PNAS
www.pnas.org

Supplementary Information for

Generalized Network Dismantling

Xiao-Long Ren, Niels Gleinig, Dirk Helbing and Nino Antulov-Fantulin

Nino Antulov-Fantulin
E-mail: anino@ethz.ch

This PDF file includes:

Supporting Information Text

1. Objective function

Let us assume that we want to partition the network $G = (V, E)$ into two parts such that the nodes from a set $M \subseteq V$ are not connected to the nodes from the complement set $\overline{M} = V \setminus M$. The indication of node i belonging to the set $M \subseteq V$ is denoted with the following vector

$$v_i := \begin{cases} +1 & i \in M, \\ -1 & i \in \overline{M} \end{cases} \quad [1]$$

The classical spectral bisection of a graph aims to minimize the number of edges between the clusters M and $\overline{M}$. However, in this paper we propose a novel node–weighted spectral cut objective function, in which the cost of cutting the edge (i, j) is equal to the sum of the weights of i and j. That is, for any specific edge (i, j), which separates nodes from different clusters M and $\overline{M}$, we define the cost of removing this edge as $-\frac{1}{2}(v_i v_j - 1) A_{i,j} (w_i + w_j)$. If the edge (i, j) separates the clusters, the associated cost is $w_i + w_j$ and zero otherwise.

If the cost is proportional to the degree centrality $w_i \propto d_i$, then the cost of removing a subset of nodes that are incident to the edges that separate cluster M from cluster $\overline{M}$ is:

$$\frac{1}{2} \sum_{i,j} -\frac{1}{2} (v_i v_j - 1) A_{i,j} (d_i + d_j - 1). \quad [2]$$

The term $(d_i + d_j - 1)$ contains the element-1, in order to correct double counting of the link that connects i and j and furthermore leads to more elegant notation in unit case scenario.

$$Wcut(M) = -\frac{1}{4} \sum_{i,j} (v_i v_j - 1) A_{i,j} (d_i + d_j - 1) =$$

$$\frac{1}{4} \sum_{i,j} (1 - v_i v_j) A_{i,j} (d_i + d_j - 1). \quad [3]$$

Now, we denote the matrix $A_{i,j} (d_i + d_j - 1)$ as $B_{i,j}$.

For **general weights**, the matrix element $B_{i,j} = A_{i,j} (w_i + w_j - 1)$, where the constant term -1 is added to all weights. This constant term is not affecting the optimization but was added for several reasons: (i) as in specific case of unit costs leads to simplified notation and (ii) unified notation with degree costs. In the following, we work with re-weighted adjacency matrix $B_{i,j}$. Then objective function, can be re-written by using the re-weighted adjacency matrix $B_{i,j}$

$$Wcut(M) = \frac{1}{4} \sum_{i,j} B_{i,j} - \frac{1}{4} \sum_{i,j} v_i v_j B_{i,j} = \frac{1}{4} \sum_{i} d_i^B - \frac{1}{4} \sum_{i,j} v_i v_j B_{i,j} =$$

$$\frac{1}{4} \sum_{i,j} d_{i,j}^B \delta_{i,j} v_i v_j - \frac{1}{4} \sum_{i,j} v_i v_j B_{i,j} = \frac{1}{4} \sum_{i,j} v_i (d_{i,j}^B \delta_{i,j} - B_{i,j}) v_j, \quad [4]$$

where d_i^B is the weighted degree of node v_i from the matrix $B_{i,j}$ and $\delta_{i,j}$ equals 1 when $i = j$ and 0 otherwise.

Finally, in matrix notation

$$Dcut(A) := \frac{1}{4} v^t L_w v \quad [5]$$

where $L_w = D_B - B$, $B = AW + WA - A$, where D and D_B are diagonal matrices with elements: $(D)_{ii} = \sum_{j=1}^{n} A_{ij}$, $(D_B)_{ii} = \sum_{j=1}^{n} B_{ij}$ respectively. If $W = I$, the problem becomes normal spectral partitioning as $B = A$. Note, that our methodology works even if we don't add constant term -1 to objective function, just minor modification is needed $B = AW + WA$.

Now, this objective function has the same form as the classical bisection problem (1–3), where the Laplacian is replaced with the node–weighted Laplacian matrix. Thus it is analytically solved by the second smallest eigenvector of the node–weighted Laplacian $\lambda_2 v^{(2)} = L_w v^{(2)}$ due to the Courant-Fisher theorem.

2. Spectral properties of weighted Laplacian

L_w is a real and symmetric matrix. Therefore it has real eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ corresponding to eigenvectors $v_1, ..., v_n$ which form an orthonormal basis of $\mathbb{R}^n$. We will now show that the largest eigenvalue λ_n of the matrix $L_w = D_B - B$ satisfies $|\lambda_n| \leq 4 d_{max}(w_{max} + 1)$, where w_{max} is the largest cost in diagonal cost matrix W and d_{max} is the maximal degree of any node of the network. For this purpose let B_{abs} denote the matrix whose entries are the absolute values of the entries of the matrix B. Then for vectors v with $\|v\|_1 = 1$, we can use the triangle inequality and the fact that the L_1-norm is submultiplicative (meaning $\|XY\|_1 \leq \|X\|_1 \|Y\|_1$ for matrices X, Y with appropiate dimensions) to obtain

$$\begin{aligned} &\| B_{abs} v \|_1 \\ &\leq \| A \|_1 \| W \|_1 \|v\|_1 + \| W \|_1 \| A \|_1 \|v\|_1 + \|A\|_1 \|v\|_1 \\ &\leq 2 d_{max}(w_{max} + 1) \end{aligned} \quad [6]$$

where we have also used $\|A\|_1 \leq d_{max}$ and $\|W\|_1 \leq w_{max}$. It follows that

$$\begin{aligned} |\lambda_n| &\leq \max_{\|v\|_1=1} \|(D_B - B)v\|_1 \\ &= \max_{\|v\|_1=1} \sum_{i=1}^{n} |v_i \sum_{j=1}^{n} B_{ij} - \sum_{j=1}^{n} v_j B_{ij}| \\ &\leq \max_{\|v\|_1=1} \sum_{i=1}^{n} \sum_{j=1}^{n} |v_i B_{ij}| + \sum_{i=1}^{n} \sum_{j=1}^{n} |v_j B_{ij}| \\ &= \max_{\|v\|_1=1} \|B_{abs} v\|_1 + \|B_{abs} v\|_1 \\ &\leq 4 d_{max}(w_{max} + 1). \end{aligned} \quad [7]$$

However, for simplicity we will use the following bound $\lambda_n \leq 6 \cdot d_{max}^2$, when $w_{max} = d_{max}$. This bound is obtained since this holds: $4d_{max}(w_{max} + 1) = 4d_{max}^2 + 4d_{max} \leq 4d_{max}^2 + 2d_{max}^2$ as for every connected network with more than 2 nodes $4d_{max} \leq 2d_{max}^2$.

3. Convergence bounds

L_w has real eigenvalues $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ corresponding to eigenvectors $v_1, ..., v_n$ which form an orthonormal basis of $\mathbb{R}^n$. From the previous section, we know that $0 = \lambda_1$ and $\lambda_n \leq 6 \cdot d_{max}^2$, where d_{max} is the maximal degree of any node of the network. So in order to compute v_2 we consider the matrix $\tilde{L} = 6 \cdot d_{max}^2 \cdot I - L_w$, which has the same eigenvectors $v_1, ..., v_n$ as L_w. Now the corresponding eigenvalues are $\tilde{\lambda_1} = 6 \cdot d_{max}^2 \geq ... \geq \tilde{\lambda_n} = 6 \cdot d_{max}^2 - \lambda_n \geq 0$ and in particular v_1 corresponds to the largest eigenvalue and v_2 to the second largest eigenvalue.

If v is a random vector uniformly drawn from the unit sphere and we force it to be perpendicular to $v_1 = c \cdot (1, ..., 1)^T$ by setting $v \leftarrow v - \frac{v_1^T v}{v_1^T v_1} \cdot v_1$, then $v = \psi_2 v_2 + ... + \psi_n v_n$ and $\psi_2 \neq 0$ almost surely.

