POLITECNICO DI TORINO
Repository ISTITUZIONALE

Network dismantling

Original
Network dismantling / Braunstein, A., Dall'Asta, L., Semerjian, G., Zdeborová, L.. - In: PROCEEDINGS OF THE NATIONAL ACADEMY OF SCIENCES OF THE UNITED STATES OF AMERICA. - ISSN 0027-8424. - 113:44(2016), pp. 12368-12373. [10.1073/pnas.1605083113]

Availability:
This version is available at: 11583/2656832 since: 2016-11-22T08:56:27Z

Publisher:
National Academy of Sciences

Published
DOI:10.1073/pnas.1605083113

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

13 November 2026

Network dismantling

Alfredo Braunstein[a,b,c], Luca Dall'Asta[a,c], Guilhem Semerjian[d], and Lenka Zdeborová[e,1]

[a]Department of Applied Science and Technology, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Turin, Italy; [b]Human Genetics Foundation, Via Nizza 52, 10126 Turin, Italy; [c]Collegio Carlo Alberto, Via Real Collegio 30, 10024 Moncalieri, Italy; [d]Laboratoire de Physique Théorique, Ecole Normale Supérieure, Paris Sciences et Lettres (PSL) Research University, Sorbonne Universités, Université Pierre et Marie Curie (UPMC), CNRS, 75005 Paris, France; and [e]Institut de Physique Théorique, CNRS, Commissariat à l'Energie Atomique (CEA) and Université Paris Saclay, 91191 Gif-sur-Yvette, France

Edited by Giorgio Parisi, University of Rome, Rome, Italy, and approved September 16, 2016 (received for review March 29, 2016)

We study the network dismantling problem, which consists of determining a minimal set of vertices in which removal leaves the network broken into connected components of subextensive size. For a large class of random graphs, this problem is tightly connected to the decycling problem (the removal of vertices, leaving the graph acyclic). Exploiting this connection and recent works on epidemic spreading, we present precise predictions for the minimal size of a dismantling set in a large random graph with a prescribed (light-tailed) degree distribution. Building on the statistical mechanics perspective, we propose a three-stage Min-Sum algorithm for efficiently dismantling networks, including heavy-tailed ones for which the dismantling and decycling problems are not equivalent. We also provide additional insights into the dismantling problem, concluding that it is an intrinsically collective problem and that optimal dismantling sets cannot be viewed as a collection of individually well-performing nodes.

graph fragmentation | message passing | percolation | random graphs | influence maximization

A network (a graph G in the discrete mathematics language) is a set V of N entities called nodes (or vertices), along with a set E of edges connecting some pairs of nodes. In a simplified way, networks are used to describe numerous systems in very diverse fields, ranging from social sciences to information technology or biological systems (reviews are in refs. 1 and 2). Several crucial questions in the context of network studies concern the modifications of the properties of a graph when a subset S of its nodes is selected and treated in a specific way. For instance, how much does the size of the largest connected component of the graph decrease if the vertices in S (along with their adjacent edges) are removed? Do the cycles survive this removal? What is the outcome of the epidemic spreading if the vertices in S are initially contaminated, constituting the seed of the epidemic? On the contrary, what is the influence of a vaccination of nodes in S preventing them from transmitting the epidemic? It is relatively easy to answer these questions when the set S is chosen randomly, with each vertex being selected with some probability independently. Classical percolation theory is nothing but the study of the connected components of a graph in which some vertices have been removed in this way.

A much more interesting case is when the set S can be chosen in some optimal way. Indeed, in all applications sketched above, it is reasonable to assign some cost to the inclusion of a vertex in S: vaccination has a socioeconomic price, incentives must be paid to customers to convince them to adopt a new product in a viral marketing campaign, and incapacitating a computer during a cyber attack requires resources. Thus, one faces a combinatorial optimization problem: the minimization of the cost of S under a constraint on its effect on the graph. These problems thus exhibit both static and dynamic features, the former referring to the combinatorial optimization aspect and the latter referring to the definition of the cost function itself through a dynamical process.

In this paper, we focus on the existence of a giant component in a network: that is, the largest component containing a positive fraction of the vertices (in the $N \to \infty$ limit). On the one hand, the existence of a giant component is often necessary for the network to fulfill its function (e.g., to deliver electricity or information bits or ensure possibility of transportation). An adversary might be able to destroy a set of nodes with the goal of destroying this functionality. It is thus important to understand what an optimal attack strategy is, possibly as a first step in the design of optimal defense strategies. On the other hand, a giant component can propagate an epidemic to a large fraction of a population of nodes. Interpreting the removal of nodes as the vaccination of individuals who cannot transmit the epidemic anymore, destroying the giant component can be seen as an extreme way of organizing a vaccination campaign (3, 4) by confining the contagion to small connected components [less drastic strategies can be devised using specific information about the epidemic propagation model (5, 6)]. Another related application is influence maximization as studied in many previous works (7–9). In particular, optimal destruction of the giant component is equivalent to selection of the smallest set of initially informed nodes needed to spread the information into the whole network under a special case of the commonly considered model for information spreading (7–9).

To define the main subject of this paper more formally, following ref. 10, we call S a C-dismantling set if its removal yields a graph with the largest component that has size (in terms of its number of nodes) at most C. The C-dismantling number of a graph is the minimal size of such a set. When the value of C is either clear from the context or not important for the given claim, we will simply talk about dismantling. Typically, the size of the largest component is a finite fraction of the total number of nodes N. To formalize the notion of destroying the giant component, we will consider the bound C on the size of the connected components of the dismantled network to be such that $1 \ll C \ll N$. It should be noted that we defined dismantling in terms of node removal; it could be rephrased in terms of edge removal (11), which turns out to be a much easier problem. The dismantling problem is also referred to as fragmentability of graphs in graph theory literature (12–14) and optimal percolation in ref. 15.

Significance

Many systems of interest can be represented by a network of nodes connected by edges. In many circumstances, the existence of a giant component is necessary for the network to fulfill its function. Motivated by the need to understand optimal attack strategies, optimal spread of information, or immunization policies, we study the network dismantling problem (i.e., the search for a minimal set of nodes in which removal leaves the network broken into components of subextensive size). We give the size of the optimal dismantling set for random networks, propose an efficient dismantling algorithm for general networks that outperforms by a large margin existing strategies, and provide various insights about the problem.

Author contributions: A.B., L.D., G.S., and L.Z. designed research, performed research, contributed new reagents/analytic tools, analyzed data, and wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

Freely available online through the PNAS open access option.

[1]To whom correspondence should be addressed. Email: lenka.zdeborova@cea.fr.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1605083113/-/DCSupplemental.

Determining whether the C-dismantling number of a graph is smaller than some constant is nondeterministic polynomial (NP)-complete decision problem (a proof is in *SI Appendix*). The concept of NP completeness concerns the worst case difficulty of the problem. The questions that we address in this paper are instead the following. What is the dismantling number on some representative class of graphs (in our case, random graphs)? What are the best heuristic algorithms, how does their performance compare with the optimum, and how do they perform on benchmarks of real world graphs? Simple heuristic algorithms for the dismantling problem were considered in previous works (16–18), where the choice of the nodes to be included in the dismantling set was based on their degrees (favoring the inclusion of the most connected vertices) or some measure of their centrality. More recently, a heuristic for the dismantling problem has been presented in ref. 15 under the name "collective" influence (CI), in which the inclusion of a node is decided according to a combination of its degree and the degrees of the nodes in a local neighborhood around it. Ref. 15 also attempts to estimate the dismantling number on random graphs.

Our Main Contribution

In this paper, we provide a detailed study of the dismantling problem, with both analytical and algorithmic outcomes. We present very accurate estimates of the dismantling number for large random networks, building on a connection with the decycling problem [in which one seeks a subset of nodes with removal that leaves the graph acyclic; also an NP-complete problem (19)] and recent studies of optimal spreading (20–23). Our results are the one-step replica symmetry broken estimate of the ground state of the corresponding optimization problem.

On the computational side, we introduce a very efficient algorithm that outperforms considerably state of the art algorithms for solving the dismantling problem. We show its efficiency and closeness to optimality on both random graphs and real world networks. The goal of our paper is closely related to the one of ref. 15; we present an assessment of the results reported therein on random as well as real world networks.

Our dismantling algorithm, which has been inspired by the theoretical insight gained on random graphs, is composed of three stages.

i) Min-Sum message passing for decycling. This part is the core of the algorithm, using a variant of a message-passing algorithm developed in refs. 20 and 21. A related but different message-passing algorithm was developed for decycling in ref. 22 and later applied to dismantling in ref. 24; it performs comparably with ours.

ii) Tree breaking. After all cycles are broken, some of the tree components may still be larger than the desired threshold C. We break them into small components, removing a fraction of nodes that vanishes in the large size limit. This operation can be done in time $O(N \log N)$ by an efficient greedy procedure (detailed in *SI Appendix*).

iii) Greedy reintroduction of cycles. As explained below, the strategy of first decycling a graph before dismantling it is the optimal one for graphs that contain few short cycles (a typical property of light-tailed random graphs). For graphs with many short cycles, we improve considerably the efficiency of our algorithm by reinserting greedily some nodes that close cycles without increasing too much the size of the largest component.

The dismantling problem, as is often the case in combinatorial optimization, exhibits a very large number of (quasi)optimal solutions. We characterize the diversity of these degenerate minimal dismantling sets by a detailed statistical analysis, computing in particular the frequency of appearance of each node in the quasioptimal solutions, and conclude that dismantling is an intrinsically collective phenomenon that results from a correlated choice of a finite fraction of nodes. It thus makes much more sense to think in terms of good dismantling sets as a whole and not about individual nodes as the optimal influencers/spreaders (15). We further study the correlation between the degree of a node and its importance for dismantling, exploiting a natural variant of our algorithm, in which the dismantling set is required to avoid some marked nodes. This study allows us to show that each of the low-degree nodes can be replaced by other nodes without increasing significantly the size of the dismantling set. Contrary to claims in ref. 15, we do not confirm any particular importance of weak nodes, apart from the obvious fact that the set of highest-degree nodes is not a good dismantling set.

To give a quantitative idea of our algorithmic contribution, we state two representative examples of the kind of improvement that we obtain with the above algorithm with respect to the state of the art (15).

i) In an Erdős–Rényi (ER) random graph of average degree 3.5 and size $N = 5^7$, we found $C = 1{,}000$ dismantling sets removing 17.8% of the nodes, whereas the best known method (adaptive eigenvalue centrality for this case) removes 20.2% of the nodes, and the adaptive CI method of ref. 15 removes 20.6% of the nodes. Hence, we provide a 13% improvement over the state of the art. Our theoretical analysis estimates the optimum dismantling number to be around 17.5% of the nodes; thus, the algorithm is extremely close to optimal in this case.

ii) Our algorithm managed to dismantle the Twitter network studied in ref. 15 (with 532,000 nodes) into components smaller than $C = 1{,}000$ using only 3.4% of the nodes, whereas the CI heuristics of ref. 15 needs 5.6% of the nodes. Here, we thus provide a 60% improvement over the state of the art.

Not only does our algorithm show beyond state of the art performance, but it is also computationally efficient. Its core part runs in linear time over the number of edges, allowing us to easily dismantle networks with tens of millions of nodes.

Relation Between Dismantling and Decycling

We begin our discussion by clarifying the relation between the dismantling and decycling problems. Although the argument below can be found in ref. 10, we reproduce it here in a simplified fashion. The decycling number (or more precisely, fraction) $\theta_{\rm dec}(G)$ of G is the minimal fraction of vertices that have to be removed to make the graph acyclic. We define similarly the dismantling number $\theta_{\rm dis}(G, C)$ of a graph G as the minimal fraction of vertices that have to be removed to make the size of the largest component of the remaining graph smaller than a constant C.

For random graphs with degree distribution $q = \{q_k\}_{k\geq 0}$, in the large size limit, the parameters $\theta_{\rm dec}$ and $\theta_{\rm dis}$ will enjoy concentration (self-averaging) properties; we shall thus write their typical values as

$$\theta_{\rm dec}(q) = \lim_{N\to\infty} \mathbb{E}[\theta_{\rm dec}(G)], \qquad [1]$$

$$\theta_{\rm dis}(q) = \lim_{C\to\infty} \lim_{N\to\infty} \mathbb{E}[\theta_{\rm dis}(G, C)], \qquad [2]$$

where $\mathbb{E}[\bullet]$ denotes an average over the random graph ensemble. For the dismantling number, we allow the connected components after the removal of a dismantling set to be large but subextensive because of the order of limits. It was proven in ref. 10 that, for some families of random graphs, an equivalent definition is $\lim_{\epsilon\to 0}\lim_{N\to\infty}\mathbb{E}[\theta_{\rm dis}(G, \epsilon N)]$ (i.e., connected components are allowed to be extensive but with a vanishing intensive size).

The crucial point for the relation between dismantling and decycling is that trees (or more generically, forests) can be efficiently dismantled. It was proven in ref. 10 that $\theta_{\rm dis}(G, C) \leq 1/(C+1)$ whenever G is a forest. This inequality means that the fraction of vertices to be removed from a forest to dismantle it into components of size C goes to zero when C grows.

