Optimal shattering of complex networks

Nicole Balashov[1], Reuven Cohen[1], Avieli Haber[1], Michael Krivelevich[2] and Simi Haber[1*]

*Correspondence:
simi.haber@biu.ac.il
[1]Department of Mathematics,
Bar-Ilan University, 5290002
Ramat-Gan, Israel
Full list of author information is
available at the end of the article

Abstract

We consider optimal attacks or immunization schemes on different models of random graphs. We derive bounds for the minimum number of nodes needed to be removed from a network such that all remaining components are fragments of negligible size.

We obtain bounds for different regimes of random regular graphs, Erdős-Rényi random graphs, and scale free networks, some of which are tight. We show that the performance of attacks by degree is bounded away from optimality.

Finally we present a polynomial time attack algorithm and prove its optimal performance in certain cases.

Keywords: random graphs; shattering

1 Introduction

One of the most studied questions in complex networks is the resilience of networks under different failure models and attack strategies [1, 2, 3]. In particular, one wishes to know the optimal attack strategy that will lead to fragmentation by removal of a minimal fraction of the nodes. This information is important for estimating the vulnerability of network infrastructures, and also for devising optimal immunization strategies for populations and computer networks.

The main methods that have been proposed for targeted attacks on networks via node removal have been based on attack by highest degree [1, 3, 4], and attack by highest betweenness centrality [5]. Some methods based on more advanced algorithms for graph partitioning have also been proposed [6], and led to improved upper bounds on the minimal fraction of nodes that should be removed to shatter a networks.

In recent years, several works studied optimal attacks on networks, presenting sophisticated, highly efficient algorithms for choosing minimal sets of nodes whose removal leads to complete fragmentation of the network. In [7] an efficient dismantling method is presented, using the replica method on the generating function. In [8] efficient dismantling is considered when costs are attributed to the different nodes. In [9] finding sets of influential nodes is considered using the cavity and Extremal Optimization (EO) methods. In [10] belief propagation is used to find efficient shattering sets. In [11] optimal attacks on multiplex networks are studied using simulated annealing methods.

As for measuring the resilience of a network to shocks, there are again quite a few papers. The following is far from being a comprehensive survey. In [12] a shock is injected on one node and the resilience of the (weighted) network is measured by its ability to absorb this shock. In [13] network robustness is evaluated in terms of the ability to identify the attack prior to network disruption. The affect of nodes removal on the diameter of the network, a natural parameter, was studied in [14]. Both of these studies regard *attack by degree*

as representative for an intentional attack. An important breakthrough was made in [15], where the *dynamics* of the system was accounted for, in addition to network topology. One may ask about the resilience of the community structure in a network, and indeed this approach was taken in [16], and tested against link removal. For a review of definitions and measures of system resilience we refer the reader to [17].

In this work, we present results for several network classes, giving upper and lower bounds on the size of the minimal set of nodes to be removed in order to shatter (or dismantle) a network. We first survey the method of generating functions and results on random attacks and attack by degree, in order to show that random attack and attack by degree are not asymptotically optimal for any class of generalized (configuration model) random graphs. We then present exact results and bounds on the size of the shattering set for various random graph classes. Eventually, we present a polynomial time algorithm for efficient shattering of random graphs. We show, using exact methods, that for 3-regular graphs our algorithm obtains asymptotically optimal results. The performance of this algorithm for other classes of random graphs remains an open question.

2 Random Attacks

The problem of removing nodes in order to shatter the network into small pieces is of importance both in order to determine the resilience of a network to various attacks and in order to immunize a network (for example, a social network) against the spreading of an epidemic disease. By "shattering" a network, we mean breaking the network into small components, each of which having size $o(n)$, where n is the number of nodes. That is, breaking the network into pieces whose sizes are negligible compared to the original network, by means of nodes removal.

