From Random Failures to Targeted Attacks in Network Dismantling

Sebastian Wandelt[a,b], Wei Lin[c], Xiaoqian Sun[a,b,*], Massimiliano Zanin[d]

[a]*National Key Laboratory of CNS/ATM, School of Electronic and Information Engineering, Beihang University, 100191 Beijing, China*
[b]*Beihang Hangzhou Innovation Institute Yuhang, Xixi Octagon City, Yuhang District, 310023 Hangzhou, China*
[c]*School of General Engineering, Beihang University, 100191 Beijing, China*
[d]*Instituto de Física Interdisciplinar y Sistemas Complejos IFISC (CSIC-UIB), Campus UIB, 07122 Palma de Mallorca, Spain*

Abstract

It is well-known that real-world systems, modeled as complex networks, are mostly robust against random failures but susceptible to targeted attacks. In this study, we propose a novel perspective to solve the network dismantling problem. Instead of designing an effective attack from scratch, we show how knowledge extracted from random failures in the network leads to extremely effective attacks. This observed connection between random failures and targeted attacks is striking on its own. Experiments on a wide range of networks show the efficacy of our novel method for network dismantling, providing an excellent trade-off between attack quality and scalability. We believe that our contribution also stimulates research in related domains, including social network influence analysis, spreading dynamics in networks, and efficiency considerations.

Keywords: Complex networks, Network dismantling, Scalability

1. Introduction

Real-world systems are often represented as complex networks, for instance, energy systems (Albert et al. 2004), communication systems (Albert and Barabási 2002), transportation systems (Zanin and Lillo 2013), economic systems (Boss et al. 2004), or social systems (Duijn et al. 2014). Based on such complex network representations, network science-based analysis often reveals hidden patterns underlying the complex connection. Robustness is a particularly interesting property of a system, measuring its ability to withstand random failures and targeted attacks. Recent events of power outages (Ash and Newth 2007), large-scale transportation disruptions (Brooker 2010), and supply-chain failures (Kim et al. 2015) highlight the importance of better understanding critical systems enabling our modern society.

The underlying, critical step of quantifying the robustness of a network against targeted attacks is to design the most harmful attack, leading to a quick breakdown of the network into small components. For many real-world

[*] Corresponding author: Email: qqian.sun@gmail.com, Tel.: + 86 10 8233 8036

networks, researchers have reported a rather high robustness against random failures, but significant vulnerability towards intentional attacks, targeting important nodes preferably (Albert et al. 2000). Given the wide range of application domains, several studies have been performed in the area of network robustness, including the proposal of various node attacking strategies. Essentially, the goal of these techniques is to derive a node sequence which maximizes the damage on the network's connectivity, usually measured by the size of the largest connected component throughout an attack. Existing research has shown that the exact solution to this problem is computationally hard, even for medium-sized networks; see (Wandelt et al. 2018, Sun et al. 2017) for extensive comparative evaluations of these techniques. Early research on networks dismantling was mainly based on node centralities (Freeman 1977, Geisberger et al. 2008), which assign a numerical indicator to each node, reflecting the importance of that node with respect to a local or global topological measure. A network attack is constructed by ranking the nodes according to their importance in decreasing order. Such a construction is very natural, as an attack to a network is expected to add important nodes first. In addition to the exploitation of generic node centralities, researchers have designed methods specifically tuned towards the problem of dismantling a network (Braunstein et al. 2016, Tian et al. 2017, Ren et al. 2018, Fan et al. 2020b, Wandelt et al. 2020), which often aim to exploit the overall structure of the network to build attacks. Figure 1 provides an overview on the existing methods for generating node attacks to complex networks. The classification induces four quadrants described as follows. The first quadrant covers local node centralities which often can be computed in linear time regarding the size of the network, e.g., including the node degree. These methods are scalable, but often lead to rather ineffective attacks; unless being applied to networks with very special structures, which rarely exist in the real world. The second quadrant covers the area of rather uninformed methods, which are slow and do not lead to effective attacks. The third quadrant covers methods which are able to compute high-quality attacks, but do not scale up well when being applied to medium-sized or large networks. The fourth quadrant represents the desired area of any dismantling algorithm: Combining the generation of highly effective attacks with the property of being scalable to very large networks. Given the NP-hardness of the underlying node ordering problem, reaching this quadrant is extremely difficult.