This observation brings us to the following two claims concerning the dismantling and decycling numbers for random graphs with degree distribution q: (*i*) for any degree distribution, $\theta_{\rm dis}(q) \leq \theta_{\rm dec}(q)$; and (*ii*) if q also admits a second moment (we

Table 1. The (1RSB) cavity predictions for the decycling number of ER random graphs of average degree d and the decycling number reached by the Min-Sum algorithm on graphs of size $N = 10^7$ nodes

d	$\theta_{\mathrm{dec}}(d)$	$\theta^{\mathrm{MS}}_{\mathrm{dec}}(d)$
1.5	0.0125	0.0135
2.5	0.0912	0.0936
3.5	0.1753	0.1782
5	0.2789	0.2823

shall call q light tailed when this is the case), then there is actually an equality between these two parameters, $\theta_{\mathrm{dis}}(q) = \theta_{\mathrm{dec}}(q)$.

The first claim follows directly from the above observation on the decycling number of forests. After a decycling set S of G has been found, one can add to S additional vertices to turn it into a C-dismantling set, the additional cost being bounded as $\theta_{\mathrm{dis}}(G, C) \leq \theta_{\mathrm{dec}}(G) + 1/(C+1)$. Taking averages of this bound and the limit $C \to \infty$ after $N \to \infty$ yields directly *i*.

To justify our second claim, we consider a C-dismantling set S of a graph G. To turn S into a decycling set, we need to add additional vertices to break the cycles that might exist in $G \backslash S$. The lengths of these cycles are certainly smaller than C, and removing at most one vertex per cycle is enough to break them. We can thus write $\theta_{\mathrm{dec}}(G) \leq \theta_{\mathrm{dis}}(G, C) + n_C(G)/N$, with $n_C(G)$ denoting the number of cycles of G of length at most C. We recall that the existence of a second moment of q implies that $n_C(G)$ remains bounded when $N \to \infty$ with C fixed. Considering the limit $N \to \infty$ and property *i*, property *ii* follows.

Network Decycling

In this section, we shall explain the results on the decycling number of random graphs that we obtained via statistical mechanics methods and how they can be exploited to build an efficient heuristic algorithm for decycling arbitrary graphs.

Testing the Presence of Cycles in a Graph. The 2-core of a graph G is its largest subgraph of minimal degree 2; it can be constructed by iteratively removing isolated nodes and leaves (vertices of degree 1) until either all vertices have been removed or all remaining vertices have degree at least 2. It is easy to see that a graph contains cycles if and only if its 2-core is nonempty. To decide if a subset S is decycling, we remove the nodes in S and perform this leaf removal on the reduced graph. To formalize this procedure, we introduce binary variables $x_i^t(S) \in \{0,1\}$ on each vertex $i \in V$ of the graph, t being a discrete time index. At the starting time $t = 0$, one marks the initially removed vertices by setting $x_i^0(S) = 1$ if $i \in S$ and 0 otherwise, and let the x variables evolve in time according to

$$x_i^{t+1}(S) = \begin{cases} 1 & \text{if } x_i^t(S) = 1, \\ \mathbb{I}\left[\sum_{j \in \partial i} \left(1 - x_j^t(S)\right) \leq 1\right] & \text{if } x_i^t(S) = 0, \end{cases} \quad [3]$$

where $\partial i = \{j : (ij) \in E\}$ denotes the local neighborhood of vertex I, and $\mathbb{I}$ denotes the indicator function (that is, one if its argument is true and zero otherwise). One can check that the x_i s are monotonous in time (they can only switch from zero to one); hence, they admit a limit $x_i^*(S)$ when $t \to \infty$. At this fixed point, $x_i^*(S) = 0$ if and only if i is in the 2-core of $G \backslash S$; hence, the sufficient and necessary condition for S to be a decycling set of G is $x_i^*(S) = 1$ for all vertices i.

Note that the leaf removal procedure can be equivalently viewed as a particular case of the linear threshold model of epidemic propagation or information spreading. By calling a removed vertex infected (or informed), one sees that the infection (or information) of node i occurs whenever the number of its infected (or informed) neighbors reaches its degree minus one. This equivalence, which was already exploited in refs. 15 and 23, allows us to build on previous works on minimal contagious sets (20, 21, 23) and influence maximization (7–9).

Optimizing the Size of Decycling Sets. From the point of view of statistical mechanics, it is natural to introduce the following probability distribution over the subsets S to find the optimal decycling sets of a given graph

$$\hat{\eta}(S) = \frac{1}{Z(\mu)} e^{\mu |S|} \prod_{i \in V} \mathbb{I}\left[x_i^*(S) = 1\right], \quad [4]$$

where $|S|$ denotes the number of vertices in S, μ is a real parameter to be interpreted as a chemical potential (or an inverse temperature), and the partition function $Z(\mu)$ normalizes this probability distribution. From the preceding discussion, this measure gives a positive probability only to decycling sets, and their minimal size can be obtained as the ground-state energy in the zero-temperature limit:

$$\theta_{\mathrm{dec}}(G) = \frac{1}{N} \lim_{\mu \to -\infty} \frac{1}{\mu} \ln Z(\mu). \quad [5]$$

The computation of this partition function remains at this point a difficult problem; in particular, the variables x_i^* depend on the choice of S in a nonlocal way. One can get around this difficulty in the following way: because the evolution of x_i^t is monotonous in time, it can be completely described by a single integer, $t_i(S) = \min\{t : x_i^t(S) = 1\}$, the time at which i is removed in the parallel evolution described above. Note that $t_i(S) = 0$ if and only if $i \in S$ and $t_i(S) > 0$ otherwise. We use the natural convention $\min \emptyset = \infty$; hence, the nodes i in the 2-core of $G \backslash S$ are precisely those with an infinite removal time $t_i(S) = \infty$. The crucial advantage of this equivalent representation in terms of the activation times is its locality along the graph. Indeed, the dynamical evolution rule (Eq. **3**) can be rephrased as static equations linking the times t_i on neighboring vertices:

$$t_i(S) = \begin{cases} 0 & \text{if } i \in S, \\ \phi_i\left(\{t_j\}_{j \in \partial i}\right) & \text{if } i \in V \backslash S, \end{cases} \quad [6]$$

$$\text{with } \phi_i\left(\{t_j\}_{j \in \partial i}\right) = 1 + \max\nolimits_2\left(\{t_j(S)\}_{j \in \partial i}\right), \quad [7]$$

where we denote $\max_2$ the second largest of the arguments [reordering them as $t_1 \geq t_2 \geq \ldots \geq t_n$, one defines $\max_2(t_1, \ldots, t_n) = t_2$]. In the leaf removal procedure, one vertex is removed in the first step after the time at which all but one of its neighbors has been

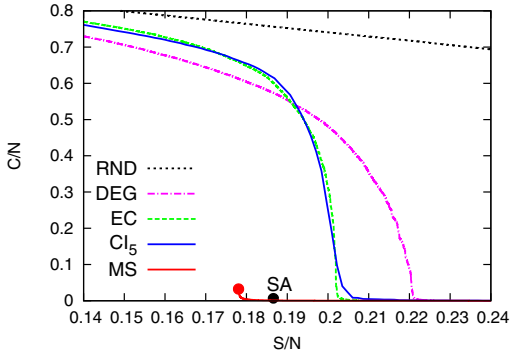

Fig. 1. Fraction of nodes in the largest component as a function of the fraction of removed nodes for an ER random graph of average degree $d = 3.5$ and size $N = 78{,}125$. We compare the result of our Min-Sum algorithm (MS) with random (RND), adaptive largest degree (DEG), adaptive EC, adaptive CI centrality, and SA.

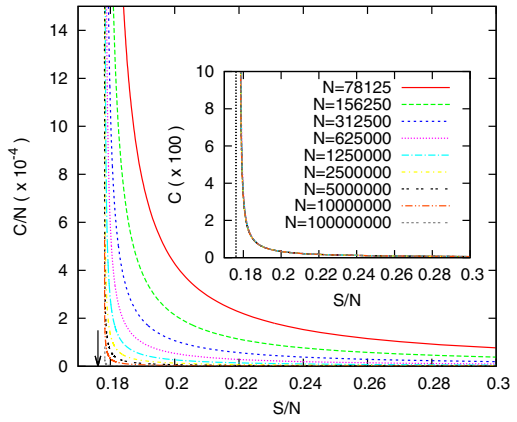

Fig. 2. Fraction of nodes in the largest component C/N as a function of the fraction of nodes removed by the Min-Sum algorithm followed by the greedy tree breaking for the ER random graph of average degree $d = 3.5$ and a range of sizes. (*Inset*) Same plot for the size of the largest component. The collapse of the curves suggests that, to reduce the largest component to a given constant size C, it is sufficient to remove $S = sN$ nodes, where $s = s(C)$ does not depend on N.

removed, making it a leaf. Eq. **6** admits a unique solution for each S; hence, the partition function can be rewritten as

$$Z(\mu) = \sum_{\{t_i\}} e^{\mu \sum_i \psi_i(t_i)} \prod_{i \in V} \mathbb{I}[t_i < \infty] \prod_{i \in V} \Phi\left(t_i, \{t_j\}_{j \in \partial i}\right), \qquad [8]$$

with $\psi_i(t_i) = \mathbb{I}[t_i = 0]$, and $\Phi(t_i, \{t_j\}_{j \in \partial i}) = \mathbb{I}[t_i = 0] + \mathbb{I}[t_i = \phi_i(\{t_j\}_{j \in \partial i})]$. We have thus obtained an exact representation of the generating function counting the number of decycling sets according to their size as a statistical mechanics model of variables (the t_i s) interacting locally along the graph G. We transformed the nonequilibrium problem of leaf removal into an equilibrium problem, where the times of removal play the role of the static variables. Note that ref. 22, which also estimates the decycling number, uses a simpler but approximate representation, where one cycle may remain in every connected component, and the correspondence between microscopic configurations and sets of removed vertices is many to one. The domain of the variables t_i should include all integers between zero and the diameter of G and the additional ∞ value. For practical reasons, in the rest of this paper, we restrict this set to $\{0,1, \ldots, T, \infty\}$, where T is a fixed parameter, and project all t_i s greater than T to ∞. This restriction means not only that we require $G \backslash S$ to be acyclic but that its connected components are trees of diameter at most T. For large-enough values of T, this additional restriction is inconsequential.

The exact computation of the partition function (Eq. **8**) for an arbitrary graph remains an NP-hard problem. However, if G is a sparse random graph, the large size limit of its free energy density $\ln Z(\mu)/(N\mu)$ can be computed by the cavity method (25, 26). The latter has been developed for statistical mechanics models on locally tree-like graphs, such as light-tailed random graphs, for which the exactness of the cavity method has been proven mathematically on several problems. The starting point of the method is based on the fact that light-tailed random graphs converge locally to trees in their large size limit; hence, models defined on them can be treated with belief propagation (BP; also called Bethe Peierls approximation in statistical mechanics). In BP, a partition function akin to Eq. **8** is computed via the exchange of messages between neighboring nodes. In this case, where an interaction in Eq. **8** includes node i and all of its neighbors $j \in \partial i$, we obtain a tree-like representation if we let pairs of variables t_i, t_j live on the edges and add consistency constraints on the nodes. The BP message $\eta_{ij}(t_i, t_j)$ from i to $j \in \partial i$ is then a function of both the activation times t_i and t_j. This message is interpreted as the marginal probability law of the local variables t_i and t_j in an amputated (cavity) graph, in which the interaction between i and j has been removed. Thanks to the locally tree-like character of the graph, some correlation decay properties are verified and allow a node's incoming messages to be treated as independent. Under this assumption, the iterative BP equations (20, 21, 23) for decycling are written as

$$\eta_{ij}(t_i, t_j) \propto \sum_{\{t_k\}_{k \in \partial i \backslash j}} e^{\mu \psi_i(t_i)} \Phi(t_i, \{t_k\}_{k \in \partial i}) \prod_{k \in \partial i \backslash j} \eta_{ki}(t_k, t_i), \qquad [9]$$

where the $\propto$ symbol includes a multiplicative normalization constant. The free energy can then be computed as a sum of local contributions depending on the messages solution of the BP equations.

Better parametrizations with a number of real values per message that scale linearly with T (rather than quadratically) can be devised (21, 23). A parametrization with $2T$ real values per message was introduced in ref. 23 and used to obtain improved results for the minimum decycling set on regular random graphs by extending the cavity method to the so-called first level of the replica symmetry breaking (1RSB) scheme. The extension of this calculation to random graphs with arbitrary light-tailed degree distributions is reported in *SI Appendix* (along with expansions close to the percolation threshold and at large degrees and a lower bound on θ_{dec} valid for all graphs). The 1RSB predictions for the decycling fraction $\theta_{\mathrm{dec}}(d)$ of ER random graphs with average degree d, obtained by solving numerically the corresponding equations and extrapolating the results in the large T limit, are presented for a few values of d in Table 1.