Furthermore $\tilde{L}v = \tilde{\lambda_2}\psi_2 v_2 + ... + \tilde{\lambda_n}\psi_n v_n$ and if we set $v^{(k)} := \tilde{L}^k v$, then

$$\begin{aligned} \frac{v^{(k)}}{\|v^{(k)}\|} &= \frac{\tilde{\lambda_2}^k \psi_2 v_2 + ... + \tilde{\lambda_n}^k \psi_n v_n}{\|\tilde{\lambda_2}^k \psi_2 v_2 + ... + \tilde{\lambda_n}^k \psi_n v_n\|} \\ &= \frac{\psi_2 v_2 + \left(\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}\right)^k \psi_3 v_3 + ... + \left(\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}\right)^k \psi_n v_n}{\|\psi_2 v_2 + \left(\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}\right)^k \psi_3 v_3 + ... + \left(\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}\right)^k \psi_n v_n\|} \end{aligned} \quad [8]$$

converges with exponential speed to some eigenvector of L with eigenvalue λ_2, because for every i with $\lambda_i > \lambda_2$ we have $|\frac{\tilde{\lambda_i}}{\tilde{\lambda_2}}| < 1$ and therefore $\left(\frac{\tilde{\lambda_i}}{\tilde{\lambda_2}}\right)^k \psi_i v_i \to 0$ with exponential speed. Hence if $\lambda_3 > \lambda_2$, then this sequence converges to $+v_2$ or $-v_2$. And if $\lambda_2 = \lambda_3 = ... = \lambda_k < \lambda_{k+1}$, then this sequence converges to a unit length linear combination of $v_2, ..., v_k$ and is therefore a vector which minimizes $\frac{v^T L_w v}{v^T v}$ (which is the quantity associated to the cut-size) among all vectors which are orthogonal to v_1. More generally one can deduce from (8) that

$$\frac{{v^{(k)}}^T L_w v^{(k)}}{{v^{(k)}}^T v^{(k)}} = \frac{\lambda_2 |\psi_2|^2 + \lambda_3 |\left(\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}\right)^k \psi_3|^2 + ... + \lambda_n |\left(\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}\right)^k \psi_n|^2}{|\psi_2|^2 + |\left(\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}\right)^k \psi_3|^2 + ... + |\left(\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}\right)^k \psi_n|^2} \quad [9]$$

and therefore this quantity converges to λ_2 with exponential speed.

From above we deduce the following algorithm.

Algorithm for computing the partition vector

Input: Network

Output: Vector v close to eigenvector v_2 for which $\frac{v^T L_w v}{v^T v}$ is close to λ_2.

1. Draw v randomly with uniform distribution on the unit sphere.
2. Set $v = v - \frac{v_1^T v}{v_1^T v_1} \cdot v_1$.
3. For $i = 1$ to $\eta(n)$
   $v = \frac{\tilde{L}v}{\|\tilde{L}v\|}$

Note that sampling a point $p = (p_1, ..., p_n)$ from the unit sphere (4) with uniform distribution can be done by sampling $x_1, ..., x_n$ independently with standard normal distribution and setting

$$p_i = \frac{x_i}{\sqrt{x_1^2 + ... + x_n^2}}. \tag{10}$$

To prove this one only needs to check $\|p\| = 1$ and that the distribution of p is rotation invariant, that is for any $g \in SO(n)$ the random vectors p and gp have the same distribution. The latter follows immediately from the rotation invariance of the multivariate normal distribution.

Theorem 1 *If v is the output of the previous algorithm, then*

a)

$$|\lambda_2 - \frac{v^T L_w v}{v^T v}| \leq \frac{12 \cdot d_{max}^2 \cdot n}{\psi_2^2} \cdot |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} = \frac{12 \cdot d_{max}^2 \cdot n}{\psi_2^2} \cdot |\frac{6 \cdot d_{max}^2 - \lambda_3}{6 \cdot d_{max}^2 - \lambda_2}|^{2\eta(n)} \tag{11}$$

b) If $\eta(n)$ grows assymptotically faster than

$$\frac{11}{2 \log\left(|\frac{\tilde{\lambda_2}}{\lambda_3}|\right)} \log(n), \tag{12}$$

then

$$\mathbb{E}\left[|\lambda_2 - \frac{v^T L_w v}{v^T v}|\right] \to 0. \tag{13}$$

In particular, this theorem implies that if we choose $\eta(n) = O(\log(n)^{1+\epsilon})$ for any $\epsilon > 0$, then we might expect assymptotically good results. Here, we assume that the $1/log\left(|\frac{\tilde{\lambda_2}}{\lambda_3}|\right)$ is bounded by a constant as the network grows.

In order to prove this theorem we recall that a beta random variable with parameters $p, q > 0$, denoted $\beta_{p,q}$, is a random variable which takes values in $[0, 1]$ and whose probability density function is given by

$$\frac{\Gamma(\frac{p-1}{2})\Gamma(\frac{q-1}{2})}{\Gamma(\frac{p+q-1}{2})} x^{p-1}(1-x)^{q-1}, \tag{14}$$

where Γ denotes the gamma function.

If $x_1, ..., x_n$ are independent standard normal random variables, then

$$\frac{x_i^2}{x_1^2 + ... + x_n^2} \sim \beta_{\frac{1}{2}, \frac{n-1}{2}}. \tag{15}$$

Together with (10), this implies that when $p = (p_1, ..., p_n)$ is uniformly distributed on the unit sphere, then

$$p_i^2 \sim \beta_{\frac{1}{2}, \frac{n-1}{2}}. \tag{16}$$

Proof: a) If v is the output of the algorithm, then we can use (9) with $k = \eta(n)$ to obtain

$$\begin{aligned}
|\lambda_2 - \frac{v^T L_w v}{v^T v}| &= |\lambda_2 - \frac{\lambda_2 |\tilde{\lambda_2}|^{2\eta(n)} \psi_2^2 + ... + \lambda_n |\tilde{\lambda_n}|^{2\eta(n)} \psi_n^2}{|\tilde{\lambda_2}|^{2\eta(n)} \psi_2^2 + ... + |\tilde{\lambda_n}|^{2\eta(n)} \psi_n^2}| \\
&= |\lambda_2 - \frac{\lambda_2 \psi_2^2 + \lambda_3 |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_3^2 + ... + \lambda_n |\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_n^2}{\psi_2^2 + |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_3^2 + ... + |\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_n^2}| \\
&\leq |\lambda_2 - \frac{\lambda_2 \psi_2^2}{\psi_2^2 + |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_3^2 + ... + |\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_n^2}| + |\frac{\lambda_3 |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_3^2 + ... + \lambda_n |\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_n^2}{\psi_2^2 + |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_3^2 + ... + |\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_n^2}|
\end{aligned} \tag{17}$$

where in the last inequality we have used the triangle inequality. Pulling out λ_2 in the first term and replacing the denominator of the second term by ψ_2^2 (which is $\leq$ than the denomiator that we had before and hence makes this fraction larger), we obtain

$$\begin{aligned}
|\lambda_2 - \frac{v^T L_w v}{v^T v}| &\leq |\lambda_2| |\frac{\psi_2^2 + |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_3^2 + ... + |\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_n^2}{\psi_2^2 + |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_3^2 + ... + |\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_n^2} - \frac{\psi_2^2}{\psi_2^2 + |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_3^2 + ... + |\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_n^2}| \\
&\quad + |\frac{\lambda_3 |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_3^2 + ... + \lambda_n |\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_n^2}{\psi_2^2}| \\
&= |\lambda_2| |\frac{|\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_3^2 + ... + |\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_n^2}{\psi_2^2 + |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_3^2 + ... + |\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_n^2}| + |\frac{\lambda_3 |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_3^2 + ... + \lambda_n |\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}|^{2\eta(n)} \psi_n^2}{\psi_2^2}|.
\end{aligned} \tag{18}$$

Using $\psi_i^2 \leq 1$ and replacing the denomiator of the first term by ψ_2^2 we obtain

$$\begin{aligned} |\lambda_2 - \frac{v^T L_w v}{v^T v}| &\leq |\lambda_2||\frac{|\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} + ... + |\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}|^{2\eta(n)}}{\psi_2^2}| + |\frac{\lambda_3|\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} + ... + \lambda_n|\frac{\tilde{\lambda_n}}{\tilde{\lambda_2}}|^{2\eta(n)}}{\psi_2^2}| \\ &\leq |6 \cdot d_{max}^2||\frac{n \cdot |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)}}{\psi_2^2}| + |\frac{6 \cdot d_{max}^2 n \cdot |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)}}{\psi_2^2}| = \frac{12 \cdot d_{max}^2 \cdot n}{\psi_2^2} \cdot |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)} \end{aligned} \quad [19]$$

where the last inequality is due to $|\frac{\tilde{\lambda_i}}{\tilde{\lambda_2}}|^{2\eta(n)} \leq |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)}$ and $\lambda_i \leq 6 \cdot d_{max}^2$.