The simplest attack mechanism on a network is removing nodes uniformly at random. This is the standard percolation model. In order to study the effect of this attack on the network, one can employ the generating function method (See, e.g., [3]). We consider the configuration model, where no correlations exist between neighbouring nodes. Given a degree distribution $P(k)$, and probability $p(k)$ for a degree k node to exist (i.e., not to be deleted), one can write the generating function for this distribution as

$$F_0(x) = \sum_{k=0}^{\infty} P(k)p(k)x^k \,. \tag{1}$$

Similarly, one can write the generating function for the reciprocal degree distribution of a node reached by following an edge (i.e., the degree of the node disregarding the edge through which it was reached):

$$F_1(x) = \frac{F_0'(x)}{\langle k \rangle} = \frac{\sum_{k=0}^{\infty} kP(k)p(k)x^{k-1}}{\langle k \rangle} \,, \tag{2}$$

where $\langle k \rangle$ is the average degree. The generating function for branch sizes reached by following a random edge is given by the recursive equation.

$$H_1(x) = 1 - F_1(1) + xF_1(H_1(x)) \,, \tag{3}$$

where $F_1(1)$ is the probability of a node reached by following a random edge to exist. and the generating function for the probability of a node to belong to a component of some

finite size is given by

$$H_0(x) = 1 - F_0(1) + xF_0(H_1(x)) . \tag{4}$$

$H_0(1)$ is the normalization of $H_0(x)$. It may happen that $H_0(1) = 1$, in which case all components are finite and no giant component exists, or that $H_0(x) < 1$ in which case, a giant component exists and contains a fraction $1 - H_0(1) = 1 - F_0(1) + xF_0(u)$ of the nodes, where u is the solution of the self consistent equation

$$u = 1 - F_1(1) + F_1(u) . \tag{5}$$

One can observe that such a solution exists only if $F_1'(1) > 1$.

For random attacks $p(k) = p := 1 - q$ is independent of k and using Eq. (5) one can deduce that the criterion for the existence of a giant component is that there exist a solution $u < 1$, which exists only if

$$p > p_c := \frac{1}{\kappa - 1} , \tag{6}$$

where $\kappa := \frac{\langle k^2 \rangle}{\langle k \rangle}$ is the ratio of the first two moments of the distribution [2]. Specifically, using the moments of the constant and Poisson distributions, respectively, one can deduce that the percolation thresholds for the random d-regular and Erdős-Rényi networks are $\frac{1}{d-1}$ and $\frac{1}{\langle k \rangle}$.

Another result that can be deduced using this formalism is that the probability of a (non-deleted) node of degree k to belong to a finite component is the probability that all of the branches emanating from it are finite, i.e., the probability is u^k.

3 Targeted Attack Strategies

Naturally, random attacks are not expected to give optimal results. Indeed, a simple and more effective attack strategy is starting the removal with the high degree nodes, as they play a more substantial role in the network connectivity. Using a function

$$q(k) = \begin{cases} 1 & k < k_0 \\ 0 & k > k_0 \\ \alpha & k = k_0 \end{cases} , \tag{7}$$

for some k_0 and α. Solving for the values of k_0 and α that give criticality, one can find the critical fraction for removal.

The attack by degree strategy is, however, suboptimal. This can be seen from the fact that for any finite k, the probability that a removed node of degree k does not even belong to the giant component, and therefore its removal is unbeneficial, is u^k. Therefore, a finite fraction of the removed nodes are not included in the giant component to begin with, and thus the method is not even asymptotically optimal. Furthermore, for random regular networks, targeted attack by degree is completely equivalent to random removal.

In order to develop a better attack strategy one may consider methods such as adaptive attack by degree or attack by betweenness centrality. However, these strategies are very hard to analyse.

4 Bounds on optimal attacks

We define c_f as the minimal fraction of nodes that are to be removed before the network is shattered (i.e., becomes fragmented into sublinear components). It is clear that for any network $c_f \leq q_c = 1 - p_c$, where p_c is the percolation threshold for random removal of nodes. For a d-regular graph (where each node has degree d) this yields the upper bound

$$c_f \leq 1 - \frac{1}{d-1} = \frac{d-2}{d-1}.$$

However, one can ask the question: Given complete knowledge of the network, and unlimited computational power, what is the smallest fraction of the nodes that can be removed in order to break the network into small pieces, each of size $o(n)$? One might be tempted to consider the possibility of shattering the network by removing only a zero fraction, $o(1)$ of the nodes. Indeed, in the case of a square grid, for example, which is a regular network with fixed degree 4, one can remove $n^{1/4} = o(n)$ equally spaced rows and columns and shatter the network into pieces of size $O(\sqrt{n})$.