In this study, we propose a conceptually novel method for the generation of targeted attacks to complex networks. In a nutshell, we describe an efficient transformation process that is able to convert a collection of random failure traces generated by purely random node sequences into highly effective attacks. Our method is inspired by recent works on the node explosive percolation (Qin et al. 2019), which was used to improve attacks obtained from belief propagation in complex networks. (Fan et al. 2020a) showed that node explosive percolation works on a wider range of inputs and recommended to use interactive degree together with node explosive percolation as a trade-off between effectiveness and efficiency. On top of these observations, we design and implement an iterative robustness

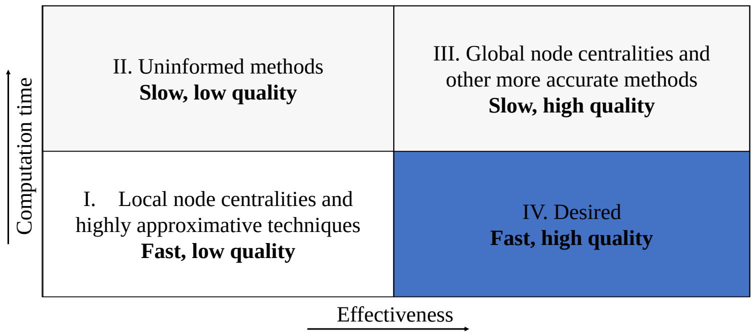

Figure 1: Classification of methods for network dismantling based on two dimensions: Scalability and effectiveness.

analysis framework, as shown in Figure 2. While in existing literature, the node explosive percolation is applied once, we show that repetitive application can significantly improve the quality while incurring low additional computation costs. Our method is evaluated against a wide range of state-of-the-art techniques in network dismantling, showing the outstanding efficiency of our method in all cases. Our work does not only lead to an excellent trade-off between attack quality and runtime, but also provides a novel view of the problem at hand: It is the first study to use zero-knowledge in order to generate attacks that sometimes even outperform the best baseline set by interactive betweenness-based attack generation; (Fan et al. 2020a) showed earlier that a single application of node explosive percolation to a random attack can outperform vanilla MinSum (Braunstein et al. 2016). We believe that this study has a wider impact beyond just proposing yet another network attacking procedure. An analysis of network based on repetitive node explosive percolation could yield important insights in various domains.

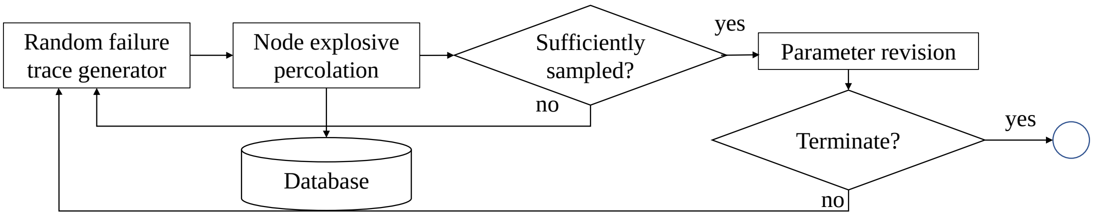

Figure 2: Overview on the transformation from random failures to targeted attacks.

The remainder of this study is organized as follows. Section 2 provides a literature review on studies related to network dismantling. Section 3 introduces the preliminaries and methodology of this study. Section 4 evaluates the efficiency and effectiveness of our novel dismantling method on a wide range of real-world networks and randomly generated networks. Section 5 concludes this study and provides a set of suggestions for future research.

Method	Time complexity	Remarks
D	O(N)	The degree of a node is a constant-time lookup for array-based adjacency lists
DI	O(N²)	After each node removal, D is computed on the remaining GCC
B	O(N*E)	
BI	O(N²E)	After each node removal, B is computed on the remaining GCC
PR	O(k*E)	Variable k is the number of iterations
PRI	O(N*k*E)	Variable k is the number of iterations
CI2	O(NlogN)	With ball size k=2
APTA	O(E)	Time for determining the articulation points
CHD	O(N*E)	After each node removal, the 2-core of the GCC is computed
COM	O(E)	
Our method	O(E*k)	Variable k is the number of iterations

Table 1: Time complexity of competing methods in this study.