b) Let v^0 be drawn randomly with uniform distribution of the unit sphere of $\mathbb{R}^n$ and make it orthogonal to v_1 by setting $v^0 \leftarrow v^0 - v_1^T v^0 \cdot v_1$ and let it have unit-length by setting $v^0 \leftarrow \frac{v_0}{\|v_0\|}$. Then for the representation

$$v^0 = \psi_2 v_2 + ... + \psi_n v_v \quad [20]$$

of v^0 in the basis of the eigenvectors $v_1, v_2, ..., v_n$, it holds that $(\psi_2, ..., \psi_n)$ is distributed uniformly on the unit sphere of $\mathbb{R}^{n-1}$. Therefore ψ_2^2 is distributed like a beta variable $\beta_{\frac{1}{2}, \frac{n-2}{2}}$ and we have

$$\mathbb{P}\left(\psi_2^2 < \frac{1}{n^8}\right) = \frac{\Gamma(\frac{1}{2})\Gamma(\frac{n-2}{2})}{\Gamma(\frac{n-1}{2})} \int_0^{\frac{1}{n^8}} x^{-\frac{1}{2}}(1-x)^{\frac{n-4}{2}} dx \leq \int_0^{\frac{1}{n^8}} x^{-\frac{1}{2}} dx = 2\frac{1}{n^4}, \quad [21]$$

where the inequality is due to $(1-x)^{\frac{n-4}{2}} \leq 1$ and $\frac{\Gamma(\frac{1}{2})\Gamma(\frac{n-2}{2})}{\Gamma(\frac{n-1}{2})} \leq 1$.
Since

$$\mathbb{E}\left[|\lambda_2 - \frac{v^T L_w v}{v^T v}|\right] =$$

$$= \mathbb{P}\left(\psi_2^2 < \frac{1}{n^8}\right) \mathbb{E}\left[|\lambda_2 - \frac{v^T L_w v}{v^T v}| \,\middle|\, \psi_2^2 < \frac{1}{n^8}\right] + \mathbb{P}\left(\psi_2^2 \geq \frac{1}{n^8}\right) \mathbb{E}\left[|\lambda_2 - \frac{v^T L_w v}{v^T v}| \,\middle|\, \psi_2^2 \geq \frac{1}{n^8}\right] \quad [22]$$

we can use the trivial bound $\mathbb{P}\left(\psi_2^2 \geq \frac{1}{n^8}\right) \leq 1$ and (21) to obtain

$$\mathbb{E}\left[|\lambda_2 - \frac{v^T L_w v}{v^T v}|\right] \leq 2\frac{1}{n^4}\mathbb{E}\left[|\lambda_2 - \frac{v^T L_w v}{v^T v}| \,\middle|\, \psi_2^2 < \frac{1}{n^8}\right] + \mathbb{E}\left[|\lambda_2 - \frac{v^T L_w v}{v^T v}| \,\middle|\, \psi_2^2 \geq \frac{1}{n^8}\right]. \quad [23]$$

Since $0 \leq \frac{v^T L_w v}{v^T v} \leq 6 \cdot d_{max}^2 \leq 6n^2$ and $0 \leq \lambda_2 \leq 6 \cdot d_{max}^2 \leq 6n^2$ we have

$$\mathbb{E}\left[|\lambda_2 - \frac{v^T L_w v}{v^T v}| \,\middle|\, \psi_2^2 < \frac{1}{n^8}\right] \leq 6n^2. \quad [24]$$

From part a) we obtain

$$\mathbb{E}\left[|\lambda_2 - \frac{v^T L_w v}{v^T v}||\psi_2^2 \geq \frac{1}{n^8}\right] \leq \mathbb{E}\left[\frac{12 d_{max}^2 n}{\psi_2^2} \cdot |\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)}|\psi_2^2 \geq \frac{1}{n^8}\right] \leq 12n^{11}|\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)}. \quad [25]$$

Plugging these bounds into (23) we obtain

$$\mathbb{E}\left[|\lambda_2 - \frac{v^T L_w v}{v^T v}|\right] \leq 12\frac{1}{n^2} + 12n^{11}|\frac{\tilde{\lambda_3}}{\tilde{\lambda_2}}|^{2\eta(n)}. \quad [26]$$

In particular

$$\mathbb{E}\left[|\lambda_2 - \frac{v^T L_w v}{v^T v}|\right] \to 0 \quad [27]$$

whenever $\eta(n)$ grows assymptotically faster than

$$\frac{11}{2\log\left(|\frac{\tilde{\lambda_2}}{\lambda_3}|\right)} \log(n). \quad [28]$$

4. Run time complexity

The complexity of the spectral bisection algorithm is the complexity of computing v_2 or some other vector v for which $\frac{v^T L_w v}{v^T v}$ is close to minimal and which is orthogonal to DI where I is an identity matrix. We compute that vector with the algorithm of appendix A.
The complexity of this algorithm equals the number of iterations $\eta(n)$ times the complexity of multiplying $\tilde{L}$ and v. That is $O(\eta(n) \cdot n \cdot \bar{d})$ where $\bar{d}$ is the average degree of the network, or equivalently $O(|E| \cdot \eta(n))$ where $|E|$ is the number of edges.

The complexity of running the hierarchical spectral clustering algorithm until we have $|GCC| = O(1)$ is then given by the sum of:

- The complexity of applying spectral bisection once on the whole network. $\rightarrow O(\eta(n) \cdot n \cdot \bar{d})$.
- The complexity of applying it on each of the 2 clusters that we obtained from the first application of spectral bisection and which will have size approximately $\frac{n}{2}$.
- The complexity of applying it on each of the 4 clusters that we obtained from the previous step and which will have size approximately $\frac{n}{4}$.
- ...
- The complexity of applying it on each of the $\frac{n}{2} = 2^{\log_2(n)-1}$ clusters that we obtained from the previous step and which will have size approximately $\frac{n}{2^{\log_2(n)-1}} = 2$.

That is in total

$$\begin{aligned} & O(\eta(n) \cdot n \cdot \bar{d}) + 2 \cdot O(\eta(n) \cdot \frac{n}{2} \cdot \bar{d}) + ... + 2^{\log_2(n)-1} \cdot O(\eta(n) \cdot \frac{n}{2^{\log_2(n)-1}} \cdot \bar{d}) \\ &= \sum_{i=0}^{\log_2(n)-1} 2^i \cdot O(\eta(n) \cdot \frac{n}{2^i} \cdot \bar{d}) \\ &= O(\eta(n) \cdot n \cdot \bar{d}) \sum_{i=0}^{\log_2(n)-1} 1 \\ &= O(\eta(n) \cdot n \cdot \bar{d} \log_2(n)), \end{aligned} \quad [29]$$

where we have made the pessimistic assumption that the number of iterations and the average degrees are in each step as large as they were in the beginning.

The choice of the function $\eta(n)$ is a little bit involved. If the initial random choice of the vector v is very unfortunate, there may be many iterations necesary in order to have a good approximation of the eigenvector v_2. In fact, if $\psi_2 = 0$, then this algorithm would not converge to v_2 at all, however this event has probability 0.

Due to this fast convergence, one can expect assymptotically good partitions when $\eta(n) = \log(n)^{1+\epsilon}$ and $\epsilon > 0$ and when the $1/log\left(|\frac{\tilde{\lambda_2}}{\tilde{\lambda_3}}|\right)$ is bounded by a constant, giving the hierarchical spectral clustering algorithm a complexity of $O(n \cdot \log^{2+\epsilon}(n))$ for sparse networks, where $\bar{d}$ is constant. Similarly, in (5) authors state that the eigenvector can be computed in $O((m+n)(\log n)^{O(1)})$ time using fast Laplacian solvers.

Another condition that might slow down the computation of v_2 is if some of the other eigenvalues λ_i, $i \geq 3$ are close to λ_2. In that case $\frac{\tilde{\lambda_i}}{\tilde{\lambda_2}}$ would be close to 1 and therefore one can see from equation (8) that the correspoding v_i might have a large contribution in $v^{(k)}$ for a long time. However when λ_i is close to λ_2, this also implies that

$$\frac{v_i^T L_w v_i}{v_i^T v_i} = \lambda_i \quad [30]$$

is close to

$$\min_{\|v\| \neq 0} \frac{v^T L_w v}{v^T v} = \lambda_2 \quad [31]$$

and therefore also provides a good partition of the network, since these are the quantities that are related to the cut-size.