For some cases, such as random regular networks, this can be shown to be impossible. Indeed, improved bounds were obtained in [18, 19]:

$$\frac{d-2}{2d-2} \leq c_f \leq \frac{d-2}{d+1}.$$

In this paper we establish better lower and upper bounds on c_f, focusing on sparse random graphs. In particular for random regular graphs we show

$$1 - 2\frac{\alpha(G)}{n} \leq c_f \leq 1 - \frac{\alpha(G)}{n} \,, \tag{8}$$

where $\alpha(G)$ is the independence number of G. When d is large enough the independence number is known [20] to satisfy $\alpha(G) \approx 2n\ln d/d$. In this case we obtain

$$c_f \approx 1 - \frac{\alpha(G)}{n}.$$

We provide matching results for Erdős-Rényi random graphs.

5 Results

For the analytical results, we mainly exploit structural graph properties such as expansion, domination number and independence and obtain deterministic connections to the shattering number. We then apply known estimations of these parameters for random graphs either directly or via contiguity arguments.

As an example we establish a lower bound on the minimal fraction of nodes needed to be removed in order to shatter a regular network into small components of size $o(n)$. Consider shattering a graph to m disjoint clusters C_i, $i = 1, \ldots, m$ by deleting a set of $|S| = c_f n$ nodes. c_f is therefore the fraction of removed nodes. Notice that $\sum_{i=1}^{m} |C_i| + |S| = n$. Thus, $\frac{1}{n}\sum_i |C_i| + c_f = 1$. Denote by B_i the nodes on the boundary of C_i, i.e., the neighbors of nodes of cluster i outside of the cluster. Since the clusters are disconnected for all i, j, all boundary nodes for any cluster i must be removed. Therefore, for every i, all nodes in B_i are deleted. Now, in a random regular graph with high probability each cluster is locally almost

tree like. Therefore, up to an additive constant, $|B_i| = (d-1)|C_i| - |C_i| = (d-2)|C_i|$. Since every node has exactly d neighbors, a node can not participate in more than d of the B_is. Thus $\frac{1}{d} \sum_i |B_i| \leq |S|$. Therefore,

$$n \geq \sum_i C_i + \frac{d-2}{d} \sum_i C_i = \frac{2d-2}{d} \sum_i C_i. \tag{9}$$

Summarizing the above we get:

$$c_f = 1 - \frac{1}{n} \sum_i C_i \geq \frac{d-2}{2d-2}. \tag{10}$$

This approach can be shown to give asymptotically tight solution for $d = 3$ (i.e., it matches the upper bound shown in [18]). However, for large values of d it deviates considerably from the exact value. Indeed, for $d \to \infty$, Eq. (10) leads to $c_f \geq \frac{1}{2}$, where as will be shown below $c_f \to 1$.

In order to give a better lower bound on c_f for random regular graphs with large constant degree we observe the following: Random regular graphs are locally tree like, having a bounded number of short cycles. Therefore after the network is shattered we expect the remaining components to be trees. A tree can be shattered into isolated vertices by removing at most half of its nodes. Therefore, it can be deduced that the number of nodes remaining after the attack is at most twice the size of the largest independent set. Since removing all but an independent set clearly shatters the graph, we obtain Equation (8). With some further effort we can improve the lower bound to close the gap and match the upper bound. The same approach may be applied to the Erdős-Rényi model $G(N, p = c/N)$ where we get

$$c_f \approx 1 - \frac{\log c}{c}. \tag{11}$$

We discuss theoretical and practical implications of our results in Section 8.

6 Shattering Scale Free Networks

Targeted attack strategies usually begin by attacking the hubs of the scale free networks. This is based on the idea that the hubs are the glue holding the network together, due to their high degree and large number of neighbours. Indeed, one may use this intuition to obtains bounds on the hardness of shattering a scale free networks.

Consider a scale free network with degree distribution

$$P(k) = \frac{k^{-\gamma}}{\zeta(\gamma)}, \tag{12}$$

where $\zeta(\gamma) = \sum_{k=1}^{\infty} k^{-\gamma}$. This network can be shattered using the following two stage process:

1 Remove all nodes above degree d. This contains a fraction of $\sum_{k=d+1}^{\infty} k^{-\gamma}/\zeta(\gamma)$ of the nodes.