2. Literature Review

Throughout recent decades, researchers have spent enormous efforts to estimate the robustness of a given complex network; see (Wandelt et al. 2018) for a recent survey and comparison of these techniques. These methods have inherent computational properties. In Table 1, we provide an overview on the time complexity of each method discussed in this study. It can be seen that some methods incur high computational costs, which prevents them to be applied to larger networks. For instance, a time complexity of $O(N^2)$ means that the number of computation steps is quadrupled once the size of the network is increased by a factor of two. Below, we describe the techniques from Table 1 in greater detail.

Many application studies have used node centralities to estimate the robustness of a network. It is beyond the scope of this study to review all centralities; the reader is referred to (Wandelt et al. 2018, Sun et al. 2017). The simplest metric is the degree (D), which counts the number of neighbors for node importance estimation. The degree is often used for two reasons. First, it is easy and efficient to compute, given that it can be read off directly from the graph representation as an adjacency list. Second, the degree of a node is highly related with a node's function as a hub in the network; which makes it potentially critical for the robustness of the whole network. It should be noted that node centralities can be computed in a dynamic way, where the centrality is recomputed after greedily removing the highest-ranked node. Such dynamic recomputations are often referred to as interactive attacks, in contrast to static attacks. An interactive version of degree (DI) iteratively attacks the node with the highest degree in the network and then recomputes the degree. Another frequently-used centrality metric is betweenness (Freeman 1977, Brandes 2008), which measures the frequency of a node appearing on all pairs shortest paths in the network. The exact calculation of betweenness centrality comes at a high computational cost, $O(NE)$ using Brandes' algorithm (Brandes 2001), where N is the number of nodes and E is the number of edges in network. Betweenness can be implemented as a static method (B) and as an interactive version (BI). A wide range of other centralities have been used, including closeness centrality, PageRank (PR), and eigenvector centrality. It has been shown in the literature that interactive betweenness is the best-suited node centrality for computing highly-destructive attacks to a network (Wandelt et al. 2018).

In recent years, several studies proposed methods specifically-tuned towards network dismantling. These techniques were often designed with scalability in mind and make trade-offs regarding the quality. In addition, these techniques are usually inspired by the idea to exploit specific topological properties, e.g., cycles or articulation points. Collective influence (CI) (Morone and Makse 2015), originally designed for identifying influential spreaders in a network, is the first method that was explicitly designed towards identifying the critical nodes in a network under limited computational resources. It is known that the concepts of influence and robustness are closely related, i.e., a node that is important for spreading is usually also critical for keeping the network connected. The collective influence of a node is computed by defining a neighborhood (usually called ball) around a node and sum up the degree of nodes on the frontier of the neighborhood. In hierarchical networks, the CI value can be easily computed in $O(N * logN)$ time, which means that the method can be efficiently computed for large networks; under exploitation of a max heap data structure (Morone et al. 2016).

Another technique, Min-Sum, assumes that for a large class of random graphs, the problem of network dismantling is connected to the decycling problem, which addressed the removal of vertices until a graph becomes acyclic (Braunstein et al. 2016). The authors propose a three-stage Min-Sum algorithm for efficiently dismantling networks, which are summarized as follows. Firstly, at the core of the algorithm is a variant of Min-Sum message passing for decycling, developed in (Altarelli et al. 2013a,b). The second step has the goal of tree breaking. After all cycles are broken, some of the tree components may still be larger than the desired threshold. These components are further broken into smaller components, removing a fraction of nodes that vanishes in the large size limit. Finally, cycles are closed greedily, in order to improve the efficiency of the algorithm with many short cycles. Belief propagation (Mugisha and Zhou 2016) is based on the Feedback Vertex Set (FVS) (Zhou 2013), which aims at selecting the set of nodes efficiently breaking the network into pieces. A variant of information spreading (Zhou 2013) is applied to FVS, in order to describe nodes' importance and to keep this information updated in each iteration of the method. Finally, nodes are ranked according to the resulting numerical value. Node Explosive Percolation (NEP) (Qin et al. 2019) aims to significantly improve existing attacks obtained by belief propagation; addressing the lack of order when removing nodes.