5. Weighted vertex cover fine-tuning

For a network $G = (V, E)$ that will be partitioned into two groups M and $\bar{M}$, we construct a subgraph $G^* = (V^*, E^*)$, where edge set E^* comprises all the links between M and $\bar{M}$, node set V^* includes all the nodes involved in E^*. The dismantling cost (weight) of the nodes in V^* is given by a diagnol matrix W, defined by $w_i = \sum_j A_{i,j}$, where $A_{i,j}$ is the adjacency matrix of

the initial graph $G = (V, E)$. We want to find the subset $V^+ \subseteq V^*$ with minimal cost such that every edge $e \in E^*$ is incident to at least one node in V^+. That is, find

$$\min \sum_{i \in V^*} w_i y_i \quad [32]$$

subject to

$$y_i + y_j \geq 1, \forall (i,j) \in E^*. \quad [33]$$

$$y_i \in \{0, 1\}, \forall i \in V^* \quad [34]$$

We have used a linear time 2-approximation algorithm (6) to solve it. It means that the cost of the approximate solution $\sum_{i \in V^*} w_i y_i$ is at most 2 times higher than the optimal vertex cover cost C_{opt} of V^*, i.e., $\sum_{i \in V^*} w_i y_i \leq 2C_{opt}$.

By Kőnig's theorem, there is an equivalence between the minimum vertex cover problem in bipartite graphs and the maximum matching problem. But our experiments, have shown that maximum matching fine-tuning is not helping us to reach to better dismantling solution at the end.

The pseudocode of the weighted vertex cover approximation we adopted in this paper is shown in Algorithm 1.

Algorithm 1 Weighted vertex cover algorithm

procedure WVC(G^*,G)
  $S = \varnothing$ ▷ cover set
  $d_i = \sum_j A^*_{ij}$ ▷ residual degree of the nodes in G^*
  $w_i = \sum_j A_{ij}$ ▷ residual weight of the nodes in G
  while $E^* \neq \varnothing$ **do**
    Let v_i be the node with minimizing w_i/d_i
    $S \leftarrow S \bigcup \{v_i\}$
    $G^* \leftarrow G^* \setminus v_i$
    $G \leftarrow G \setminus v_i$
    update d, w
  return S ▷ return the list of nodes that should be removed

6. Pseudocode of GND algorithm

To compare the fragmentation effectiveness of different target attack algorithms, in general we plot the results like Fig. 5 of our paper: x-axis is the ratio of the nodes that has been removed, while y-axis is the ratio of the size of GCC. Then, the area under the curve (AUC) is usually is considered an effective method to reflect the effectiveness of the attack algorithms. Smaller value of AUC implies that the attack has a better dismantling effect over all possible budgets(7).

For a network $G = (V, E)$ with adjacency matrix A and node weights matrix W, the aim is to decompose the network such that the size of GCC is smaller than a target size. Then the GND algorithm is defined as follows.

Algorithm 2 GND algorithm

procedure GND(A,W)
  $A_{gcc}, W_{gcc} \leftarrow A, W$
  L_S ▷ the list of nodes that should be removed
  while $GCC_{size} > C$ **do** ▷ C is the target size
    $B = A_{gcc}W_{gcc} + W_{gcc}A_{gcc} - A_{gcc}$
    $(D_B)_{ii} = \sum_{j=1}^{n} B_{ij}$
    $L_w = D_B - B$
    $\tilde{L} = 6 \cdot d^2_{max} \cdot I - L_w$
    $v^{(2)} \leftarrow \tilde{L}$ ▷ spectral approximation
    Groups $M, \bar{M} \leftarrow v^{(2)}$ ▷ partition the GCC into two groups
    ▷ according to the eigenvector $v^{(2)}$
    $G^* = (V^*, E^*) \leftarrow M, \bar{M}$ ▷ construct subgraph G^*
    $S \leftarrow WVC(G^*, G)$ ▷ Weighted Vertex Cover
    $G \leftarrow G \setminus S$ ▷ removing nodes from network
    $L_S \leftarrow L_S.append(S)$
    update GCC_{size}, A_{gcc}, W_{gcc}
  end while
return L_S ▷ return the list of nodes that should be removed

The code of the GND algorithm will be available on github (https://github.com/renxiaolong/Generalized-Network-Dismantling) once this paper is published.

7. Generalized Network Dismantling Optimization Problem

A. Generalized Network Dismantling. In the *generalized network dismantling problem*, the cost of removing a node i can be any arbitrary non-negative number $w_i \in \mathcal{R}$. More formally, for a given network $G(V,E)$ with costs $W = (w_1, \ldots, w_{|V|})$, we aim to find the subset of nodes $S(G,W,C) \subseteq V$ with the minimum cost of removal (where the cost is given by the sum of the weights of the vertices in $S(G,W,C)$), which will result in a fragmentation of the network into components of size at most C (where the size of a component is simply the cardinality). We will denote the corresponding cost of such an optimal set by $Cost(G,W,C)$. It is easy to see that the case $W = I$, the identity matrix, is the standard *network dismantling problem* and its solution is related to the solution of the generalized problem by the following inequalities:

$$w_{min} Cost(G,I,C) \leq Cost(G,W,C) \leq w_{max} Cost(G,I,C),$$

where w_{min}, w_{max} denote the minimal and maximal weight over all nodes.

Generalized Network Dismantling problem GNDP$(G(V,E),W,C)$ is NP-hard. Every instance of 3SAT can be reduced (8, 9) to vertex cover; every instance of vertex cover can be reduced (10) to network dismantling $(G(V,E),C)$; and every instance of network dismantling $(G(V,E),C)$ can be reduced to generalized network dismantling problem, trivially by setting all weights to one, i.e., $(G(V,E),I,C)$.

Generalized Network Dismantling problem $(G(V,E),W,C)$ can not have fully polynomial-time approximation scheme (FPTAS).
If we assume that generalized network dismantling problem can be approximated up to arbitrary ϵ in polynomial time, it would imply that GNDP$(G(V,E),I,C=1)$ can also be approximated up to arbitrary ϵ in polynomial time. This would imply that the vertex cover can be approximated up to arbitrary ϵ as GNDP$(G(V,E),I,C=1)$ is equivalent to vertex cover. Then it would contradict the Dinur et. al. (2005) result "On the hardness of approximating minimum vertex cover" (11) and the unique games conjecture (12, 13).

In section 3 of the SI, we give the convergence bounds for spectral approximation. Note, that the spectral partitioning is related to relaxed node-weighted bisection problem. But, we are not giving approximation guarantee for the optimal solution of the (generalized) network dismantling problem.

B. Relation to other problems. Here, we will define the relation to another similar problems and highlight their difference.

Given a graph $G(V,E)$, finding a small separator S whose removal results in the partition to two roughly equal size sets A and B of disconnected vertices is called a graph separator problem. In case of removing vertices it is called balanced **vertex separator**, and in case of removing edges a balanced **edge separator**. Lipton et. al. showed that every n-vertex planar graph has a balanced vertex separator of size $O(\sqrt{n})$, which can be found in polynomial time (14). However, finding the minimum vertex or edge separator for general graphs is a NP-hard problem (15). Moreover, approximating vertex separator is at least as hard as approximating edge separators. Based on linear programming, an algorithm that approximates the minimum edge separator within a ratio of $O(\log n)$ was presented in (16). Next improvement (17) is based on semidefinite programming to approximate the minimum edge separator within a ratio of $O(\sqrt{\log n})$. In (18), the novel results for weighted vertex separator were introduced i.e. approximation of minimum vertex separator within a ratio of $O(\sqrt{\log n})$, again based on semidefinite programming. In the same work (18) the approximation bound $O(\sqrt{\log opt})$ w.r.t. to the size of optimal separator *opt*.
However, fast algorithms that solve semidefinite programming optimization problems for balanced separators run in $\tilde{O}(n^{4.5})$ and $\tilde{O}(n^{3.5}/\epsilon^2)$ run time complexity (19), where $\tilde{O}(.)$ is the big-O notation that "ignores" logarithmic factors.

An **α-separator** of graph $G = (V,E)$, separates the vertices to disjoint sets A, S and B, such that there is no edge between A and B, and $\max\{|A|,|B|\} \leq \alpha|V|$. This problem is generalization of the previous balanced vertex cover problem and fall to class of parameterized graph separation problems (20).

A set S is called a C-dismantling set if the largest connected component of a network after removing set S contains at most C nodes (21). Finding a minimum C-dismantling set was called Network Dismantling Problem (10). However, the same problem exists under the name K-separator problem (22), where authors provide useful polynomial case algorithms for special instances of graphs. Note, that in the network dismantling scenario, the number of partitions is not defined like in vertex or edge separator problems.