2 Shatter the remaining network, requiring at most $\frac{d-2}{d+1}$ of the network.

One can choose d as to minimize the sum of these two terms in order to obtain an upper bound on the size of the required shattering set. In fact, a better bound can be obtained by

noticing that after the removal of the hubs, the remaining nodes form a random network with degree distribution

$$\tilde{P}(k) = \sum_{c=k}^{\infty} \frac{k^{-\gamma}}{\zeta(\gamma)} \binom{c}{k} p^k (1-p)^{c-k} \,, \tag{13}$$

with

$$p = \sum_{1}^{d} \frac{kP(k)}{\langle k \rangle} \tag{14}$$

denoting the probability of an edge to lead to an undeleted node (i.e. the fraction of edges leading to undeleted nodes). If one can bound the size of the shattering set for a random graph with this degree distribution, a better bound on the overall shattering set size can be obtained. In particular, by showing that the remaining graph is close enough to random graph in the sense of having all local neighborhoods as almost trees with high probability, one may replace the upper bound $\frac{d-2}{d+1}$ in Point 2 above by $1 - \alpha(G)/n$.

7 Algorithmic aspects

Finding an optimal shattering set is NP-hard, but when the input is a random graph the problem becomes tractable. We propose the following algorithm for finding a shattering set in a graph. Below we describe the algorithm and demonstrate its asymptotic optimality for random cubic graphs. Let $G \sim G_{n.3}$ be a random cubic graph. Consider the following algorithm:

Algorithm Shatter, Phase I

1   Input: a graph G and threshold t
2   Find a Hamilton cycle H in G
3   Start from an arbitrary node v_0 and advance along H creating a segment.
4   When visiting a node v, it is incident with two edges on the cycle and a third edge e.
    If e is the second edge in the segment going backward (in H), delete v.

Each edge is seen once going forward and once going backward, and we delete nodes for half of the edges seen going backward, we remove exactly $1/4$ of the nodes. This is optimal as can be seen in Eq. (10). A demonstration of Phase I is given in Figure 1.
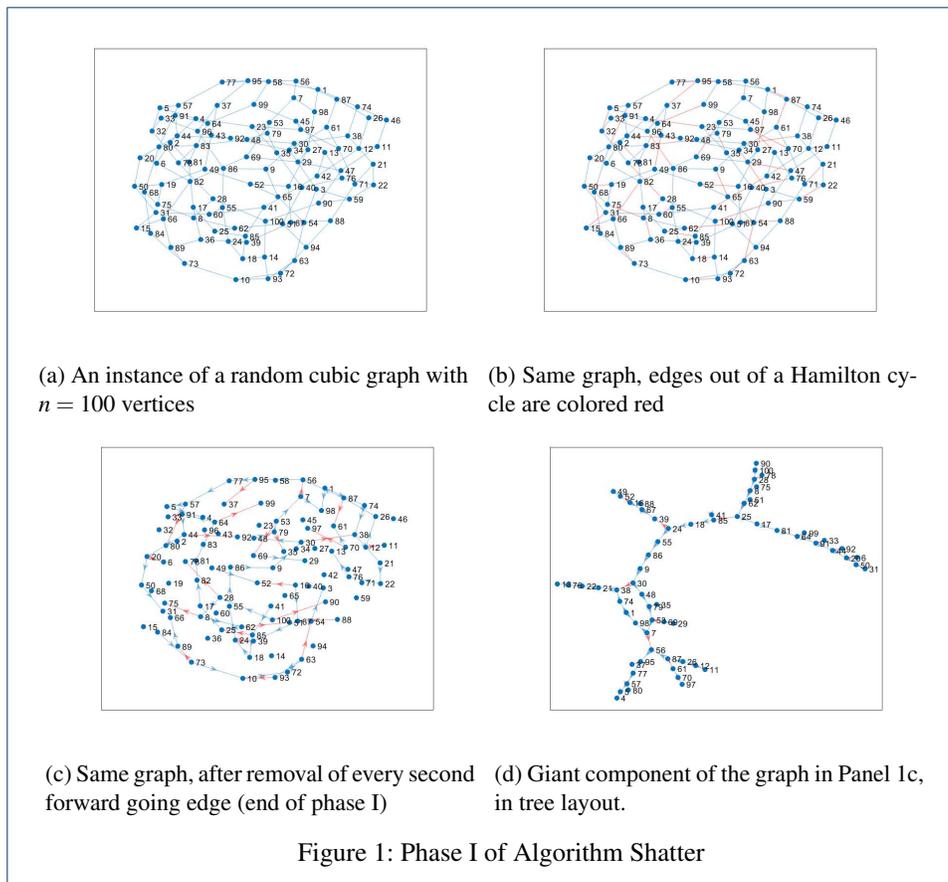
When Phase I is complete we are left with a tree T of segments of H. These segments are with high probability of length $o(n)$ and are unicyclic. Shattering T can be easily done by removing a center vertex of the tree, leaving at least two subtrees with at most half of the original number of segments in each. We continue in this manner until the graph is shattered, that is until every connected component is of size smaller than a predefined threshold. Summarizing we get:

