu, Y. Zhao, L. Cai, H. Xue, X. Zhu, H. Lu, J. Zhang, S. Sun, L. Ling, N. Zhang, G. Li, and R. Chen, Topological structure

analysis of the protein–protein interaction network in budding yeast, Nucleic Acids Res. **31**, 2443 (2003).

[46] J. Leskovec, L. A. Adamic, and B. A. Huberman, The dynamics of viral marketing, ACM Trans. Web **1**, 5–es (2007).

[47] B. Rozemberczki, C. Allen, and R. Sarkar, Multi-scale attributed node embedding, J. Complex Netw., **9**(2), cnab014 2021.

[48] B. Rozemberczki and R. Sarkar, Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models, in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management* (2020), pp. 1325–1334.

[49] J. J. McAuley and J. Leskovec, Learning to discover social circles in ego networks, NIPS 2012, 548 (2012).

[50] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton, Gemsec: Graph embedding with self clustering, in *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (2019), pp. 65–72.

[51] S. A. Cho, E.and Myers and J. Leskovec, Friendship and mobility: user movement in location-based social networks, in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2011), pp. 1082–1090.