Min-Sum Algorithm for the Decycling Problem. We turn now to the description of our heuristic algorithm for finding decycling sets of the smallest possible size. The above analysis shows the equivalence of this problem with the minimization of the cost function $\sum_i \psi_i(t_i)$ over the feasible configurations of the activation times $\{t_i\} \in \{0, \ldots, T\}^V$, where feasible means that, for all vertices i, either $t_i = 0$ (then i is included in the decycling set S) or if $t_i > 0$, it obeys the constraint $t_i = 1 + \max_2(\{t_j\}_{j \in \partial i})$. Because this minimization is NP hard, we formulate a heuristic strategy in the following manner. We first consider a slightly modified cost function with $\psi_i(t_i) = \mathbb{I}[t_i = 0] + \varepsilon_i(t_i)$, where $\varepsilon_i(t_i)$ is a randomly chosen infinitesimally small cost associated with the removal of node i at time t_i. The minimum $\{t_i^*\}$ of this cost function is now unique with probability one and can be constructed as $t_i^* = \operatorname{argmin} h_i(t_i)$, where the field $h_i(t_i)$ is the minimum cost among the feasible configurations with a prescribed value for the removal

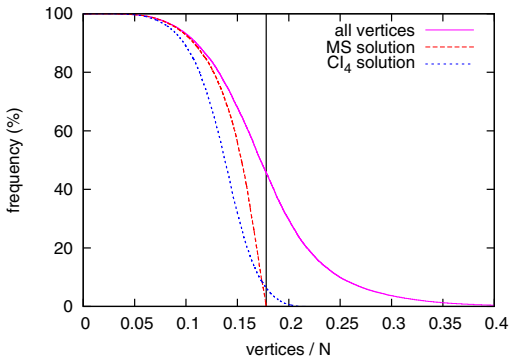

Fig. 3. Frequencies with which vertices appear in different decycling sets on an ER graph with $N = 78{,}125$ and $d = 3.5$. The y axis gives the frequency with which a given vertex appears in close to optimal decycling sets found by the Min-Sum (MS) algorithm. We ordered the vertices by this frequency and depict their ordering divided by N on the x axis. The different curves correspond to all vertices, vertices appearing in one randomly chosen decycling set found by the MS algorithm, and one found by the CI algorithm.

time t_i of site i. From the solution of this combinatorial optimization problem, we construct one of the minimal decycling sets S by including vertex i in S if and only if $t_i^* = 0$. It remains now to find a good approximation for h_i; we compute it by the Min-Sum algorithm, which corresponds to the $\mu \to -\infty$ limit of BP and is similarly based on the exchange of messages $h_{ij}(t_i, t_j)$ between neighboring vertices, an analog of $\eta_{ij}(t_i, t_j)$, but interpreted as a minimal cost instead of a probability. We defer to *SI Appendix* for a full derivation and implementation details, stating here the final equations.

$$h_i(t_i) = \psi_i(t_i) + \sum_{k \in \partial i} L_{ki}(t_i) + M_i(t_i), \qquad [10a]$$

$$h_i(0) = \psi_i(0) + \sum_{k \in \partial i} R_{ki}(0), \qquad [10b]$$

$$M_i(t_i) = \min\left\{0, \min_{k \in \partial i}\{R_{ki}(t_i) - L_{ki}(t_i)\}\right\}, \qquad [11]$$

for $T \geq t_i > 0$, where L_{ij}, R_{ij}, M_{ij}, h_{ij}^0, and h_{ij}^1 form a solution of the following system of fixed point equations for messages defined on each directed edge $i \to j$ of the graph:

$$L_{ki}(t_i) = \min_{t_k < t_i} h_{ki}^0(t_k), \qquad [12a]$$

$$R_{ki}(t_i) = \min\left\{h_{ki}^0(t_i), \min_{t_k > t_i} h_{ki}^1(t_k)\right\}, \qquad [12b]$$

$$M_{ij}(t_i) = \min\left\{0, \min_{k \in \partial i \setminus j}\{R_{ki}(t_i) - L_{ki}(t_i)\}\right\}, \qquad [12c]$$

$$h_{ij}^0(t_i) \propto \psi_i(t_i) + \sum_{k \in \partial i \setminus j} L_{ki}(t_i), \qquad [12d]$$

$$h_{ij}^1(t_i) \propto \psi_i(t_i) + \sum_{k \in \partial i \setminus j} L_{ki}(t_i) + M_{ij}(t_i), \qquad [12e]$$

$$h_{ij}^0(0) \propto \psi_i(0) + \sum_{k \in \partial i \setminus j} R_{ki}(0), \qquad [12f]$$

where $\propto$ includes now an additive normalization constant. An intuitive interpretation of all of these quantities and equations is provided in *SI Appendix*; let us only mention at this point that the message $h_{ij}^0(t_i)$ [respectively $h_{ij}^1(t_i)$] is the minimum feasible cost on the connected component of i in $G\backslash j$ under the condition that i is removed at time t_i in the original graph, assuming that j is not removed yet (respectively assuming that j is already removed from G).

This system can be solved efficiently by iteration. The computation of one iteration takes $O(|E|T)$ elementary $(+, -, \times, \min)$ operations, where $|E|$ denotes the number of edges of the graph, and a relatively small number of iterations is usually sufficient to reach convergence. In principle, one should take the cutoff T on the removal times to be greater than N to solve the decycling problem. We found, however, that using large but finite values of T (i.e., constraining the diameter of the tree components after the node removal) did not increase extensively the size of the decycling set; in the simulations presented below, we used $T = 35$. Note that our algorithm is very flexible, and many variations can be implemented by appropriate modifications of the cost function. For example, we exploited the possibility to forbid the removal of certain marked nodes i by setting $\psi_i(t_i = 0) = \infty$ for them.

Results for Dismantling

Results on Random Graphs. The outcome of our algorithm applied to an ER random graph of average degree 3.5 is presented in Fig. 1. Here, the red circle corresponds to the output of its first stage (decycling with Min-Sum), which yields, after the removal of a fraction 0.1781 of the nodes, an acyclic graph in which the largest components contain a fraction 0.032 of the vertices. The red line corresponds to the second stage, which further reduces the size of the largest component by greedily breaking the remaining trees. We compare with simulated annealing (SA; black circle) as well as several incremental algorithms that successively remove the nodes with the highest scores, where the score of a vertex is a measure of its centrality. Other than a trivial function that gives the same score to all vertices [hence removing the vertices in random order (RND)] and the score of a vertex equal to its degree, we used the eigenvector centrality (EC) measure and the recently proposed CI measure (15). We used all of these heuristics in an adaptive way, recomputing the scores after each removal. Additional details on all of these algorithms can be found in *SI Appendix*.

We see from Fig. 1 that the Min-Sum algorithm outperforms the others by a considerable margin: it dismantles the graph using 13% fewer nodes than the CI method. The Monte Carlo-based SA algorithm performs rather well but is considerably slower than all of the others.

In Fig. 2, we zoom in on the results of the second stage of our algorithm and perform a finite size scaling analysis, increasing the size of the dismantled graphs up to $N = 10^8$. In this way, we identify a threshold for decycling (and thus, for dismantling) by the Min-Sum algorithm that converges toward the value $\theta_{\rm dec}^{MS} \approx 0.1782$, which is close but not equal to the theoretical prediction of the 1RSB calculation $\theta_{\rm dec}^{\rm 1RSB} \approx 0.1753$ (vertical arrow in Fig. 2). Fig. 2, *Inset* shows a remarkable scaling that indicates that the size of the largest component after dismantling by removing a given fraction of nodes does not depend on the graph size.

Combinatorial optimization problems typically exhibit a very large degeneracy of their (quasi)optimal solutions. We performed a detailed statistical analysis of the quasioptimal dismantling sets constructed by our algorithm, exploiting the fact that the Min-Sum algorithm finds different decycling sets for different realizations of the random tie-breaking noise $\varepsilon_i(t_i)$.

For a given ER random graph of average degree 3.5 and size $N = 78{,}125$, we ran the algorithm for 1,000 different realizations of the tie-breaking noise $\varepsilon_i(t_i)$ and obtained 1,000 different decycling sets, all of which had sizes within 40 nodes of one another. Randomly chosen pairs among these 1,000 decycling sets coincided, on average, on 82% of their nodes. For each node, we computed its frequency of appearance among the 1,000 decycling sets that we obtained. We then ordered nodes by this frequency and plotted the frequency as a function of this

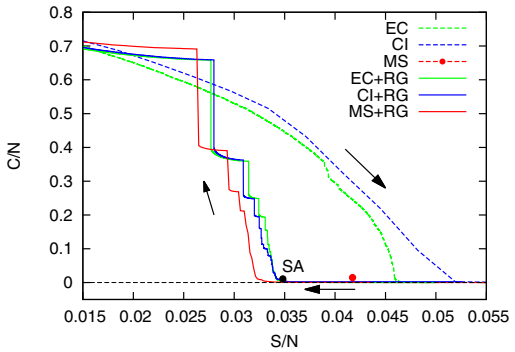

Fig. 4. Fraction C/N taken by the largest component in the Twitter network achieved after removing a fraction S/N of nodes using the Min-Sum (MS) algorithm and the adaptive versions of CI and EC measures. The red circle marks the result obtained by decycling using MS (followed by the curve from the optimal tree-breaking process). The branches at lower values of S/N are obtained after the application of the RG strategy from the graph obtained when the largest component has $C = 100$ nodes. The black circle denotes the dismantling fraction obtained by SA.

ordering in Fig. 3. We see that some nodes appear in almost all found sets, that about 60% of nodes do not appear in any sets, and that a large portion of nodes appears only in a fraction of the decycling sets. We compare the frequencies of nodes belonging to one typical set found by Min-Sum and the CI heuristics.

An important question to ask about dismantling sets is whether they can be thought of as a collection of nodes that are in some sense good spreaders or whether they are a result of highly correlated optimization. We use the result of the previous experiment and remove the nodes that appeared most often (i.e., have the highest frequencies in Fig. 3). If the nature of dismantling was additive rather than collective, then such a choice should further decrease the size of the discovered dismantling set. This scenario is not what happens; with this strategy, we need to remove 20.1% of nodes to dismantle the graph compared with the 17.8% of nodes found systematically by the Min-Sum algorithm. From this observation, we conclude that dismantling is an intrinsically collective phenomenon, and one should always speak of the full set rather than a collection of influential spreaders.

We also studied the degree histogram of nodes that the Min-Sum algorithm includes in the dismantling sets and saw that, as expected, most of the high-degree nodes belong to most of the dismantling sets. Each of the dismantling sets also included some nodes of relatively low degrees; for instance, for an ER random graph of average degree $d = 6$ and size 5^7, a typical decycling set found by the Min-Sum algorithm has around 460 (i.e., around 17% of the decycling set) nodes of degree 4 or lower. To assess the importance of low-degree nodes for dismantling, we ran the Min-Sum algorithm under the constraint that only nodes of degree at least 5 can be removed, and we find decycling sets almost as small (only about 50 nodes; i.e., 0.2% larger) as without this constraint. From this observation, we conclude that none of the low-degree nodes (even those with high CI centrality) are indispensable for dismantling, going against a highlight claim of ref. 15.

More General Graphs. Up to this point, our study of dismantling relies crucially on the relation to decycling. For light-tailed random graphs, these two problems are essentially asymptotically equivalent. However, for arbitrary graphs that contain many small cycles, the decycling number can be much larger than the dismantling one. We argue that, from the algorithmic point of view, decycling still provides a very good basis for dismantling. For instance, consider a portion of $N = 532{,}000$ nodes of the Twitter network already analyzed in ref. 15. The decycling solution found by Min-Sum improves considerably the results obtained with the CI and EC heuristics (Fig. 4).

In a network that contains many short cycles, decycling removes a large proportion of nodes expressly to destroy these short cycles. Many of these nodes can be put back without increasing the size of the largest component. For this reason, we introduce a reverse greedy (RG) procedure, in which starting from a dismantled graph with dismantling set S, maximum component size C, and a chosen target value $C' > C$ for the maximum allowed component size, removed nodes are iteratively reinserted. At each step, among all removed nodes, the one that ends up in the smallest connected component is chosen for reinsertion (details are in *SI Appendix*). The computational cost of this operation is bounded by $k_{\max} C' \log(k_{\max} C')$, where $k_{\max}$ is the maximal degree of the graph; the update cost is thus typically sublinear in N.

In graphs where decycling is an optimal strategy for dismantling, such as the random graphs, a vanishing fraction of nodes can be reinserted by the RG procedure before the size of the largest component starts to grow steeply. For real world networks, the RG procedure reinserts a considerable number of nodes, negligibly altering the size of the largest component. For the Twitter network in Fig. 4, the improvement obtained by applying the RG procedure is impressive: 32% fewer nodes for the CI method and 20% fewer nodes for the Min-Sum algorithm, which ends up being the best solution that we found, removing only 3.4% of nodes to dismantle into components smaller than $C = 1{,}000$ nodes. RG makes it possible to reach, and even improve, the best result obtained with SA that solves the dismantling problem directly and is not affected by the presence of short loops (*SI Appendix* has details on SA). Qualitatively similar results are achieved on other real networks [e.g., on the YouTube network with 1.13 million nodes (27), the best dismantling set that we found with Min-Sum + RG included 4.0% of nodes; this result is a 22% improvement with respect to the CI heuristics].

The RG procedure is introduced as a heuristic that provides a considerable improvement for the examples that we treated. The theoretical results of this paper are valid only for classes of graphs that do not contain many small cycles, and hence, our theory does not provide a principled derivation or analysis of the RG procedure. This point is an interesting open direction for future work. More detailed study (both theoretical and algorithmic) of dismantling of networks for which decycling is not a reasonable starting point is an important direction of future work.

ACKNOWLEDGMENTS. A.B. and L.D. acknowledge support by Fondazione Cassa di Risparmio di Torino (CRT) under the initiative La Ricerca dei Talenti. L.D. acknowledges the European Research Council Grant 267915.