Algorithm Shatter, Phase II

1   Let T be the tree of segments remaining after Phase I
2   Until the maximal size of a connected component is smaller than t
    (a) Find a center v in the largest connected component
    (b) Remove v

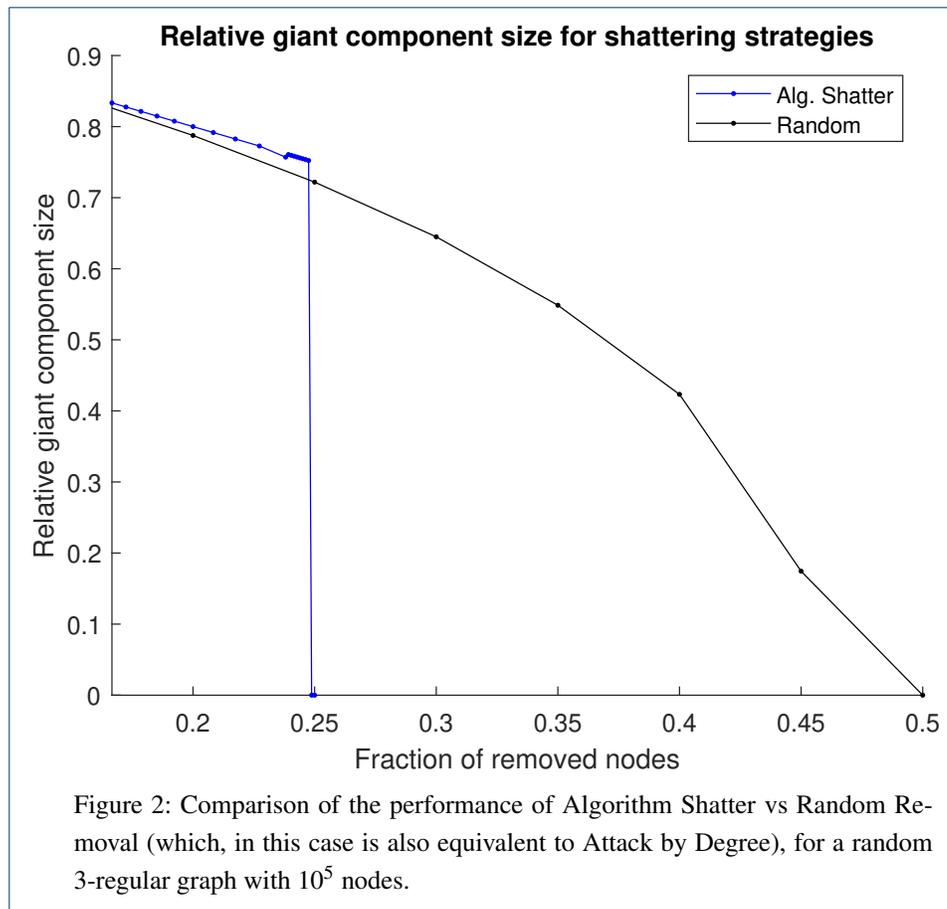The vertices removed in Phase I and Phase II together form the shattering set.