Articulation Point Targeted Attacks (APTA) (Tian et al. 2017) exploits the existence of so-called articulation points in a network. Such articulation points - upon removal - disconnect a network. All articulation points in a network can be computed in linear time of the number of edges, based on a variant of depth-first search. Conceptually, articulation points are highly critical for the connectedness of a network; accordingly, an efficient method based on such articulation points has very high potential for dismantling. In real-world networks, which often contain cycles, an attack needs to target redundant nodes, in order to effectively break the network into smaller components. Accordingly, the major limitation of APTA is observed once there are no (critical) articulation points left in a network. In this case, the attack is usually completed by falling back to a standard network metric, such as degree. APTA is a greedy method, in that it always selects the most-destructive articulation point in each iteration. FINDER (Fan et al. 2020b) proposes to use deep reinforcement learning to identifying key players in a network. The key idea is to learn node importance based on a set of training networks, e.g., obtained over a collection of Barabasi-Albert networks with different generating parameters, and then apply the learned model to previously-unseen networks for identifying the critical nodes. By using an adaptive sampling over the learned model for extracting key nodes, the prediction phase can be significantly accelerated, while obtaining a dynamic dismantling strategy.

Community-based network attacks (COM) (Wandelt et al. 2021) exploits the presence of communities in a network. Intuitively, by attacking the inter-community node/links in a network, it is ensured that the network is being dismantled quickly; see also (Requião da Cunha et al. 2015, Wandelt et al. 2020). By neglecting intra-community nodes, the time efficiency of community-based methods is often high, especially if clear community structures exist in the network. There are several design choices, including community detection methods and parameter thresholds. Approximately linear-time algorithms are chosen, such as the widely-used Louvain method (Blondel et al. 2008), which can scale up to networks with millions of nodes. A related method is based on spectrality (Zahedi and Khansari 2015) and uses Fiedler vectors (Newman 2013) together with a likelihood measure to break a network into two distinct partitions. Similarly, generalized network dismantling (Ren et al. 2018) disintegrates networks while taking into account node-specific costs, using spectral cuts with an efficient spectral approximation by a Power Laplacian operator. Moreover, the K-shell iteration factor (Wang et al. 2016) measures the coreness (Kitsak et al. 2010) of a node, combining shell decomposition and iterative node removal. Core High-Degree (CoreHD) (Zdeborová et al. 2016) combines interactive degree and k-core (Kitsak et al. 2010) to achieve a decycling of networks.

Several studies in the literature build customized models and analysis techniques for domain-specific problems. For instance, (Wu et al. 2021) designed an evolutionary algorithm for analyzing node importance in cyber-physical power systems, Chaoqi et al. (2021) use a game-model for describing the interactions between attacker and defender on critical infrastructure networks, and several other studies built upon various Bayesian approaches (Eldosouky et al. 2021, Dehghani et al. 2021). Another important recent development is the study of inter-connected or multi-modal networks (Liu et al. 2021, Munikoti et al. 2021, He et al. 2021). While our study, instead focuses on single-layer networks, the former studies include another level of complexity by allowing interactions between multiple layers. Several studies in the recent literature consider the flow on top of networks, for instance, in transportation systems (Zhang et al. 2021, Almotahari and Yazici 2021). Such studies are targeted for specific instances and can often lead to more detailed insights, compared to the purely structural or topological analysis.

3. Methodology

In this section, we propose our novel technique for transforming random failure traces into targeted attacks. The method is significantly faster than the state-of-the-art, while reaching attack quality on par with the best node centrality-based attacking strategy: interactive betweenness (BI).