1. Boccaletti S, Latora V, Moreno Y, Chavez M, Hwang DU (2006) Complex networks: Structure and dynamics. *Phys Rep* 424(4):175–308.
2. Barrat A, Barthelemy M, Vespignani A (2008) *Dynamical Processes on Complex Networks* (Cambridge Univ Press, Cambridge, UK).
3. Pastor-Satorras R, Vespignani A (2002) Immunization of complex networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 65(3 Pt 2A):036104.
4. Cohen R, Havlin S, Ben-Avraham D (2003) Efficient immunization strategies for computer networks and populations. *Phys Rev Lett* 91(24):247901.
5. Britton T, Janson S, Martin-Löf A (2007) Graphs with specified degree distributions, simple epidemics, and local vaccination strategies. *Adv Appl Probab* 39(4):922–948.
6. Altarelli F, Braunstein A, Dall'Asta L, Wakeling JR, Zecchina R (2014) Containing epidemic outbreaks by message-passing techniques. *Phys Rev X* 4(2):021024.
7. Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of influence through a social network. *KDD* 03:137–146.
8. Chen N (2008) On the approximability of influence in social networks. *SODA* 08: 1029–1037.
9. Dreyer PA, Jr, Roberts FS (2009) Irreversible k-threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion. *Discrete Appl Math* 157(7): 1615–1627.
10. Janson S, Thomason A (2008) Dismantling sparse random graphs. *Comb Probab Comput* 17(02):259–264.
11. Bradonjic M, Molloy M, Yan G (2013) Containing viral spread on sparse random graphs: Bounds, algorithms, and experiments. *Internet Math* 9(4):406–433.
12. Edwards K, McDiarmid C (1994) New upper bounds on harmonious colorings. *J Graph Theory* 18(3):257–267.
13. Edwards K, Farr G (2001) Fragmentability of graphs. *J Comb Theory Ser B* 82(1):30–37.
14. Edwards K, Farr G (2008) Planarization and fragmentability of some classes of graphs. *Discrete Math* 308(12):2396–2406.
15. Morone F, Makse HA (2015) Influence maximization in complex networks through optimal percolation. *Nature* 524(7563):65–68.
16. Albert R, Jeong H, Barabási AL (2000) Error and attack tolerance of complex networks. *Nature* 406(6794):378–382.
17. Callaway DS, Newman MEJ, Strogatz SH, Watts DJ (2000) Network robustness and fragility: Percolation on random graphs. *Phys Rev Lett* 85(25):5468–5471.
18. Cohen R, Erez K, ben-Avraham D, Havlin S (2001) Breakdown of the internet under intentional attack. *Phys Rev Lett* 86(16):3682–3685.
19. Karp RM (1972) Reducibility among combinatorial problems. *Complexity of Computer Computations*, eds Miller RE, Thatcher JW, Bohlinger JD (Springer, Boston), pp 85–103.
20. Altarelli F, Braunstein A, Dall'Asta L, Zecchina R (2013) Large deviations of cascade processes on graphs. *Phys Rev E Stat Nonlin Soft Matter Phys* 87(6):062115.
21. Altarelli F, Braunstein A, Dall'Asta L, Zecchina R (2013) Optimizing spread dynamics on graphs by message passing. *J Stat Mech* 09:P09011.
22. Zhou HJ (2013) Spin glass approach to the feedback vertex set problem. *Eur Phys J B* 86(11):1–9.
23. Guggiola A, Semerjian G (2015) Minimal contagious sets in random regular graphs. *J Stat Phys* 158(2):300–358.
24. Mugisha S, Zhou HJ (2016) Identifying optimal targets of network attack by belief propagation. *Phys Rev E Stat Nonlin Soft Matter Phys* 94(1-1):012305.
25. Mézard M, Parisi G (2001) The Bethe lattice spin glass revisited. *Eur Phys J B* 20:217–233.
26. Mézard M, Montanari A (2009) *Information, Physics and Computation* (Oxford Univ Press, London).
27. Leskovec J, Krevl A (2014) *SNAP Datasets: Stanford Large Network Dataset Collection*. Available at snap.stanford.edu/data. Accessed September 30, 2015.

Supplementary Information: Network dismantling

Alfredo Braunstein, Luca Dall'Asta, Guilhem Semerjian, and Lenka Zdeborová

PACS numbers:

Contents

I. Proof of NP-Completeness of the dismantling problem 1

II. Analytic results 2
A. Details on the cavity equations for the decycling number of random graphs 2
B. A simple lower bound 4
C. Decycling close to the percolation threshold and for large degrees 6

III. Algorithms 6
A. The Min-Sum algorithm and its implementation 6
B. Tree-breaking in decycled graphs 9
1. Optimal tree breaking 9
2. Greedy tree-breaking 10
C. Greedy reintroduction of cycles 10
D. Competing algorithms 11
1. Simulated Annealing 11
2. Score-based algorithms 13

IV. Other real-world and scale-free graphs 13

References 15

We provide in this Supplementary Information further technical details and additional results in support of the main text. It is organized as follows. In Section I we prove the NP-completeness of the dismantling decision problem. In Sec. II we extend the analytic results of the main text, presenting the details of the cavity method computation of the decycling number of random graphs (II A), a lower bound on the decycling number valid for all graphs (II B), and an expansion of the decycling number for Erdős-Rényi random graphs close to their percolation threshold and for large average degrees (II C). Section III is then devoted to several algorithmic aspects: in III A, III B and III C we detail the three stages of our main algorithm (derivation of the Min-Sum equations, tree dismantling and greedy reintroduction of cycles respectively), while in III D we give further details on the other dismantling algorithms we have studied. Finally in Section IV we provide further results on other real-world and artificial scale-free networks.

I. PROOF OF NP-COMPLETENESS OF THE DISMANTLING PROBLEM

For our proof, we will employ the decisional (minimum) Vertex Cover problem, which is NP-Complete, and is defined as follows. Remember that a vertex cover is a subset of vertices $W \subset V$ such that for each (i, j) in E, $i \in W$ or $j \in W$.

Vertex Cover: Given a graph $G = (V, E)$ and $F \in \mathbb{N}$, does a vertex cover $W \subset V$ with $|W| \leq F$ exist?

The Vertex Cover problem is NP-Complete.

$C(N)$–Dismantling: Given a graph $G' = (V', E')$ and $F \in \mathbb{N}$, does a $C(|V'|)$–dismantling set S with $|S| \leq F$ of G' exist?

Theorem 1. *Assume $C : \mathbb{N} \to \mathbb{N}$ to be a non-decreasing (polynomially computable) function with $C(N) < N^a$ for $N \geq N_0$ with $0 \leq a < 1$. Then the $C(N)$–Dismantling problem is NP-Complete.*

Proof. $C(N)$–DISMANTLING belongs clearly to NP. If $C \equiv 1$, one can see that C–DISMANTLING is identical to VERTEX COVER and is thus is NP–Complete. Otherwise, take N_1 such that $C(N_1) \geq 2$ and consider $N \geq \max\{N_0, N_1\}$. Define

$$C' = C'(N) = \min\left\{k \in \mathbb{N} : \frac{C(kN)}{k} < 2\right\}. \tag{1}$$

Remark. For constant $C(N) \equiv C \geq 2$, then $C'(N) \equiv \frac{C+1}{2}$ if C is odd, and $C'(N) \equiv \frac{C}{2} + 1$ if C is even.

Note that $C' \geq 2$ and C' is polynomial in N: the value $k = \left\lceil \frac{1}{2} N^a \right\rceil^{\frac{1}{1-a}}$ belongs to the set in the RHS of (1), as $k \geq \left(\frac{1}{2}N\right)^{\frac{a}{1-a}}$ so $k^{a-1} \leq \left(\frac{1}{2}N\right)^{-a}$ and then $\frac{C(kN)}{k} < \frac{(kN)^a}{k} = k^{a-1}N^a \leq 2$; so $C' \leq k = \left\lceil \frac{1}{2} N^a \right\rceil^{\frac{1}{1-a}}$. We will prove that

$$C' \leq C(C'N) < 2C'. \tag{2}$$

The second inequality in (2) follows from (1). For the first inequality,

$$\begin{aligned} C(C'N) &\geq C((C'-1)N) && (3)\\ &\geq 2(C'-1) && (4)\\ &\geq C', && (5) \end{aligned}$$

where (3) follows from the fact that C is non-decreasing, (4) follows from the minimality of C' in its definition (1) and (5) from the fact that $C' \geq 2$.

Now, take a graph G with $|V| = N$, and construct G' by adding $C' - 1$ leaves to any vertex of G. Precisely, let $G' = (V', E')$ with $V' = V \cup \{\ell_{v,i} : v \in V, i \in \{1, \dots, C'-1\}\}$ and $E' = E \cup \{(v, \ell_{v,i}) : v \in V, i \in \{1, \dots, C'-1\}\}$ where we assume the two unions to be disjoint. The number of vertices of G' is $|V'| = N' = NC'$ (which is polynomial in N). The construction of G' is clearly polynomial.

Take any vertex cover W of G. Then W is a C'–dismantling of G': thanks to the vertex cover property, each $v \in V \setminus W$ can only be connected to the $C'-1$ extra leaves $\ell_{v,i}$. As $C' \leq C(N')$, then W is also a $C(N')$–dismantling of G'.

Conversely, take any $C(N')$–dismantling set S of G'. Define $\pi : V' \to V$ by $\pi(v) = v$ for $v \in V$ and $\pi(\ell_{v,i}) = v$ for $v \in V, 1 \leq i \leq C'-1$. Consider the set $W = \pi(S)$. In short, W is constructed from S by replacing all occurrences $\ell_{v,i}$ by v. Then clearly $|W| = |\pi(S)| \leq |S|$ and W is still a $C(N')$–dismantling of G': replacing $\ell_{v,i}$ by v introduces a new component $\{\ell_{v,i}\}$ of size 1 but can only reduce the size of the other components. Moreover, W is also a vertex cover of G: suppose on the contrary that it is not, and take an edge $(i,j) \in E$ such that $i, j \notin W$. Then both vertices belong to a connected component of $G' \setminus W$ of size $2C' > C(N')$, which contradicts the fact that W was a $C(N')$–dismantling of G'. Thus, W must be a vertex cover of G with size no greater than $|S|$ and that proves the result. □

Corollary 2. *For $C(N) = const$, $C(N) = \log N$, and $C(N) = N^a$ with $0 \leq a < 1$, $C(N)$–DISMANTLING is NP-Complete.*

Remark. $(N-k)$–DISMANTLING is polynomial for any constant k.

II. ANALYTIC RESULTS

A. Details on the cavity equations for the decycling number of random graphs

We give here some more details on the cavity method computation of the decycling number of sparse random graphs, in particular on the derivation and solution of the BP equations. A full derivation in a more general context can be found in [1].

When computing the typical free-energy of a large random graph with degree distribution q one has to determine the probability law $P(\eta)$ of the messages η, which is the solution of an integral equation of the form:

$$P(\eta) = \sum_{k=0}^{\infty} \widetilde{q}_k \int \mathrm{d}P(\eta^{(1)}) \dots \mathrm{d}P(\eta^{(k)})\ \delta(\eta - f_k(\eta^{(1)}, \dots, \eta^{(k)}))\ , \tag{6}$$

where $\widetilde{q}_k = (k+1)q_{k+1} / \sum_k k q_k$ is the size-biased distribution associated to q (i.e. the probability of finding a vertex of degree $k+1$ when choosing an edge uniformly at random), and f_k the function encoding the local BP equation, eq. (9)

in the main text, between messages around a vertex of degree $k+1$. This type of equation can be efficiently solved numerically via a population dynamics procedure, in which P is approximated by a large sample of representative values of η, updated according to (6) until convergence to a fixed point. The free-energy density of the model can then be computed as the average with respect to P of suitable functions of the messages. In the present model these messages are real vectors of a dimension which grows linearly with the parameter T introduced above as a cutoff on the allowed times in the leaf-removal dynamics.