The two phases are demonstrated in the following figures:

(a) An instance of a random cubic graph with $n = 100$ vertices

(b) Same graph, edges out of a Hamilton cycle are colored red

(c) Same graph, after removal of every second forward going edge (end of phase I)

(d) Giant component of the graph in Panel 1c, in tree layout.

Figure 1: Phase I of Algorithm Shatter

Random cubic graphs are known to be Hamiltonian [21] and using a variant of ideas from the proofs from [22] we could get an algorithm finding such a cycle with high probability in time $O(n^{7/2})$. The rest of the algorithm (in both phases) runs in linear time. Preliminary results using a Branch and Bound approach suggest we may be able to reduce the running time of the Hamilton cycle finding module.

Notice that unlike removal by degree or by betweeness, which are local algorithms, the first phase of *Shatter* finds a global structure in the network. This is our main insight — we take advantage of the network randomness in order to find a global structure, then we use this structure to achieve high performance. In particular we get an asymptotically optimal constant for random cubic graphs. While, as stated below, we can not guarantee optimality in all cases, we still believe this approach is favorable to local strategies, either passive or adaptive. This belief stems from the global nature of the problem, for which it is natural to suggest a global solution, and our attempt to optimize over a set rather than greedily point by point. Indeed, even when considering shattering to components of size one (i.e. finding an independent set), greedy algorithms are known to deliver poor results (in terms of approximation factor) when applied to random graphs [23].

For d-regular graphs with $d > 3$ the same algorithm can be applied: tour along the Hamiltonian path, and remove the node at which a second edge going backward is observed. However, finding a Hamiltonian cycle in a d regular graph with $d > 3$ induces non-trivial correlations between the edges, and thus the performance of the algorithm is hard to eval-

Figure 2: Comparison of the performance of Algorithm Shatter vs Random Removal (which, in this case is also equivalent to Attack by Degree), for a random 3-regular graph with 10^5 nodes.

uate. It is expected, however, that this algorithm will only retain an $O(1/k)$ fraction of the nodes, which is suboptimal.

For Erdős-Rényi graphs $G(n,p)$ one may use a similar algorithm, with the following performance improvements:

1   Every node of degree two can be replaced with an edge connecting its two neighbours. This is true, since long chains of length $O(n)$ occur with negligible probability in $G(n,p)$.

2   Once the above step is performed, one can consider only the giant 3-core of the network, and perform *Algorithm Shatter* on it.

8 Summary

We provide bounds on the performance of optimal attack strategies on random networks, show that local strategies fail to achieve optimality even if used in adaptive manner and demonstrate an algorithm using global structure with optimal performance in certain situations. Our work draws the limits of feasibility for this well studied problem and shows that in some cases these limits are practically achievable. Besides their immediate value, our results may have broader implications. First, Algorithm Shatter is applicable whenever a long path or cycle may be found efficiently, e.g. when the network is based on a topological structure. Moreover, we see these results as an evidence for the "global solution to global problem" approach, and hope it will help in promoting this idea.

Author details
[1]Department of Mathematics, Bar-Ilan University, 5290002 Ramat-Gan, Israel. [2]Department of Mathematics, Tel-Aviv University, 6997801 Tel-Aviv, Israel.