Alternative approach to the edge separator problems is by the integer relaxation that leads to **spectral partitioning** (1, 2). The detailed theoretical analysis of the quality of spectral separators can be found here (23). The problem of minimum-cut partitioning of a network into more than two parts by means of spectral embedding is derived from first principles (3).

The term **graph fragmentation** (24, 25) is a general term, that describes the process of removing nodes or edges and it's effect on the connected components, with no specific constraints in the optimization problem. Usually, the term was mostly used in the scenario of the ensemble of random graphs and random failures. **Network attack** (26–28) is a general term, which was used to describe the fragmentation process that is not completely random but involves some kind of optimization with the main purpose of harming the normal functioning of a system.

C. Benchmark network.

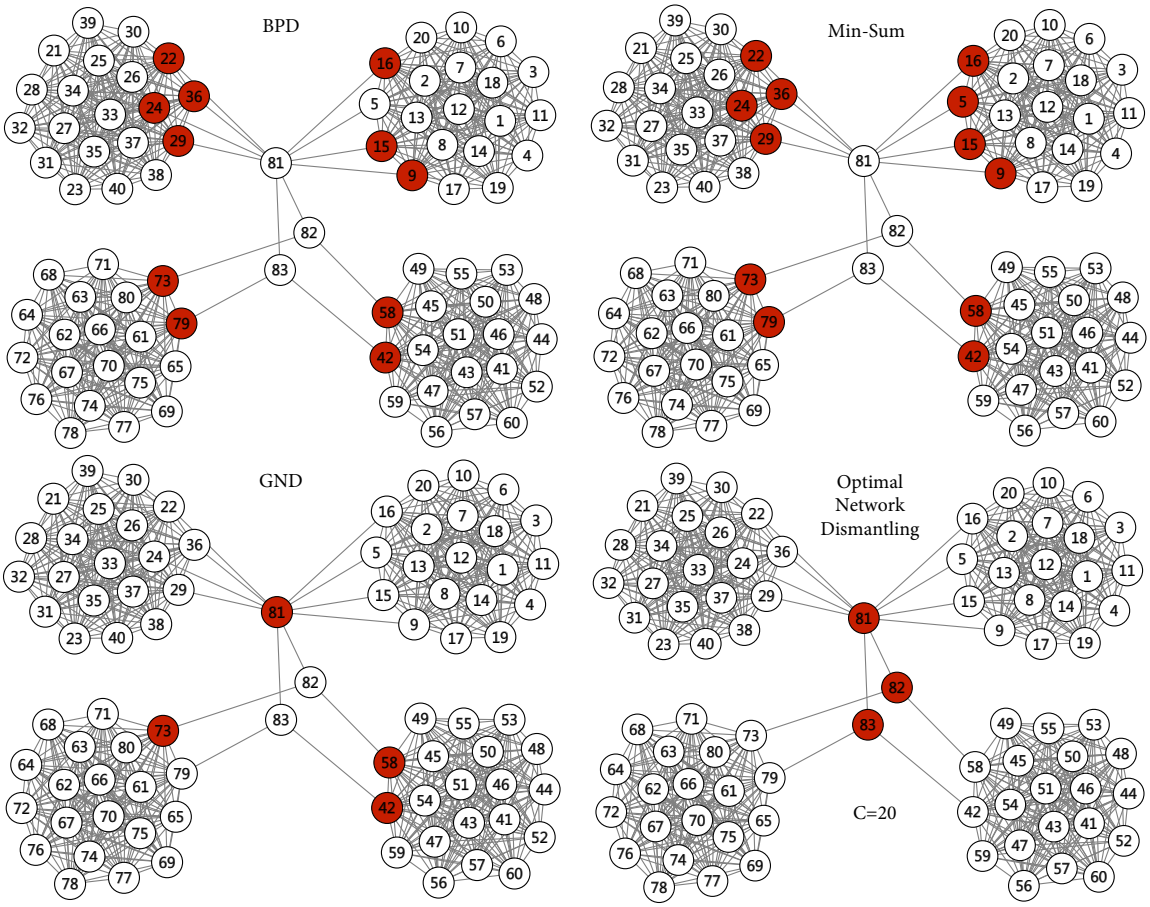

Fig. S1. Benchmark network G of 83 nodes and Network Dismantling Problem NDP(G, I, C) with target size $C = 20$ nodes, i.e., $c \approx 0.24$. The network consists out of 4 groups of complete subgraphs K_{20}: the group with nodes 1-20, the group with nodes 21-40, the group with nodes 41-60 and the group with nodes 61-80. Complete subgraphs are connected via three nodes: node 81 with degree 8, node 82 with degree 3 and node 83 with degree 3. The removed nodes are shown in red. The optimal solution is set of nodes $S^*_{NDP} = \{81, 82, 83\}$ and has an overall dismantling $Cost(G, I, C) = 3$, i.e. three nodes need to be removed such that the largest connected component has at most $C = 20$ size or ≈ 0.24 of the original GCC size. On this benchmark network, the best performing state-of-the-art algorithms BPD (29), Min-Sum (10) and the proposed GND algorithm have following cost, $Cost$(BPD)$= 11$, $Cost$(Min-Sum)$= 12$ and $Cost$(GND)$= 4$. We observe that the loops make problems for Min-Sum and BPD algorithm, but not for GND algorithm. The EGP algorithm (30) on this benchmark had the cost much larger than BPD and Min-Sum, so it was omitted.

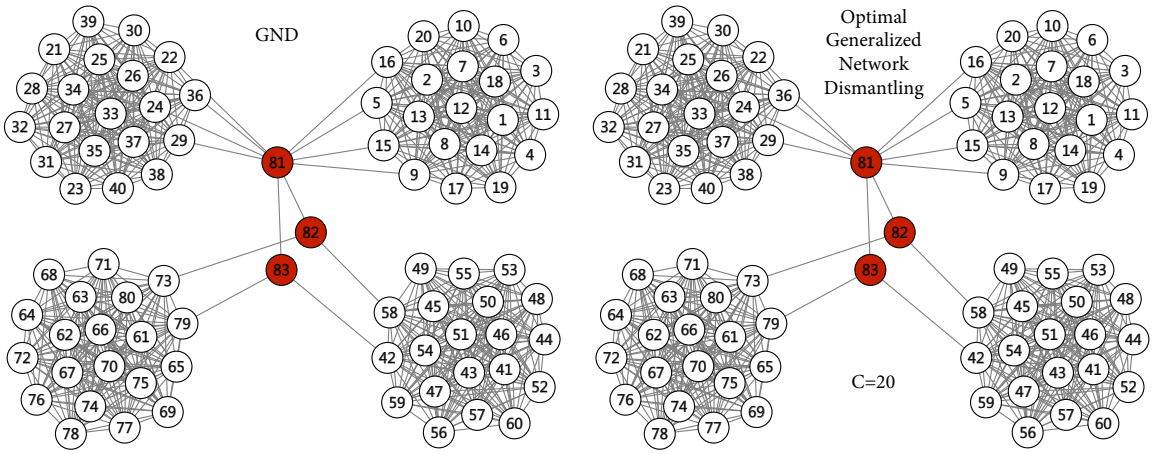

Fig. S2. Benchmark network G of 83 nodes and **Generalized Network Dismantling Problem** GNDP(G, W, C) with target size $C = 20$ nodes, i.e., $c \approx 0.24$. See network description in Fig. S1. Removing cost weights are equal to the degree of a node, e.g. $w_{81} = 8$, $w_{82} = 3$, $w_3 = 19$, $w_{66} = 19$, etc. The optimal solution for GNDP(G, W, C) with target size $C = 20$ nodes is set of nodes $S^*_{GNDP} = \{81, 82, 83\}$ with the overall cost $Cost(G, W, C) = 14$. The proposed GND algorithm in this case finds the optimal solution. Note that in the case of $C = 20$, the GNDP and NDP problems have the same optimal solution, i.e. $S^*_{GNDP} = S^*_{NDP}$, see Fig. S1. Furthermore, the inequality $w_{min}Cost(G, I, C) \leq Cost(G, W, C) \leq w_{max}Cost(G, I, C)$ connects the unit and non-unit case, where $w_{min} = 3$, $w_{max} = 20$, $Cost(G, I, C) = 3$ and $Cost(G, W, C) = 14$.