In the Replica Symmetric version of the cavity method a message (or field) η of (6) corresponds to a $2T$ dimensional vector of components denoted $(a_0, a_1, \ldots, a_T, b_{T-1}, \ldots, b_1)$. The function f_k which gives η as a function of $\eta^{(1)}, \ldots, \eta^{(k)}$ reads explicitly:

$$\begin{aligned} e^{-\mu b_t} &= 1 + e^{-\mu + \mu \sum_{i=1}^{k} (a_0^{(i)} - b_{t-1}^{(i)})} , \\ e^{-\mu a_t} - e^{-\mu a_{t+1}} &= e^{-\mu + \mu \sum_{i=1}^{k} (a_0^{(i)} - b_t^{(i)})} \sum_{i=1}^{k} \left(e^{\mu(b_t^{(i)} - a_{t+1}^{(i)})} - e^{\mu(b_t^{(i)} - a_{t+2}^{(i)})} \right) , \end{aligned} \tag{7}$$

with the conventions used to have more compact expressions: $b_0 = 0$, $b_T = a_T$, $a_{T+1} = b_{T-1}$. Once the self-consistent equation on $P(\eta)$ is solved the thermodynamic quantities are obtained as follows. The limit of $(\ln Z)/N$ reads

$$\phi = \mu + \mathbb{E}[\ln z_{\text{site}}(\eta^{(1)}, \ldots, \eta^{(k)})] - \frac{d}{2} \mathbb{E}[\ln z_{\text{edge}}(\eta^{(1)}, \eta^{(2)})] ,$$

where $\mathbb{E}[\cdot]$ denotes the average over the i.i.d. copies $\eta^{(i)}$ drawn from $P(\eta)$ and over the integer k drawn from the degree distribution q, and d is the mean of q. The two functions z_{site} and z_{edge} arise from the local contributions to the Bethe free-energy of sites and edges respectively, and read

$$z_{\text{site}} = 1 + e^{-\mu + \mu \sum_{i=1}^{k} a_0^{(i)}} \left[e^{-\mu \sum_{i=1}^{k} b_{T-1}^{(i)}} + \sum_{t=1}^{T} e^{-\mu \sum_{i=1}^{k} b_{t-1}^{(i)}} \sum_{i=1}^{k} \left(e^{\mu(b_{t-1}^{(i)} - a_t^{(i)})} - e^{\mu(b_{t-1}^{(i)} - a_{t+1}^{(i)})} \right) \right] , \tag{8}$$

$$z_{\text{edge}} = e^{\mu(a_0^{(1)} + a_0^{(2)})} \left[e^{-\mu(b_T^{(1)} + b_T^{(2)})} + \sum_{t=0}^{T-1} \left\{ (e^{-\mu a_t^{(1)}} - e^{-\mu a_{t+1}^{(1)}}) e^{-\mu b_t^{(2)}} + (e^{-\mu a_t^{(2)}} - e^{-\mu a_{t+1}^{(2)}}) e^{-\mu b_t^{(1)}} \right\} \right] . \tag{9}$$

The fraction θ of vertices included in the decycling sets selected by the conjugated chemical potential μ and the entropy s (the Legendre transform of ϕ) then read:

$$\theta = \mathbb{E}\left[\frac{1}{z_{\text{site}}(\eta^{(1)}, \ldots, \eta^{(k)})} \right] , \quad s = \phi - \mu\, \theta . \tag{10}$$

Varying the parameter μ one can compute in this way the entropy $s(\theta)$ counting the exponential number of decycling sets containing a fraction θ of vertices. The RS estimate of the decycling number θ_{dec} is then obtained as the point where s vanishes.

This estimate is, however, only a lower bound to the true value of θ_{dec} because of the effects of the replica symmetry breaking. A more precise estimate is obtained by using the (energetic) cavity method at the first level of replica symmetry breaking (1RSB), in which the parameter μ is replaced by the Parisi breaking parameter y; the message η of (6) is then a vector $(p_0, \ldots, p_{T-1}, r_T, \ldots, r_0)$ constrained by the normalization $p_0 + \cdots + p_{T-1} + r_T + \cdots + r_0 = 1$

(hence the number of independent parameters is again $2T$). These are updated according to

$$p_t = \frac{1}{Z} e^y \widetilde{p}_t \ , \tag{11}$$

$$r_t = \frac{1}{Z} e^y \widetilde{r}_t \quad \text{for} \quad t \geq 1 \ , \tag{12}$$

$$r_0 = \frac{1}{Z} \left(1 - \sum_{t=0}^{T-1} \widetilde{p}_t - \sum_{t=1}^{T} \widetilde{r}_t \right) \ , \tag{13}$$

$$Z = 1 + (e^y - 1) \left(\sum_{t=0}^{T-1} \widetilde{p}_t + \sum_{t=1}^{T} \widetilde{r}_t \right) \ , \tag{14}$$

$$\widetilde{p}_t = \sum_{i=1}^{k} p_{t+1}^{(i)} \prod_{j \neq i} \left(\sum_{t'=0}^{t} r_{t'}^{(j)} \right) \ , \tag{15}$$

$$\widetilde{r}_t = \prod_{i=1}^{k} \left(\sum_{t'=0}^{t-1} r_{t'}^{(i)} \right) - \prod_{i=1}^{k} \left(\sum_{t'=0}^{t-2} r_{t'}^{(i)} \right) \ , \tag{16}$$

with $p_T = r_T$ by convention. One computes then a thermodynamic potential $\Phi(y)$ with a formula similar to the one yielding $\phi(\mu)$ at the RS level, namely

$$\Phi = -y + \mathbb{E}[\ln \mathcal{Z}_{\text{site}}(\eta^{(1)}, \ldots, \eta^{(k)})] - \frac{d}{2} \mathbb{E}[\ln \mathcal{Z}_{\text{edge}}(\eta^{(1)}, \eta^{(2)})] \ ,$$

with

$$\mathcal{Z}_{\text{site}} = 1 + (e^y - 1) \left[\prod_{i=1}^{k} \left(\sum_{t=0}^{T-1} r_t^{(i)} \right) + \sum_{i=1}^{k} \sum_{t=1}^{T} p_t^{(i)} \prod_{j \neq i} \left(\sum_{t=0}^{t-1} r_t^{(j)} \right) \right] \ , \tag{17}$$

$$\mathcal{Z}_{\text{edge}} = e^{-y} + (1 - e^{-y}) \left[\left(\sum_{t=0}^{T} r_t^{(1)} \right) \left(\sum_{t=0}^{T} r_t^{(2)} \right) + \sum_{t=0}^{T-1} \left\{ p_t^{(1)} \left(\sum_{t'=0}^{t} r_{t'}^{(2)} \right) + p_t^{(2)} \left(\sum_{t'=0}^{t} r_{t'}^{(1)} \right) \right\} \right] \ . \tag{18}$$

The energetic complexity function (the equivalent of the entropy at the 1RSB level) is then obtained by an inverse Legendre transform with respect to Φ, namely

$$\Sigma = \Phi + y\,\theta \ , \qquad \theta = 1 - \mathbb{E}\left[\frac{\mathcal{Z}'_{\text{site}}}{\mathcal{Z}_{\text{site}}} \right] + \frac{d}{2} \mathbb{E}\left[\frac{\mathcal{Z}'_{\text{edge}}}{\mathcal{Z}_{\text{edge}}} \right] \ ,$$

where the prime denotes the derivative with respect to the explicit dependence in y of the expressions of $\mathcal{Z}_{\text{site}}$ and $\mathcal{Z}_{\text{edge}}$ given above. The 1RSB estimate of the decycling number is then obtained from the criterion of cancellation of the complexity Σ. Both the replica symmetric and 1RSB results for a range of values of T are reported in Table 1 in the main text. Extrapolating the 1RSB estimate of the decycling number in the limit $T \to \infty$ leads to the values reported in Table 1 in the main text.

The replica symmetric and 1RSB computations yield improving lower bounds on the decycling number of light-tailed random graphs, in the sense that $\theta^{\text{RS}}(q) \leq \theta^{\text{1RSB}}(q) \leq \theta_{\text{dec}}(q)$. For some degree distributions q these inequalities become equalities (see [1] for details on random regular graphs), for others the 1RSB estimate is strictly tighter than the RS one. It is probable that for some choices of q the 1RSB estimate is not equal to the decycling number, whose exact determination would require the use of the so-called full RSB computation. The latter is not tractable numerically for models of sparse random graphs, we expect in any case the quantitative difference between the 1RSB and full RSB results to be rather small.

B. A simple lower bound

We present here a lower bound on the decycling number $\theta_{\text{dec}}(G)$ valid for any graph G (a similar reasoning can be found in [2]), generalizing the bound $\theta_{\text{dec}}(G) \geq \frac{d-2}{2(d-1)}$ for d-regular graphs.

TABLE I: The T-dependence of the RS and 1RSB cavity predictions for the decycling number of Erdös-Rényi random graphs of average degree $d = 3.5$. The decycling numbers reported in table 1 in the main text are obtained by interpolation of the 1RSB results to $T \to \infty$.

T	$\theta^{\rm RS}(T)$	$\theta^{\rm 1RSB}(T)$
3	0.22714	0.22797
5	0.20042	0.20077
9	0.18507	0.18515
13	0.18046	0.18051
19	0.17795	0.17797
30	0.17638	0.17638
40	0.17590	0.17590
50	0.17569	0.17569

We denote k_i the degree of vertex i and M the number of edges. With

$$\langle k \rangle = \frac{1}{N} \sum_{i \in V} k_i \tag{19}$$

the empirical average degree, one has $M = N \frac{\langle k \rangle}{2}$.

Consider now a subset S of the vertices, and its complement $S^c = V \setminus S$. One can divide the edges in three categories, with $M = M_1 + M_2 + M_3$, where M_1 is the number of edges between two vertices of S, M_2 counting the edges between S and S^c, and M_3 the edges inside S^c. One has

$$\sum_{i \in S} k_i = 2M_1 + M_2 \ , \qquad \sum_{i \in S^c} k_i = 2M_3 + M_2 \ , \tag{20}$$

and in particular

$$M_1 + M_2 \leq \sum_{i \in S} k_i \ . \tag{21}$$

Suppose now that S is a decycling set of the graph, in such a way that S^c induces a forest. Hence one has

$$M_3 \leq |S^c| - 1 = N - |S| - 1 \ . \tag{22}$$

Summing these two inequalities, and expressing M in terms of the average degree, yields

$$\frac{1}{N} \sum_{i \in S} (k_i - 1) \geq \frac{\langle k \rangle}{2} - 1 + \frac{1}{N} \ . \tag{23}$$

This inequality constrains the possible decycling sets. To obtain a simpler lower bound on the size of the decycling sets, consider a permutation σ from $\{1, \ldots, N\}$ to V that orders the vertices according to their degrees: $k_{\sigma(1)} \geq k_{\sigma(2)} \geq \ldots$. The inequality above can then be continued to get

$$\frac{1}{N} \sum_{i=1}^{|S|} (k_{\sigma(i)} - 1) \geq \frac{\langle k \rangle}{2} - 1 + \frac{1}{N} \ . \tag{24}$$

Let us call the left hand side of this inequality $l(\theta = |S|/N, G)$, which is an increasing function of θ, and define $\theta_{\rm lb}(G)$ as the smallest value of θ such that the inequality is fulfilled. Then the decycling number $\theta_{\rm dec}(G)$ of this graph is certainly lower-bounded by $\theta_{\rm lb}(G)$.

The shape of $l(\theta)$ can be described in terms of the empirical degree distribution of the graph,

$$\widehat{q}_k = \frac{1}{N} \sum_{i \in V} \delta_{k,k_i} \ . \tag{25}$$

As the graph is finite so is its maximal degree, let us call it K. One realizes easily that $l(\theta, G)$ is a piecewise linear continuous increasing function, starting from 0 in $\theta = 0$, linearly increasing on $\theta \in [0, \widehat{q}_K]$ with slope $K - 1$, then again

with a constant slope $K-2$ on the interval $\theta \in [\widehat{q}_K, \widehat{q}_K + \widehat{q}_{K-1}]$, and so on and so forth. It is thus more convenient to introduce two integrated quantities:

$$\widehat{Q}_k = \sum_{k'=k}^{\infty} \widehat{q}_{k'} \ , \qquad \widehat{T}_k = \sum_{k'=k}^{\infty} \widehat{q}_{k'}(k'-1) \ , \tag{26}$$

the summations being cut off at K in this finite graph case. Indeed for all k one has $l(\widehat{Q}_k, G) = \widehat{T}_k$, and the function $l(\theta, G)$ is the linear interpolation between this discrete set of points. One can thus determine its intersection with the right hand side of (24) to compute the lower bound $\theta_{\rm lb}(G)$.

In the case of random graphs drawn with a degree distribution q the typical decycling number $\theta_{\rm dec}(q)$ can be lower-bounded as above by replacing the empirical distribution $\widehat{q}$ by q: $\theta_{\rm lb}(q) = l^{-1}\left(\frac{\langle k\rangle}{2} - 1\right)$, where $\langle k\rangle$ is now averaged with respect to q_k, and l is defined by replacing $\widehat{Q}_k$ and $\widehat{T}_k$ by their counterparts

$$Q_k = \sum_{k'=k}^{\infty} q_{k'} \ , \qquad T_k = \sum_{k'=k}^{\infty} q_{k'}(k'-1) \ . \tag{27}$$

The numerical evaluation of this lower bound for a Poissonian random graph of average degree $d = 3.5$ yields $\theta_{\rm lb} = 0.141084$, not that far from the 1RSB prediction $\theta_{\rm dec} = 0.175$. The lower bound matches the asymptotic expansion presented below when $d \to 1$ (i.e. close to the percolation threshold), while it reaches the limit $1/2$ when d diverges (the large d limit of $\theta_{\rm dec}$ being 1).

C. Decycling close to the percolation threshold and for large degrees

In addition to the numerical results obtained by the cavity method let us state analytical asymptotic expansions for the decycling number of Poissonian random graphs with average degree close to the percolation threshold ($d = 1+\epsilon$) or very large ($d \to \infty$). Close to the percolation a random graph is essentially made of a 3-regular kernel of vertices joined by paths of degree 2 nodes; decycling the kernel is sufficient to decycle the whole graph, and the decycling number of a random 3-regular is known [3], which yields

$$\theta_{\rm dec}(d = 1+\epsilon) = \frac{1}{3}\epsilon^3 + O(\epsilon^4) \ . \tag{28}$$

On the other hand when d is very large the Poissonian random graph behaves like a regular graph (the degree distribution being concentrated around its average), an asymptotic expansion in this case was obtained in [1] (in agreement with the rigorous bounds of [4]), hence

$$\theta_{\rm dec}(d) = 1 - \frac{2\ln d}{d} - \frac{2}{d} + O\left(\frac{1}{d\ln d}\right) \ . \tag{29}$$

III. ALGORITHMS

A. The Min-Sum algorithm and its implementation

In this section we derive the Min-Sum (MS) algorithm introduced in eqs. (10-12) of the main text, that aims at finding decycling sets of the smallest possible size. As explained in the main text this amounts to find the unique minimum of the cost function $\sum_i \psi_i(t_i)$, with $\psi_i(t_i) = \mathbb{I}[t_i = 0] + \varepsilon_i(t_i)$, over the feasible configurations of the activation times $\{t_i\} \in \{0, \dots, T\}^V$. These variables have to fulfill the constraint that for all vertices i either $t_i = 0$ (when i belongs to the decycling set) or it is determined by the adjacent variables according to $t_i = 1 + \max_2(\{t_j\}_{j\in\partial i})$. We recall that the $\varepsilon_i(t_i)$ are infinitesimally small random variables that are introduced to ensure the uniqueness of the minimum of the cost function, and its closeness to one of the minima of the original cost function. In practice we took ϵ_i to be uniformly random between 0 and 10^{-7}.