References
1. Albert, R., Jeong, H., Barabási, A.-L.: Error and attack tolerance of complex networks. Nature **406**, 378–382 (2000). doi:10.1016/0031-8914(72)90045-6
2. Cohen, R., Erez, K., ben-Avraham, D., Havlin, S.: Resilience of the internet to random breakdown. Phys. Rev. Lett. **85**(21), 4626–4628 (2000). doi:10.1103/PhysRevLett.85.4626
3. Callaway, D.S., Newman, M.E.J., Strogatz, S.H., Watts, D.J.: Network robustness and fragility: Percolation on random graphs. Phys. Rev. Lett. **85**, 5468–5471 (2000). doi:10.1103/PhysRevLett.85.5468
4. Cohen, R., Erez, K., ben-Avraham, D., Havlin, S.: Breakdown of the internet under intentional attack. Phys. Rev. Lett. **86**(16), 3682–3685 (2001). doi:10.1103/PhysRevLett.86.3682
5. Magoni, D.: Tearing down the internets. IEEE J. Sel. Areas Commun. **21**(6), 949–960 (2003). doi:10.1109/JSAC.2003.814364
6. Paul, G., Cohen, R., Sreenivasan, S., Havlin, S., Stanley, H.E.: Graph partitioning induced phase transitions. Phys. Rev. Lett. **99**(11), 115701 (2007). doi:10.1103/PhysRevLett.99.115701
7. Braunstein, A., Dall'Asta, L., Semerjian, G., Zdeborová, L.: Network dismantling. Proceedings of the National Academy of Sciences **113**(44), 12368–12373 (2016). doi:10.1073/pnas.1605083113. https://www.pnas.org/content/113/44/12368.full.pdf
8. Ren, X.-L., Gleinig, N., Helbing, D., Antulov-Fantulin, N.: Generalized network dismantling. Proceedings of the National Academy of Sciences **116**(14), 6554–6559 (2019). doi:10.1073/pnas.1806108116. https://www.pnas.org/content/116/14/6554.full.pdf
9. Morone, F., A. Makse, H.: Influence maximization in complex networks through optimal percolation. Nature **524** (2015). doi:10.1038/nature14604
10. Mugisha, S., Zhou, H.-J.: Identifying optimal targets of network attack by belief propagation. Phys. Rev. E **94**, 012305 (2016). doi:10.1103/PhysRevE.94.012305
11. Osat, S., Faqeeh, A., Radicchi, F.: Optimal percolation on multiplex networks. Nature Communications **8** (2017). doi:10.1038/s41467-017-01442-2
12. Cerqueti, R., Ferraro, G., Iovanella, A.: Measuring network resilience through connection patterns. Reliability Engineering & System Safety **188**, 320–329 (2019). doi:10.1016/j.ress.2019.03.030
13. Chen, P.-Y., Cheng, S.-M.: Sequential defense against random and intentional attacks in complex networks. Phys. Rev. E **91**, 022805 (2015). doi:10.1103/PhysRevE.91.022805
14. Ferraro, G., Iovanella, A.: Clairvoyant targeted attack on complex networks. International Journal of Computational Economics and Econometrics **8**(1), 41–62 (2018). doi:10.1504/IJCEE.2018.088314. https://www.inderscienceonline.com/doi/pdf/10.1504/IJCEE.2018.088314
15. Gao, J., Barzel, B., Barabási, A.-L.: Universal resilience patterns in complex networks. Nature **530**, 307–312 (2016)
16. Ramirez-Marquez, J.E., Rocco, C.M., Barker, K., Moronta, J.: Quantifying the resilience of community structures in networks. Reliability Engineering & System Safety **169**, 466–474 (2018). doi:10.1016/j.ress.2017.09.019
17. Hosseini, S., Barker, K., Ramirez-Marquez, J.E.: A review of definitions and measures of system resilience. Reliability Engineering & System Safety **145**, 47–61 (2016). doi:10.1016/j.ress.2015.08.006
18. Edwards, K., Farr, G.: Fragmentability of graphs. Journal of Combinatorial Theory, Series B **82**(1), 30–37 (2001). doi:10.1006/jctb.2000.2018
19. Edwards, K., Farr, G.: Planarization and fragmentability of some classes of graphs. Discrete Mathematics **308**(12), 2396–2406 (2008). doi:10.1016/j.disc.2007.05.007
20. Frieze, A.M., Łuczak, T.: On the independence and chromatic numbers of random regular graphs. Journal of Combinatorial Theory, Series B **54**(1), 123–132 (1992). doi:10.1016/0095-8956(92)90070-E
21. Robinson, R.W., Wormald, N.C.: Almost all cubic graphs are Hamiltoniann. Random Structures & Algorithms **3**(2), 117–125 (1992). doi:10.1002/rsa.3240030202
22. Frieze, A.M., Jerrum, M., Molloy, M., Robinson, R.W., Wormald, N.C.: Generating and counting hamilton cycles in random regular graphs. Journal of Algorithms **21**(1), 176–198 (1996). doi:10.1006/jagm.1996.0042

23. Grimmett, G.R., McDiarmid, C.J.H.: On colouring random graphs. Mathematical Proceedings of the Cambridge Philosophical Society **77**(2), 313–324 (1975). doi:10.1017/S0305004100051124