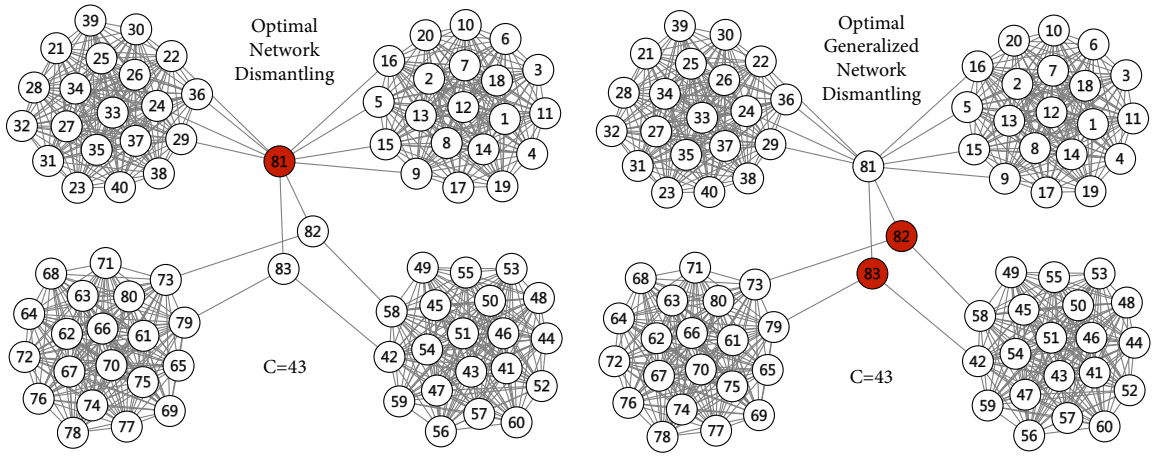

Fig. S3. Benchmark network G of GCC size 83 nodes and comparison of optimal **Generalized Network Dismantling Problem** and **Network Dismantling Problem** for target size $C = 43$, i.e. $c \approx 0.52$. See Fig. S2 and S1 for more details on network. The optimal solution for non-unit problem GNDP(G, W, C) with target size $C = 43$ nodes is set of nodes $S^*_{GNDP} = \{82, 83\}$ with the overall cost $Cost(G, W, C = 43) = 6$. The optimal solution for unit problem NDP(G, I, C) with target size $C = 43$ nodes is the set of nodes $S^*_{NDP} = \{81\}$ with the overall cost $Cost(G, I, C = 43) = 1$. Note that, in the case of $C = 43$ the GNDP and NDP problems have different optimal solutions, i.e. $S^*_{GNDP} \neq S^*_{NDP}$. Furthermore, the inequality $w_{min}Cost(G, I, C) \leq Cost(G, W, C) \leq w_{max}Cost(G, I, C)$ connects the unit and non-unit case, where $w_{min} = 3$, $w_{max} = 20$, $Cost(G, I, C = 43) = 1$ and $Cost(G, W, C = 43) = 6$.

8. More comparison of different network removal approaches on more networks

Fig. S4 provides a supplementary to Fig. 3 of the main paper with two more algorithms: Collective Influence (CI) (31) and CoreHD (32). Fig. S5 provides a supplementary to Fig. 4 of the main paper with two more algorithms (CI and CoreHD) and two more networks. Fig. S6 presents the performance of different algorithms on six more networks for the unit costs case. Table S1 summarizes the results for different algorithms for different target sizes c in unit cost and non-unit cost cases. The results show that the proposed GNDR algorithm has a better performance over all situations.

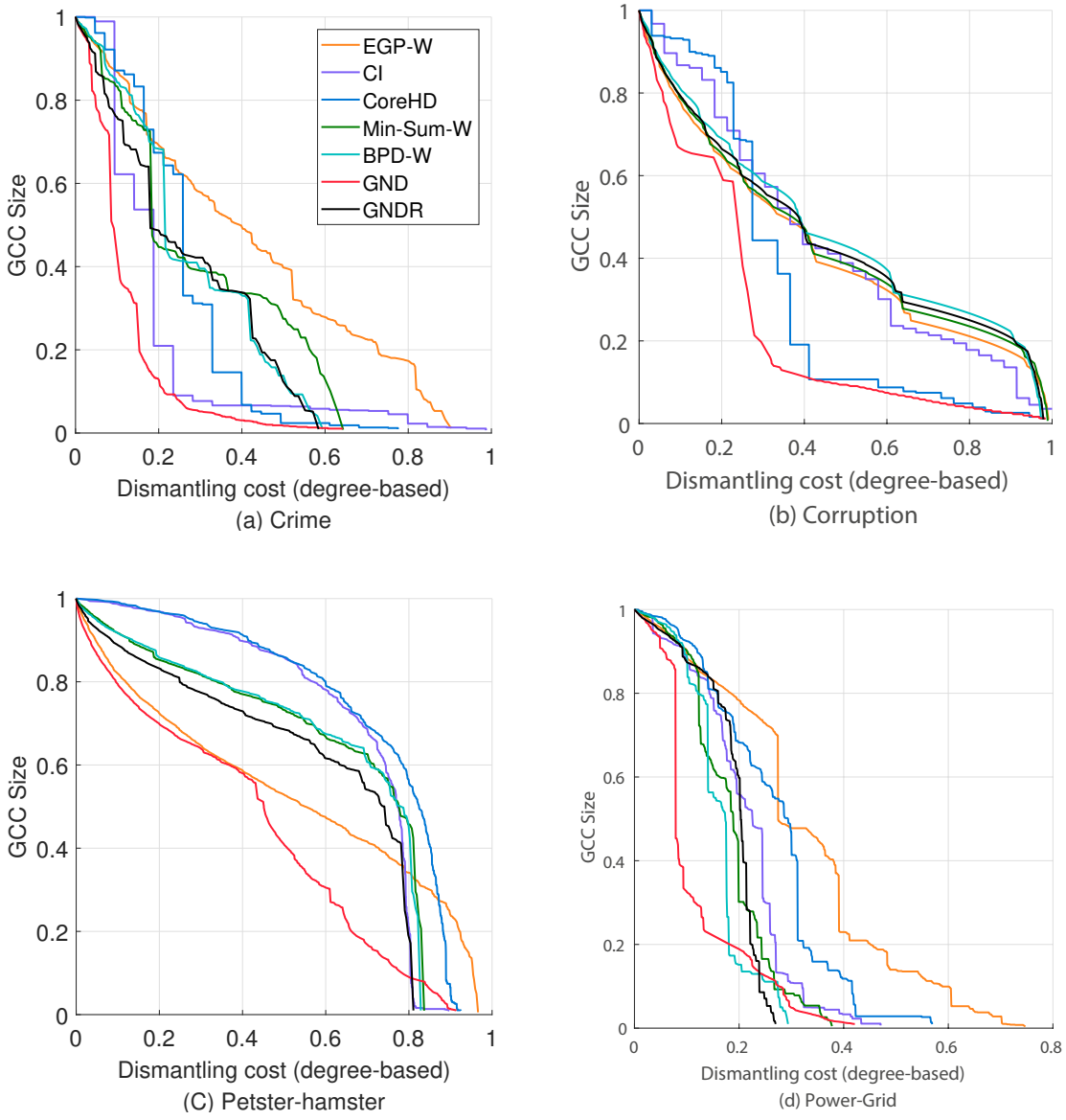

Fig. S4. The comparison of the GND and GNDR algorithms with more existing algorithms for degree-based costs case. In the Fig 3 of our main paper, only part of the existed algorithms were compared. Here we also present the comparison of GND and GNDR algorithm with CI and CoreHD algorithm. The dismantling curve shows the process of reducing the giant connected component (GCC) of a network with number of the removed nodes. The strategy with a lower area under the curve implies better dismantling. The performances of the algorithms when the cost of removing a node is equal for all nodes (unit costs) on (a) Crime network (33), (b) Corruption network (34), (c) Petster-hamster (35) online social network and (d) Power-Grid (36) network. We observe that for the case of degree-based costs with different target size, the proposed methodology (GND and GNDR) can always provide good solutions.