It turns out to be easier to study a slight modification of this optimization problem, with a relaxed constraint

$$t_i \geq 1 + \max_2(\{t_j\}_{j\in\partial i}) \qquad \text{if } \ 0 < t_i \leq T \tag{30}$$

that corresponds to a lazy version of the leaf removal algorithm, in which a node can be removed once it became a leaf, but it is not necessarily removed as soon as it could be. Thanks to the monotonicity of the leaf removal procedure its

final outcome, the 2-core of the graph, is independent of the order in which the leaves are removed, and of the parallel or sequential character of these updates. Hence the optimization problem with the strict or relaxed constraints are completely equivalent if $T \geq N$, the maximal number of possible steps of the leaf removal. For smaller values of T this equivalence is not ensured, but the optimum with the relaxed constraints still provides a valid decycling set. It will be useful in the following to use the following logical equivalent of (30),

$$\sum_{j \in \partial i} \mathbb{I}\left[t_j \geq t_i\right] \leq 1 \qquad \text{if } 0 < t_i \leq T \ . \tag{31}$$

Our goal now is to compute the field $h_i(t_i)$ defined as the minimum of the cost function over feasible configurations with a given value of t_i, as indeed the unique minimum can be deduced from the fields through $t_i^* = \operatorname{argmin} h_i(t_i)$, and then the corresponding decycling set is identified with the vertices i where $t_i^* = 0$. To justify the MS heuristics for the approximate computation of the fields h_i on any graph it is simpler to consider first a tree graph, on which the MS approach is exact for any local cost function. The function under consideration here is the sum of local terms $\psi_i(t_i)$ on each vertex, $h_i(t_i)$ can thus be decomposed as a sum of its own contribution ψ_i and of the contributions of the vertices in each of the subtrees rooted at one of its neighbor $j \in \partial i$. Taking into account the constraint (31) for the positive times, denoted $\mathcal{C}$ in the following, it yields

$$h_i\left(t_i\right) = \psi_i\left(t_i\right) + \min_{\{t_j\}_{j \in \partial i} : \mathcal{C}} \sum_{j \in \partial i} h_{ji}\left(t_j, t_i\right) \qquad \text{for } 0 < t_i \leq T \ , \tag{32}$$

$$h_i\left(0\right) = \psi_i\left(0\right) + \sum_{j \in \partial i} \min_{t_j} h_{ji}\left(t_j, 0\right) \ , \tag{33}$$

where $h_{ji}(t_j, t_i)$ are messages defined on each directed edge $j \to i$ of the graph, that give the minimum cost of the variables in the subtree rooted at j and excluding i, over the feasible configurations with prescribed values of t_j and t_i. Thanks to the recursive structure of a tree these messages obey themselves similar equations,

$$h_{ij}\left(t_i, t_j\right) = \psi_i\left(t_i\right) + \min_{\{t_k\}_{k \in \partial i \setminus j} : \mathcal{C}} \sum_{k \in \partial i \setminus j} h_{ki}\left(t_k, t_i\right) \qquad \text{for } 0 < t_i \leq T \ , \tag{34}$$

$$h_{ij}\left(0, t_j\right) = \psi_i\left(0\right) + \sum_{k \in \partial i \setminus j} \min_{t_k} h_{ki}\left(t_k, 0\right) \ . \tag{35}$$

These Min-Sum equations involve $O(T^2)$ quantities for each edge of the graph because of the two time indices of the messages h_{ij}. Fortunately this quadratic dependence on T can be reduced to a linear one by some further simplifications that we now explain.

As can be readily seen from (34) and (31), the dependence of h_{ij} on t_j is only through $\mathbb{I}\left[t_j < t_i\right]$. We will thus define

$$h_{ij}\left(t_i, t_j\right) = \begin{cases} h_{ij}^1\left(t_i\right) & \text{if } t_j < t_i \ , \\ h_{ij}^0\left(t_i\right) & \text{if } t_j \geq t_i \ , \end{cases} \tag{36}$$

which gives a parametrization of each message with $O(T)$ real numbers.

Calling $\mathcal{T}_{ij}^s = \{\{t_k\}_{k \in \partial i \setminus j} : \sum_{k \in \partial i \setminus j} \mathbb{I}\left[t_k \geq t_i\right] = s\}$, Eq. (34) can be rewritten as follows for $0 < t_i \leq T$:

$$\begin{aligned} h_{ij}^0\left(t_i\right) &= \psi_i\left(t_i\right) + \min_{\mathcal{T}_{ij}^0} \sum_{k \in \partial i \setminus j} h_{ki}\left(t_k, t_i\right) \\ &= \psi_i\left(t_i\right) + \sum_{k \in \partial i \setminus j} \min_{t_k < t_i} h_{ki}\left(t_k, t_i\right) \\ &= \psi_i\left(t_i\right) + \sum_{k \in \partial i \setminus j} \min_{t_k < t_i} h_{ki}^0\left(t_k\right) \ , \end{aligned} \tag{37}$$

as indeed $t_j \geq t_i$ in the definition of h_{ij}^0 all the other removal times t_k for $k \in \partial i \setminus j$ have to be strictly smaller than t_i for the condition (31) to be fulfilled. On the other hand in the situation described by h_{ij}^1 at most one of the removal

times t_k for $k \in \partial i \setminus j$ can be greater or equal than t_i, hence for $0 < t_i \leq T$:

$$\begin{aligned}
h^1_{ij}(t_i) &= \psi_i(t_i) + \min_{\mathcal{T}^0_{ij} \cup \mathcal{T}^1_{ij}} \sum_{k \in \partial i \setminus j} h_{ki}(t_k, t_i) \\
&= \psi_i(t_i) + \min \left\{ \sum_{k \in \partial i \setminus j} \min_{t_k < t_i} h^0_{ki}(t_k), \min_{\mathcal{T}^1_{ij}} \sum_{k \in \partial i \setminus j} h_{ki}(t_k, t_i) \right\} \\
&= \psi_i(t_i) + \min \left\{ \sum_{k \in \partial i \setminus j} \min_{t_k < t_i} h^0_{ki}(t_k), \min_{k \in \partial i \setminus j} \left[\min \left\{ h^0_{ki}(t_i), \min_{t_k > t_i} h^1_{ki}(t_k) \right\} + \sum_{\ell \in \partial i \setminus j, k} \min_{t_\ell < t_i} h^0_{\ell i}(t_\ell) \right] \right\} \\
&= \psi_i(t_i) + \sum_{k \in \partial i \setminus j} \min_{t_k < t_i} h^0_{ki}(t_k) + \min \left\{ 0, \min_{k \in \partial i \setminus j} \left[\min \left\{ h^0_{ki}(t_i), \min_{t_k > t_i} h^1_{ki}(t_k) \right\} - \min_{t_k < t_i} h^0_{ki}(t_k) \right] \right\} . \qquad (38)
\end{aligned}$$

The equations (10-12) of the main text can now be readily obtained by defining the following quantities:

$$L_{ki}(t_i) = \min_{t_k < t_i} h^0_{ki}(t_k) , \qquad (39)$$

$$R_{ki}(t_i) = \min \left\{ h^0_{ki}(t_i), \min_{t_k > t_i} h^1_{ki}(t_k) \right\} , \qquad (40)$$

$$M_{ij}(t_i) = \min \left\{ 0, \min_{k \in \partial i \setminus j} \{ R_{ki}(t_i) - L_{ki}(t_i) \} \right\} , \qquad (41)$$

$$(42)$$

in terms of which the equations (37,38) can be rewritten

$$h^0_{ij}(t_i) = \psi_i(t_i) + \sum_{k \in \partial i \setminus j} L_{ki}(t_i) \qquad \text{for } 0 < t_i \leq T , \qquad (43)$$

$$h^1_{ij}(t_i) = \psi_i(t_i) + \sum_{k \in \partial i \setminus j} L_{ki}(t_i) + M_{ij}(t_i) \qquad \text{for } 0 < t_i \leq T , \qquad (44)$$

$$h^0_{ij}(0) = \psi_i(0) + \sum_{k \in \partial i \setminus j} R_{ki}(0) , \qquad (45)$$

the last equation corresponding to the unconstrained minimization over the removal times of the neighbors of a vertex i included in the decycling set.

Let us give a more explicit interpretation of the quantities L, R and of the last equations. $L_{ki}(t_i)$ is the minimum feasible cost in the subtree of $G \setminus i$ rooted in k with the only condition that $t_k < t_i$ (see Eq. (39)). On the other hand, in Eq. (40) we define $R_{ki}(t_i)$ to be the minimum feasible cost in the subtree of $G \setminus i$ rooted in k with $t_k \geq t_i$. As the message $h^1_{ij}(t_i)$ corresponds to a situation in which j has already been removed at time t_i, one of the neighbors $k \in \partial i \setminus j$ can be removed after i. It follows that for $t_i > 0$ the minimum feasible cost is given by the cost $\psi_i(t_i)$ plus the minimum between the minimum feasible cost when all neighbors $k \in \partial i \setminus j$ is removed before i and the same quantity when one of the neighbors is allowed to be removed at a later time.

The field $h_i(t_i)$ is then obtained from the messages (see Eqs. (32,33)) according to

$$h_i(t_i) = \psi_i(t_i) + \sum_{k \in \partial i} L_{ki}(t_i) + M_i(t_i) , \qquad (46)$$

$$h_i(0) = \psi_i(0) + \sum_{k \in \partial i} R_{ki}(0) , \qquad (47)$$

$$M_i(t_i) = \min\{0, \min_{k \in \partial i} \{ R_{ki}(t_i) - L_{ki}(t_i) \}\} . \qquad (48)$$

Finally a more efficient implementation can be devised, noting that common quantities can be pre-computed in order to obtain the h_{ij} for all the outgoing edges around a given vertex i. One indeed obtains an implementation

which runs in linear time both in T and in the degree k_i of i by defining

$$S_i^0(t_i) = \sum_{k\in\partial i} L_{ki}(t_i) \ , \tag{49}$$

$$S_i^1 = \sum_{k\in\partial i} R_{ki}(0) \tag{50}$$

$$k_i(t_i) \in \arg\min_{k\in\partial i}\{R_{ki}(t_i) - L_{ki}(t_i)\} \tag{51}$$

$$Q_i(t_i) = \min\left\{0, \min_{k\in\partial i\setminus k_i(t_i)}\{R_{ki}(t_i) - L_{ki}(t_i)\}\right\} \tag{52}$$

$$M_{ij}(t_i) = \begin{cases} M_i(t_i) & \text{if} \quad j \neq k_i(t_i) \\ Q_i(t_i) & \text{if} \quad j = k_i(t_i) \end{cases} \tag{53}$$

which can all be computed in time $O(Tk_i)$; we can then express the different values of the messages h_{ij} as

$$h_{ij}^0(t_i) = \psi_i(t_i) + S_i^0(t_i) - L_{ji}(t_i) \tag{54}$$

$$h_{ij}^1(t_i) = \psi_i(t_i) + S_i^0(t_i) - L_{ji}(t_i) + M_{ij}(t_i) \tag{55}$$

$$h_{ij}^0(0) = \psi_i(0) + S_i^1 - R_{ji}(0) \tag{56}$$

which can be also computed in time $O(T)$ for each $j \in \partial i$. The computation time for a complete iteration on all vertices i is thus $O(\sum_i k_i T) = O(|E|T)$. The computation of the field $h_i(t_i)$ in (46)-(47) is similar:

$$h_i(t_i) = \psi_i(t_i) + S_i^0(t_i) + M_i(t_i) \qquad \text{for} \ \ 0 < t_i \leq T \ , \tag{57}$$

$$h_i(0) = \psi_i(0) + S_i^1 \quad . \tag{58}$$

This derivation of the Min-Sum equations shows that the algorithm is exact on a tree: the recurrence equations on h_{ij} are guaranteed to converge, and the configuration t_i^* obtained from the MS expression of h_i is the unique minimum of the cost function over feasible configurations. One can, however, iterate the recurrence equations on h_{ij} for any graph, and use the MS formalism as an heuristic algorithm that provides a good approximation to the optimum, in particular when there are not many short loops. There are, however, two issues with the convergence of the message passing equations on h_{ij}:

- the h_{ij} defined above are extensive energies, that would grow indefinitely in presence of loops in the graph. This problem is easily cured by adding a constant value C_{ij} to all fields $h_{ij}(t_i, t_j)$, in such a way to keep the maximum entry of this matrix equal to a constant, for instance zero. This does not spoil the validity of the algorithm, as we only need informations about the relative energies of configurations to construct the decycling set: the optimum $t_i^* = \operatorname{argmin} h_i(t_i)$ is obviously invariant by a shift of the reference energy.
- even with this normalization the message passing equations are not guaranteed to converge. When they did not we enforced their convergence by employing a reinforcement procedure, that consists in taking $\psi_i(t_i) = \mathbb{I}[t_i = 0] + \varepsilon_i(t_i) + \tau\gamma h_i(t_i)$ where h_i is the local field computed with (57)-(58) in the previous iteration, γ is a small real value and τ is the iteration time. Typically we use in our simulation $\gamma = 10^{-3}$.