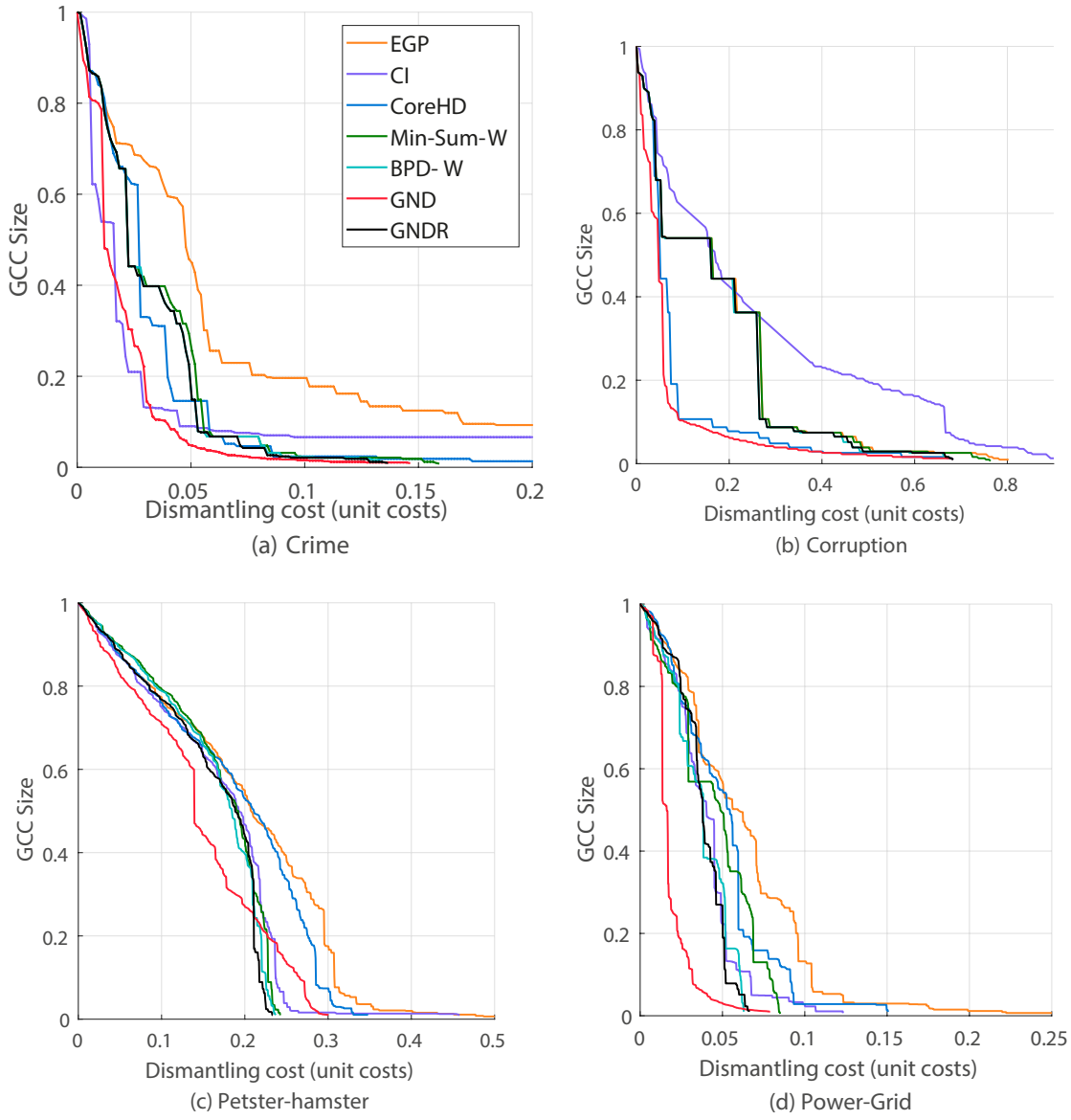

Fig. S5. The comparison of the GND and GNDR algorithms with more existing algorithms for unit cost case. In the Fig 4 of our main paper, only part of the existing algorithms were compared and only on two networks. Here we also present the comparison of GND and GNDR algorithm with CI and CoreHD algorithm on four networks. Size of the GCC versus unit-based dismantling cost for the four different networks: (a) Crime network (33), (b) Corruption network (34), (c) Petster-hamster (35) online social network and (d) Power-Grid (36) network. The dismantling represents creating firewalls for stopping the spread of disinformation, malicious cyber data, quarantines for epidemic and computer virus spreading, engineered breaking points for criminal and corruption networks, etc. The dismantling cost is measured with the fraction of the removed nodes, i.e., unit costs.

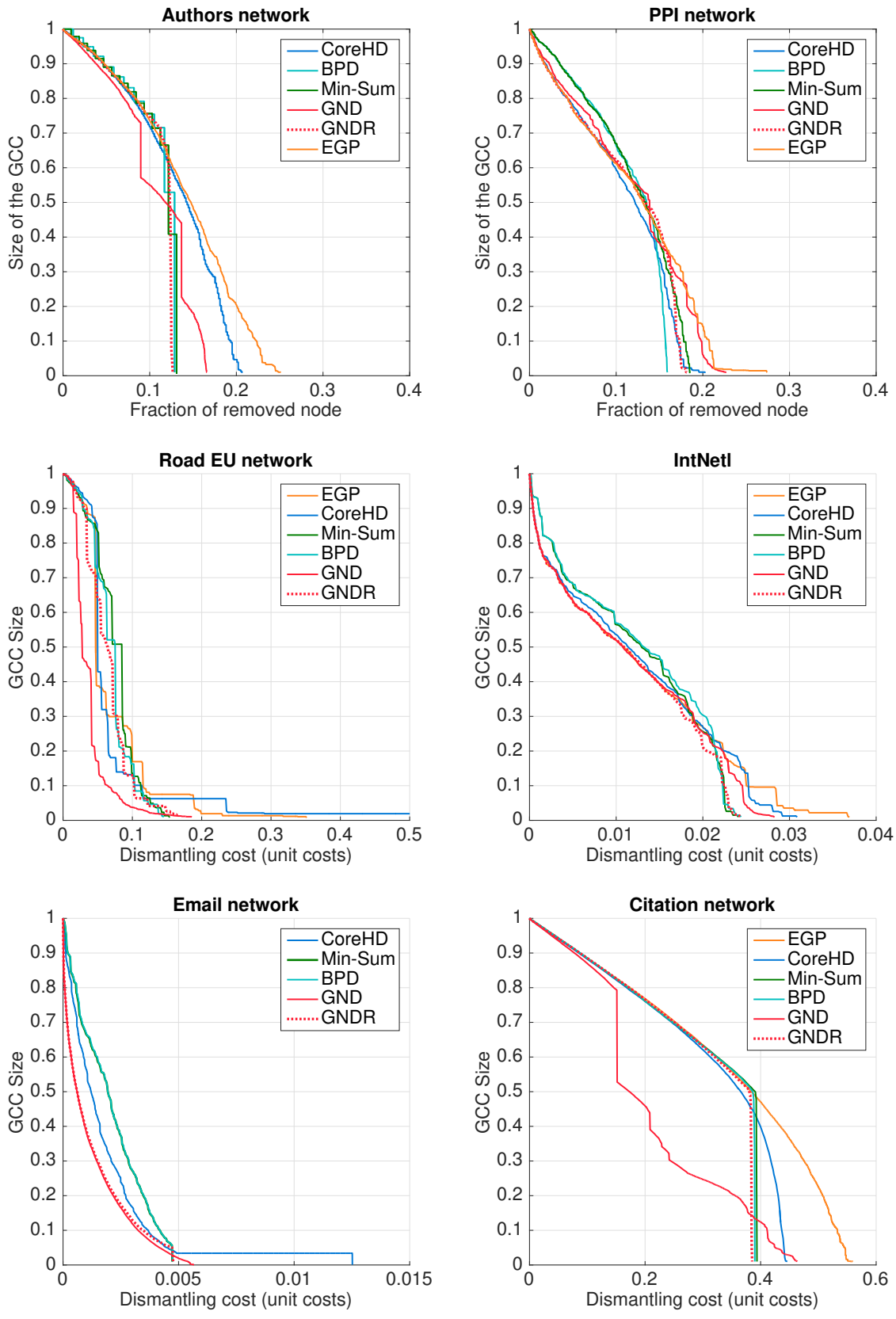

Fig. S6. Comparison of the performances of different algorithms for unit cost case on six more networks, including Authors network (N=21363, M=91286), PPI network (N=2224, M=6609), Road EU network (N=1039, M=1305), IntNet1 network (N=6474, M=12572), Email network (N=224832, M=339925), and Citation network (N=34401, M=420784) (32).