B. Tree-breaking in decycled graphs

We explain now the second stage of our algorithm, namely the dismantling of the acyclic graph obtained using MS in the first stage.

1. *Optimal tree breaking*

The computation of the C-dismantling number of a tree G can be performed in a time growing polynomially with C and with the size N of the graph, by the following dynamic programming approach.

Let us denote $G_{i\to j}$ the connected component of the vertex i in the graph obtained from G by removing one of its neighbors j, and call $S_{ij}(c)$ the minimum number of vertices to be removed from $G_{i\to j}$ to have that no component

of the reduced graph is larger than C and that the component of i is no larger than c. These quantities satisfy the following recursion:

$$\begin{aligned} S_{ij}(c) &= \min_{\substack{\{c_k\}_{k\in\partial i\setminus j} \\ \sum_{k\in\partial i\setminus j} c_k \leq c-1}} \sum_{k\in\partial i\setminus j} S_{ki}(c_k) \text{ if } 0 < c \leq C \ , \\ S_{ij}(0) &= 1 + \sum_{k\in\partial i\setminus j} S_{ki}(C) \ . \end{aligned}$$

Using max-convolutions (see e.g. [5]) these quantities can be computed on all directed edges of the tree in time $O(NC^2)$. By adding an extra leaf i' attached to a node i on the tree, the quantity $S_{ii'}(C)$ gives the decycling number. A small modification can be used to also find optimal dismantling sets in time $O(NC^2)$. Even though polynomial, this complexity is often too expensive in practice even for moderate values of C. Fortunately we will see below a greedy strategy that achieves almost the same performance.

2. Greedy tree-breaking

An alternative approach to the dismantling of a forest is to follow a greedy heuristic, removing iteratively the node in the largest connected component of the forest (i.e. a tree) that leaves the smallest largest component. This procedure is guaranteed to C-dismantle the forest by removing $S = N/C$ vertices or less. This ensures the dismantling to a sublinear size of the largest component $C \leq N/\log N$ by removing a sublinear number of vertices $S \leq \log N$. Moreover, it can be implemented in time $O(N(\log N + T))$ where T is the maximal diameter of the trees inside this forest. The worst case in terms of number of removed nodes is reached in the case of a one-dimensional chain, in which one needs to remove $S = 2^k - 1$ nodes to obtain components of size $C(S) \leq N/2^k$.

For a given tree G on N vertices, let us call F the subset of vertices i which are optimal in the above sense, namely such that the removal of i from G minimizes the size of the largest component of $G \setminus i$. The elements of F can be characterized in a very simple way. Denote by $C(i)$ the size of the largest component of $G\setminus i$, so $C(i) = \max_{j\in\partial i} |G_{j\to i}|$ and $F = \arg\min_{i\in V} C(i)$. Then $i^\star \in F$ if and only if $C(i^\star) \leq N/2$. Suppose indeed that for $i^\star \in F$, $C(i^\star) > N/2$ and take $j \in \partial i$ such that $|G_{j\to i^\star}| = C(i^\star)$. Then, as $|G_{i^\star\to j}|+|G_{j\to i^\star}| = N$, we have that $C(j) < \max\{N/2, C(i^\star)\} = C(i^\star)$ which is absurd. Conversely, suppose that $C(i) \leq N/2$ and take $i^\star \in F \setminus i$. Consider the unique path $(i, k_1, \ldots, k_n, i^\star)$ in G. Then $|G_{k_1\to i}| \leq C(i)$, and $|G_{i\to k_1}| \geq N - C(i) \geq N/2$. But $G_{i\to k_1} \subseteq G_{k_n\to i^\star}$ so $C(i^\star) \geq N/2$.

This characterization of F can be used constructively to find an $i^\star \in F$ efficiently. Pick for each connected component of the initial forest a "root" vertex $i_0 \in V$. For each i compute $w_i = |G_{i\to j}|$ where j is the unique neighbor of i on the path between i and the root i_0, starting from the leaves and exploiting the relation $w_i = 1 + \sum_{k\in\partial i\setminus j} w_k$; note that $C(i_0) = \max_{j\in\partial i_0} w_j$. Place i_0 into a priority queue with priority given by the component size $K(i_0) = 1 + \sum_{j\in\partial i_0} w_j$. Iteratively pick the largest component from the queue. Then construct the sequence i_t as follows: for every t, if $C(i_t) \leq N/2$, then $i^\star = i_t \in F$ and the process stops. Otherwise, iteratively choose i_{t+1} such that $w_{i_{t+1}} = C(i_t) > N/2$.

Once $i^\star$ is chosen and removed, the component is broken into $|\partial i^\star|$ components, each one rooted at $k \in \partial i^\star$. From these, only the component rooted at i_{t-1} needs to have its w values updated, as its orientation changed. The only needed adjustments are along the path $i_0, i_1, \ldots, i_t$ and can computed in time proportional to t, which is bounded by the diameter of the tree, which is in turn bounded by T.

As the cost of the priority queue updates scale as $O(\log N)$, the total number of operations for each vertex removal is thus $O(\log N+T)$, hence the total number of operations for greedily dismantling a forest scales as $O(N\cdot(\log N+T))$, as claimed above.

We performed an extensive comparison of the optimal and greedy procedure for values of C sufficiently small for the optimal one to be doable in a reasonable time, using as a benchmark the forest output by the MS algorithm applied to an Erdös-Rényi random graph of 78125 nodes and average degree 3.5. As shown in Fig. 1 we found the greedy strategy to have very close to optimal performances, therefore we used this much faster procedure in all other numerical simulations.

C. Greedy reintroduction of cycles

The initial condition for the reverse greedy procedure is the graph obtained after the removal from G of a set S^0 of nodes (dismantling set) and characterized by largest connected components of size C. Let us consider a target value $C' > C$ for the size of the largest connected components. As long as the size of the largest connected components

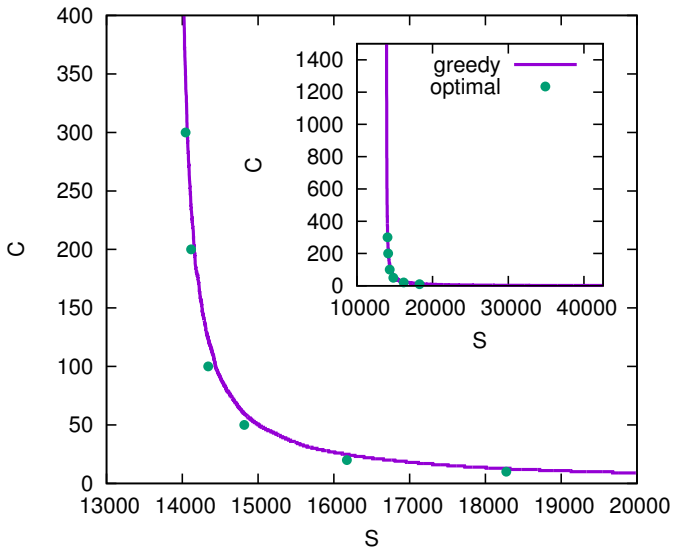

FIG. 1: Comparison between greedy and optimal forest-breaking on the output of the MS algorithm for an Erdös-Rényi random graph with 78, 125 vertices and average degree 3.5.

in the graph is smaller than C', the removed nodes are reintroduced one at a time by means of the following greedy strategy: at each iteration step t, we choose for reinsertion the node $i \in S^t$ (and the edges to vertices in $V \setminus S^t$) such that the connected component V_i^t the node i ends up in is the smallest possible. An efficient implementation of the greedy reinsertion is easily obtained by maintaining a priority queue of the removed vertices with priority given by the size $|V_i^t|$. When a vertex i is reintroduced in the graph, the number of connected components that get merged is at most equal to the degree k_i of the vertex (in the original graph G). The number of elements in the priority queue that have to be modified after the reinsertion of i is bounded by the number of nodes $j \in S^t$ that are connected to the new component in the original graph G. As the size of the largest component is at most C', this number is at most $k_{\max}C'$, where $k_{\max}$ is the maximal degree of the graph. The computational cost of reintroducing a vertex in the graph is entirely given by the one of updating the queue, which is thus bounded by $k_{\max}C' \log(k_{\max}C')$. In a sparse graph, the update cost is thus typically sublinear in N, making the reverse greedy strategy very efficient.

D. Competing algorithms

1. *Simulated Annealing*

Besides our main algorithm based on the Min-Sum procedure we have studied the network dismantling problem using simulated annealing, i.e. building a Monte Carlo Markov Chain that makes a random walk in the space of configurations of the subsets $S \subset V$ of removed vertices. We assign an energy to each configuration according to

$$\mathcal{E} = |S|\nu + C, \tag{59}$$

in which $|S|$ is the number of removed nodes, C the size of the largest connected component in the graph obtained by removing S, and ν is a free parameter. Note indeed that a set S of removed vertices can be considered "good" for two reasons: either because it is small, or because its removal fragments the graph into small components. These two figures of merits obviously contradict each other and cannot be optimized simultaneously, ν thus controls the balance between these two frustrating goals.

As usual in simulated annealing algorithms we introduce an inverse temperature β that is slowly increased during the evolution of the Markov Chain, and at each time step t we consider a move from the current configuration S^t to a new configuration S_{new}, that is accepted according to a standard Metropolis criterion, i.e. with probability

$$\min\left[1, e^{-\beta[\mathcal{E}_{\text{new}} - \mathcal{E}^t]}\right], \tag{60}$$

where $\mathcal{E}_{\text{new}}$ and $\mathcal{E}^t$ are the energies of S_{new} and S^t respectively. If the move is accepted we set $S^{t+1} = S_{\text{new}}$, $\mathcal{E}^{t+1} = \mathcal{E}_{\text{new}}$, otherwise the Markov Chain remains in the same configuration. The proposed configuration S_{new} is

constructed in the following way at each time step: a node i is chosen uniformly at random among all the N vertices of the graph, and its status is reversed (if $i \in S^t$ then $S_{\rm new} = S^t \setminus i$, if $i \notin S^t$ then $S_{\rm new} = S^t \cup \{i\}$). We then need to compute the energy $\mathcal{E}_{\rm new}$ of this proposed configuration; the first term is easily dealt with as $|S|$ varies by ± 1 depending on whether $i \in S^t$ or not. We thus only need to compute the size $C_{\rm new}$ of the largest component in the new configuration, facing three possible cases:

1. if $i \notin S^t$ and i belongs to the largest component of $G \setminus S^t$ the size $C_{\rm new}$ of the largest component is recomputed;
2. if $i \notin S^t$ but i does not belong to the largest component of $G \setminus S^t$ then $C_{\rm new} = C^t$ does not change;
3. if $i \in S^t$ then it is only necessary to compute the size C_i of the cluster i belongs to once it is reintroduced in the graph and compare the latter with the current largest component, i.e. $C_{\rm new} = \max(C^t, C_i)$.

The Markov chain is irreducible, recurrent and aperiodic, thus ergodic and the Metropolis criterion ensures detailed balance, therefore the SA algorithm would sample correctly the probability measure $\propto e^{-\beta\mathcal{E}}$ if run with an infinitesimally small annealing velocity. Unlike standard applications of simulated annealing, such as spin systems with short-range interactions, in the present problem a single move (node removal/reintroduction) can produce energy variations over a large range of scales, with the consequence that there is no natural criterion to choose the annealing protocol. We tested several different annealing protocols and we adopted one in which the inverse temperature β is increased linearly from $\beta_{\rm min}$ to $\beta_{\rm max}$ (thus concentrating the measure on close to ground-state configurations), with an increment of $d\beta$ at each time step (i.e. after each one attempted move). Protocols in which the inverse temperature is varied only after $O(N)$ attempted moves were also considered, with no relevant difference in the results. Similarly, there is no natural choice of the initial conditions. We tested the cases in which the initial set S^0 is empty and in which nodes are randomly assigned to S^0 independently with probability 1/2, but for sufficiently small values of $\beta_{\rm min}$, different choices had no relevant effects on the optimization process.

Fig. 2 displays the minimum energy achieved using the SA algorithm (with $\nu = 0.6$) on Erdös-Rényi random graphs of average degree $d = 3.5$ and increasing sizes from $N = 1024$ to $N = 16384$. For comparison we also plot the results obtained using the Min-Sum algorithm (horizontal lines). For small sizes, the SA algorithm outperforms Min-Sum when the annealing scheme is sufficiently slow ($d\beta$ very small). Increasing N, the quality of the results obtained with SA degrades, as it would require an increasingly slower annealing protocol in order to achieve the same results obtained using Min-Sum. These results show that, even though the SA implementation proposed is simple and relatively fast even on large networks, the necessity of an increasingly slower annealing protocol prevents SA from reaching optimal results in a reasonable computational time.

The results for simulated annealing presented in Fig. 1 and Fig. 4 of the main text are obtained with parameters $d\beta = 10^{-8}$, $\beta_{\rm min} = 0.5$ for both, $\beta_{\rm max} = 20$, $\nu = 1.2$ for Fig. 1, and $\beta_{\rm max} = 10$, $\nu = 2.0$ for Fig. 4.

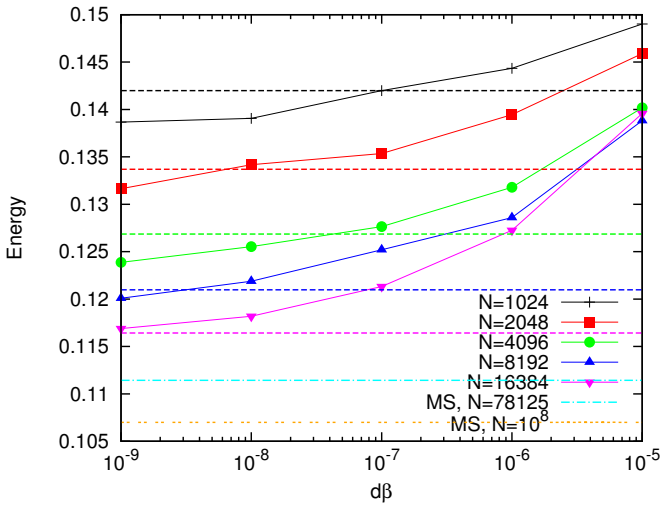

FIG. 2: Energy $\mathcal{E}$ as function of the rate $d\beta$ used to increase β from $\beta_{\rm min} = 0.5$ to $\beta_{\rm max} = 30$ in the SA algorithm ($\nu = 0.6$) used to dismantle Erdös-Rényi random graphs of average degree $d = 3.5$ and increasing size from $N = 1024$ (top) to $N = 16384$ (bottom). We also report the minimum energy achieved with the Min-Sum (MS) algorithm on the same graphs and for graphs of larger sizes $N = 78125$ (cyan line) and $N = 10^8$ (orange line).

2. Score-based algorithms

Let us give here more details about the other dismantling algorithms to which we compared our own proposals. They all proceed by the (irreversible) removal of nodes from the graphs to be dismantled, the differences between them relying in the choice of a score function that assigns to each vertex i of the graph a score e_i, the vertices being removed in the order of decreasing scores (with random choices in case of ties). This quantity e_i should be an heuristic measure of the importance, or centrality, of the vertex i, in the sense that more central nodes should lead to a larger decrease in the size of the largest component when i is removed. We have investigated the following score functions, the names corresponding to the key in the figures:

- RND, $e_i = 1$ for all i; this leads to a random choice of the removed vertices, i.e. to classical site percolation.
- DEG, $e_i = k_i$ the degree of node i, this corresponds to removing the highest degree nodes first.
- EC, for eigenvector centrality, uses as a score the eigenvector e_i associated to the largest eigenvalue λ of the adjacency matrix $A_{ij} = \mathbb{I}[\langle i, j\rangle \in E]$ of the graph, in other words the solution of the linear system of equations

$$\lambda\, e_i = \sum_{j \in \partial i} e_j \ . \tag{61}$$

  For a connected graph the Perron-Frobenius theorem ensures that this eigenvector is unique and that it can be chosen positive.

- CI_ℓ, for collective influence at level ℓ, is a centrality measure introduced by Morone and Makse [6] to provide a heuristic measure of the influence that a node has on the neighbors within a certain distance ℓ from it. The collective influence of node i at level ℓ is defined as

$$\mathrm{CI}_\ell(i) = (k_i - 1) \sum_{j \in \partial B(i,\ell)} (k_j - 1) \tag{62}$$

  where $\partial B(i, \ell)$ denotes the set formed by all the nodes that are at distance ℓ from node i [6]. The CI value of node i takes two contributions, the degree of node i and the number of edges emerging at distance ℓ from a ball surrounding i. On expander graphs, such as random graphs, the number of nodes contained in a ball $B(i, \ell)$ grows exponentially with ℓ, hence the calculation of the collective influence scores for all nodes of the graph becomes computationally demanding already for moderately small distance values ($\ell = 4, 5$).

We also made some tests with the score function defined as the betweenness centrality and as the non-backtracking centrality [7], but for the graphs we considered the results we obtained were both qualitatively and quantitatively similar (or worse) to those obtained using EC, hence we do not report them.

For a given score function one can envision different ways to implement the dismantling algorithm; the simplest would be to compute the scores for all vertices of the original graph, and then to remove the vertices in the order defined by this ranking. We used instead an adaptive version, which gives much better results, that consist in recomputing the scores of all remaining vertices after each removal of the node with highest score in the current graph; all the results presented in the main text and the SI have been obtained in this way. Even if it performs better this adaptive strategy is also much more computationally demanding; an intermediate compromise between these two extreme strategies would be to recompute the scores only after a finite fraction x of nodes is removed. Another implementation twist consists in recomputing the scores only for the vertices belonging to the currently largest connected components, as the removal of a vertex outside it would not decrease the size of the largest component. This is useful in particular if one tries to compute the EC scores by the power method (multiplying several times an initial guess by the adjacency matrix); instead of the full adjacency matrix one can consider only the submatrix corresponding to the vertices in the largest component. By construction this submatrix is irreducible and the power method will converge, hence solving the possible convergence issues encountered by the power method in the case of coexistence of several connected components in the graph. The restriction to the largest component modifies also the behavior of the DEG heuristic, as it avoids the removal of large degree nodes in already small components.

IV. OTHER REAL-WORLD AND SCALE-FREE GRAPHS

We already explained in the main text that dismantling a graph by means of the decycling (plus greedy tree breaking) is guaranteed to be optimal only for sparse random graphs with locally tree-like structure. Nevertheless,

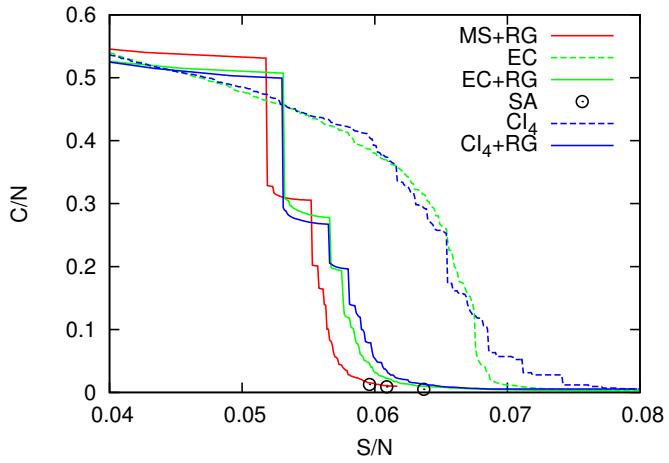

FIG. 3: Size of the largest component in scale-free random graphs, of size $N = 10^4$ nodes and degree distribution $P(d) \propto d^{-\gamma}$ with $\gamma = 2.5$, as function of the fraction of removed nodes S. The nodes are removed using adaptive Eigenvector Centrality (EC), adaptive Eigenvector Centrality plus Reverse Greedy (EC+RG), Collective Influence with diameter $\ell = 4$ (CI_4), the same plus Reverse Greedy (CI_4+RG), Min-Sum plus Reverse Greedy (MS+RG), and Simulated Annealing (SA) with $d\beta = 10^{-8}$ and $\beta_{\min} = 0.5, \beta_{\max} = 20$ (and several values of ν).

we observed that when the algorithm is complemented by a simple reverse greedy (RG) strategy the final result is usually very good also on networks in which many small loops are present, such as in the case of the Twitter graph in Fig. 4 in the main text. Our way to state the quality of the result is the direct comparison with the other available algorithms, that are the Simulated Annealing algorithm and the other heuristics (e.g. EC, CI) also complemented by the RG strategy.

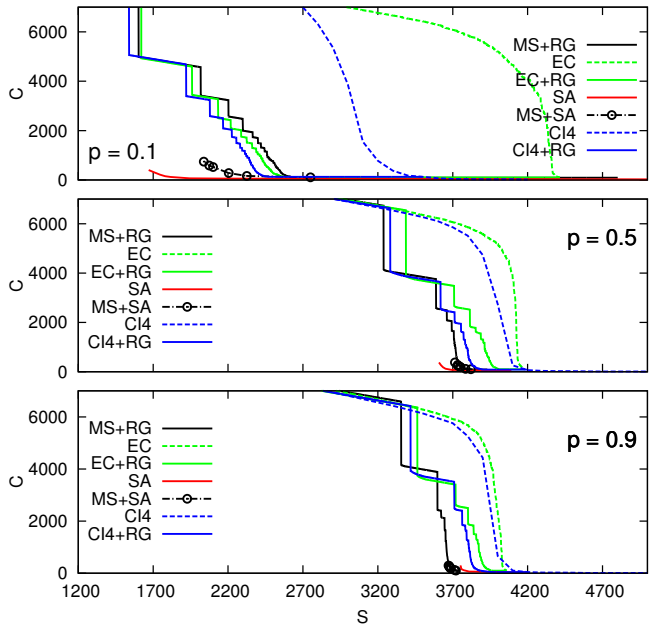

FIG. 4: Size of the largest component in WS small-world networks with rewiring probability $p = 0.1, 0.5, 0.9$ achieved by removing a fraction of nodes using Eigenvector Centrality (EC), Eigenvector Centrality plus Reverse Greedy (EC+RG), Collective Influence with diameter $\ell = 4$ (CI4), Collective Influence with diameter $\ell = 4$ plus Reverse Greedy (CI4+RG), Min-Sum (MS), Min-Sum plus Reverse Greedy (MS+RG), Min-Sum plus SA (MS+SA), and Simulated Annealing (SA) with $d\beta = 10^{-8}$ and $\beta_{\min} = 0.5, \beta_{\max} = 20$ (and several values of ν).

We studied dismantling in the youtube network [8] with 1.13 million nodes and concluded that the reverse greedy is of immense importance here. Specifically we obtained that in order to dismantle the network into components smaller that $C = 1000$ nodes the CI methods removes 5.12%, the ER removes 4.97%, the MS removes 5.67% nodes. The reverse greedy procedure improves all the these methods and gets dismantling sizes 4.03% for CI+RG, 4.07% for EC+RG, and 3.97% for MS+RG.

We also studied dismantling on an example of a synthetic scale-free network. Results reported in Fig. 3, are qualitatively comparable to the ones for real networks.

In order to better quantify the effect of a large clustering coefficient on the different algorithmic methods under study, we considered a well-known class of random graphs with tunable clustering coefficient, the small-world network model introduced by Watts and Strogatz [9]. The WS network is generated starting from a one-dimensional lattice in which every node is connected with $d/2$ nearest-neighbors on both sides, then each edge (i, j) with $i > j$ is rewired to a randomly chosen node $k \neq j$ with probability p. Fig. 4 shows the result of dismantling WS networks of size $N = 10^4$, $d = 6$ and rewiring probability $p = 0.1, 0.5, 0.9$. For $p = 0.9$ the WS network is topological similar to a random graph, with very small clustering coefficient, because almost all edges have been rewired. On this network, Min-Sum plus reverse greedy outperforms centrality-based heuristics (EC+RG and CI+RG) and gives results that are comparable with the best obtained using SA. For $p = 0.5$, MS+RG still gives a very good result, only slightly worse than SA. We also replaced the reverse greedy procedure with a reverse Monte Carlo method, in which a dismantling set is sought by performing the SA algorithm from the solution of the MS algorithm, by keeping only an optimal subset of the nodes already removed. The replacement of the reverse greedy procedure with a Monte Carlo based method gives improved results for both $p = 0.9$ and $p = 0.5$. We stress that this could be another useful strategy to improve heuristic results even in large networks, because the SA algorithm runs on a fraction of the original graph.

When p is further decreased, the structure of the WS network significantly departs from that of a random graph and short loops start to play a very important role, it is clear that we do not expect decycling to be a good strategy for dismantling in this regime. In this regime SA performs about 30% better than any other algorithm, even though complemented with the reverse greedy strategy. When we perform SA from the solutions obtained using MS, the results are improved but still far from the best results obtained using SA alone. This is due to the fact that, in clustered networks, the dismantling set obtained by SA is not a subset of the dismantling set obtained using any other heuristic strategy, with an overlap that is usually small.

[1] Guggiola A, Semerjian G (2015) Minimal contagious sets in random regular graphs. *J. Stat. Phys.* 158(2):300–358.

[2] Beineke LW, Vandell RC (1997) Decycling graphs. *Journal of Graph Theory* 25(1):59–77.

[3] Bau S, Wormald NC, Zhou S (2002) Decycling numbers of random regular graphs. *Random Structures & Algorithms* 21(3-4):397–413.

[4] Haxell P, Pikhurko O, Thomason A (2008) Maximum acyclic and fragmented sets in regular graphs. *Journal of Graph Theory* 57(2):149–156.

[5] Baldassi C, Braunstein A (2015) A Max-Sum algorithm for training discrete neural networks. *J. Stat. Mech.* 2015(8):P08008.

[6] Morone F, Makse HA (2015) Influence maximization in complex networks through optimal percolation. *Nature* 524(7563):65–68.

[7] Martin T, Zhang X, Newman M (2014) Localization and centrality in networks. *Physical Review E* 90(5):052808.

[8] Leskovec J, Krevl A (2014) SNAP Datasets: Stanford large network dataset collection (`http://snap.stanford.edu/data`).

[9] Watts DJ, Strogatz SH (1998) Collective dynamics of small-world networks. *Nature* 393(6684):440–442.