UNIT CASE

		PPI network	
c	ϕ_c(Min-Sum)	ϕ_c(BPD)	ϕ_c(GNDR)
0.8	**0.0341**	**0.0341**	0.0404
0.6	0.0876	0.1079	**0.0867**
0.4	0.1276	0.1200	**0.1092**
0.2	0.1357	**0.1326**	0.1492
		Grid network	
c	ϕ_c(Min-Sum)	ϕ_c(BPD)	ϕ_c(GNDR)
0.8	**0.0036**	0.0123	0.0070
0.6	**0.0036**	0.0123	0.0082
0.4	0.0103	0.0131	**0.0087**
0.2	0.0190	0.0184	**0.0097**
		Authors network	
c	ϕ_c(Min-Sum)	ϕ_c(BPD)	ϕ_c(GNDR)
0.8	0.0724	0.0711	**0.0578**
0.6	0.0954	0.0928	**0.0737**
0.4	0.1084	0.1063	**0.1057**
0.2	0.1140	**0.1129**	0.1247

NON-UNIT CASE (degree-based cost)

		PPI network	
c	ϕ_c(Min-Sum-W)	ϕ_c(BPD-W)	ϕ_c(GNDR)
0.8	0.3166	0.3304	**0.1202**
0.6	0.6159	0.6725	**0.2914**
0.4	0.7415	0.7079	**0.4944**
0.2	0.7606	0.7515	**0.6448**
		Grid network	
c	ϕ_c(Min-Sum-W)	ϕ_c(BPD-W)	ϕ_c(GNDR)
0.8	0.0175	0.0636	**0.0048**
0.6	0.0175	0.0636	**0.0048**
0.4	0.0468	0.0677	**0.0394**
0.2	0.0912	**0.0905**	0.1507
		Authors network	
c	ϕ_c(Min-Sum-W)	ϕ_c(BPD-W)	ϕ_c(GNDR)
0.8	0.4339	0.4356	**0.1144**
0.6	0.5383	0.5288	**0.3057**
0.4	0.5807	0.5724	**0.4609**
0.2	0.5973	0.5953	**0.5091**

Table S1. Comparison of results for different algorithms for different target sizes c in unit cost and non-unit cost cases. The corresponding overall dismantling cost is ϕ_c, i.e., the ratio of the removed cost. In each computational experiment, the best case is shown in bold. We observe that the proposed GNDR algorithm has a better performance over all situations.

9. Ethics

The method presented in this paper aims at offering a possible solution for emergencies where cutting a dysfunctional network into pieces can restore its functionality. However, we also warn of potential misuses or dual uses. When not applied in appropriate contexts and ways, the use of the dismantling approach may undermine the proper functionality of networks. Therefore, we point out that related ethical issues must be sufficiently, appropriately, and transparently addressed when the method is applied. The method must be restricted to legitimate uses and actors. It may be justified to stop harmful cascading problems such as deadly epidemics and the spreading of disruptive computer malware, or to dismantle criminal organizations or corruption networks. The method may also be used to identify more resilient system designs and network operations. Note, however, that the use of dismantling strategies to contain misinformation can be potentially problematic, as it may result in censorship if a government, company, news agency or other institution decides what is misinformation or not. Stopping the spread of true information can seriously obstruct the societal evolution towards better insights and solutions. Also note that, if public discourse is shaped by a few people only, this may promote the misuse of power, corruption and crime. In order to contain fake news, dis- and misinformation, we recommend a suitable combination of the use of AI, collective intelligence (such as Wikipedia and crowd-sourced fact checking), reputation systems for messages and information sources, elected community moderators, complaint mechanisms, qualification mechanisms, quality-based message ranking and reach, as well as verification/measurement-based approaches.

References

1. Fiedler M (1973) Algebraic connectivity of graphs. *Czechoslovak Math J* 23(2):298–305.
2. Pothen A, Simon HD, Liou KP (1990) Partitioning sparse matrices with eigenvectors of graphs. *SIAM J Matrix Anal Appl* 11(3):430–452.
3. Riolo MA, Newman MEJ (2014) First-principles multiway spectral partitioning of graphs. *J Complex Networks* 2(2):121–140.
4. Muller ME (1959) A note on a method for generating points uniformly on n-dimensional spheres. *Commun. ACM* 2(4):19–20.
5. Benson AR, Gleich DF, Leskovec J (2016) Higher-order organization of complex networks. *Science* 353(6295):163–166.
6. Bar-Yehuda R, Even S (1981) A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms* 2(2):198 – 203.
7. Ren XL, Gleinig N, Tolić D, Antulov-Fantulin N (2018) Underestimated cost of targeted attacks on complex networks. *Complexity* 2018.
8. Karp R (1972) Reducibility among combinatorial problems in *Complexity of Computer Computations*, eds. Miller R, Thatcher J. (Plenum Press), pp. 85–103.
9. Garey MR, Johnson DS (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness.* (W. H. Freeman & Co., New York, NY, USA).
10. Braunstein A, Dall'Asta L, Semerjian G, Zdeborová L (2016) Network dismantling. *Proc Natl Acad Sci USA* 113(44):12368–12373.
11. Dinur I, Safra S (2004) On the hardness of approximating minimum vertex cover. *Annals of Mathematics* 162:2005.
12. Khot S, Regev O (2008) Vertex cover might be hard to approximate to within 2-ϵ. *Journal of Computer and System Sciences* 74(3):335–349.
13. Dinur I, Khot S, Kindler G, Minzer D, Safra M (2018) Towards a proof of the 2-to-1 games conjecture? in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing - STOC 2018.* (ACM Press).
14. Lipton RJ, Tarjan RE (1979) A Separator Theorem for Planar Graphs. *SIAM Journal on Applied Mathematics* 36(2):177–189.
15. Bui TN, Jones C (1992) Finding good approximate vertex and edge partitions is NP-hard. *Information Processing Letters* 42(3):153–159.
16. Leighton T, Rao S (1999) Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM* 46(6):787–832.
17. Arora S, Rao S, Vazirani U (2004) Expander flows, geometric embeddings and graph partitioning in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing - STOC '04.* (ACM Press).
18. Feige U, Hajiaghayi M, Lee JR (2008) Improved approximation algorithms for minimum weight vertex separators. *SIAM Journal on Computing* 38(2):629–657.
19. Arora S, Hazan E, Kale S (2005) Fast algorithms for approximate semidefinite programming using the multiplicative weights update method in *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05).* (IEEE).
20. Marx D (2006) Parameterized graph separation problems. *Theoretical Computer Science* 351(3):394–406.
21. Janson S, Thomason A (2008) Dismantling sparse random graphs. *Combin Probab Comput* 17(2):259–264.
22. Ben-Ameur W, Mohamed-Sidi MA, Neto J (2013) The k-separator problem in *Computing and Combinatorics*, eds. Du DZ, Zhang G. (Springer Berlin Heidelberg, Berlin, Heidelberg), pp. 337–348.
23. Guattery S, Miller GL (1998) On the quality of spectral separators. *SIAM Journal on Matrix Analysis and Applications* 19(3):701–719.
24. Molloy M, Reed B (1995) A critical point for random graphs with a given degree sequence. *Random Structures Algorithms* 6(2/3):161–179.

25. Cohen R, Erez K, ben Avraham D, Havlin S (2000) Resilience of the internet to random breakdowns. *Phys Rev Lett* 85(21):4626–4628.
26. Tanizawa T, Paul G, Cohen R, Havlin S, Stanley HE (2005) Optimization of network robustness to waves of targeted and random attacks. *Phys Rev E* 71(4).
27. Schneider CM, Moreira AA, Andrade JS, Havlin S, Herrmann HJ (2011) Mitigation of malicious attacks on networks. *Proc Natl Acad Sci USA* 108(10):3838–3841.
28. Dong G, et al. (2013) Robustness of network of networks under targeted attack. *Phys Rev E* 87(5):052804.
29. Mugisha S, Zhou HJ (2016) Identifying optimal targets of network attack by belief propagation. *Phys Rev E* 94(1):012305.
30. Chen Y, Paul G, Havlin S, Liljeros F, Stanley HE (2008) Finding a better immunization strategy. *Phys Rev Lett* 101(5):58701.
31. Morone F, Makse HA (2015) Influence maximization in complex networks through optimal percolation. *Nature* 524(7563):65–68.
32. Zdeborová L, Zhang P, Zhou HJ (2016) Fast and simple decycling and dismantling of networks. *Scientific Reports* 6.
33. Kunegis J (2013) KONECT – The Koblenz Network Collection in *Proc. Int. Conf. on World Wide Web Companion*. pp. 1343–1350.
34. Ribeiro HV, Alves LGA, Martins AF, Lenzi EK, Perc M (2018) The dynamical structure of political corruption networks. *Journal of Complex Networks* pp. cny002–cny002.
35. Kunegis J (2013) The koblenz network collection in *Proceedings of the International Web Observatory Workshop*. pp. 1343–1350.
36. Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. *Nature* 393(6684):440–442.