3.1. (Un)informed network dismantling

When analyzing the robustness of a network, one can take several perspectives (Cohen et al. 2000, Newman 2010, Cohen and Havlin 2010, Callaway et al. 2000). First, statistical physicians are mostly interested in the phase transitions which lead to a sudden disintegration of the network. Second, operators are interested in identifying key nodes, for instance in critical infrastructure networks, that need to be protected better, in order to ensure a safe and robust operation of the system. Third, when conducting a cross-comparison of two or more network's robustness, there is a need to compare the overall performance of networks under disruption scenarios. While these perspectives have inherent peculiarities, they all share the goal of identifying critical nodes' order as an underlying operation on the network. The quality of the node order can be obtained by consideration of changes to the giant component during the evolution of an attack. Here, the intuition is that a smaller remaining giant component indicates a network that is considered to have been disintegrated (Newman 2003). Accordingly, this study refers to the commonly-used robustness measure R (Schneider et al. 2011). Given a network composed of N nodes, R is defined as $R = \frac{1}{N}\sum_{Q=1}^{N} s(Q)$, where $s(Q)$ represents the fraction of nodes in the giant component once the first Q nodes have been removed. Given that R allows for a comparison of two distinct attacks, one is usually interested in the attack with the minimum R value, concluding that this attack causes the maximum damage to the network. Note that the computation of this optimal attack (node order) is a NP-hard problem; and, accordingly, difficult to be solved towards optimality even for medium-sized networks. Therefore, many heuristics for finding good attacks have been proposed in the literature, see Section 2.

In order to better illustrate the problems that arise from using existing heuristics for attack generation, we show the results obtained for a special network: the so-called grid network. In this network, all nodes are layout out in a grid structure and neighbor nodes are connected by a link. This network is interesting for our purpose, since local node centralities are not able to distinguish nodes well; and standard dismantling techniques have significant problems when attacking such a network. Figure 3 illustrates a 30x30 nodes instance of a grid network and highlights the

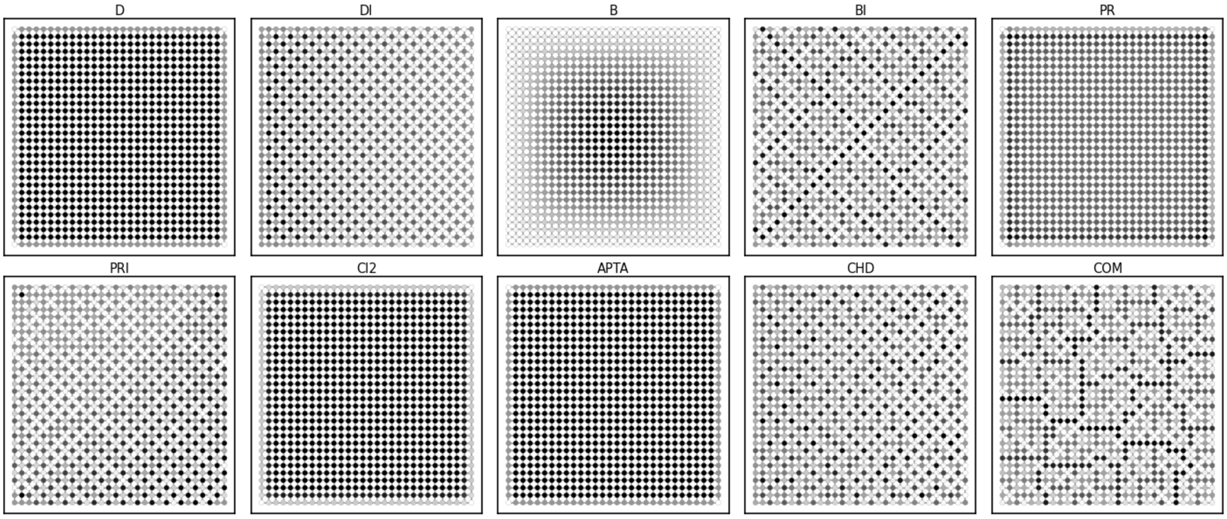

Figure 3: Node importance in a 30x30 grid network with state-of-the-art methods.

attack order for ten selected techniques in the literature: The node transition color indicates the position of a node in an attack, from black (attacked early) to white (attacked late). Degree (D) and collective influence (CI; with ball size 2) can only distinguish three and four types of nodes respectively, considering their local view on the network. Interactive degree (DI) and APTA can identify the inner parts of the network, but are unable to derive an effective attacking strategy; notably, APTA performs similar to degree, given the absence of articulation points in the network. PageRank (PR) identifies the transition between core and periphery as important nodes. Betweeness (B) is able to identify the important nodes from a static perspective, leading to the insight that inner nodes are most critical. Finally, Interactive Betweenness (BI), Interactive PageRank (PRI), and CoreHD (CHD) identify interesting attacks that split the network into smaller components, given their interactive design. Finally, Community-based attacking (COM)

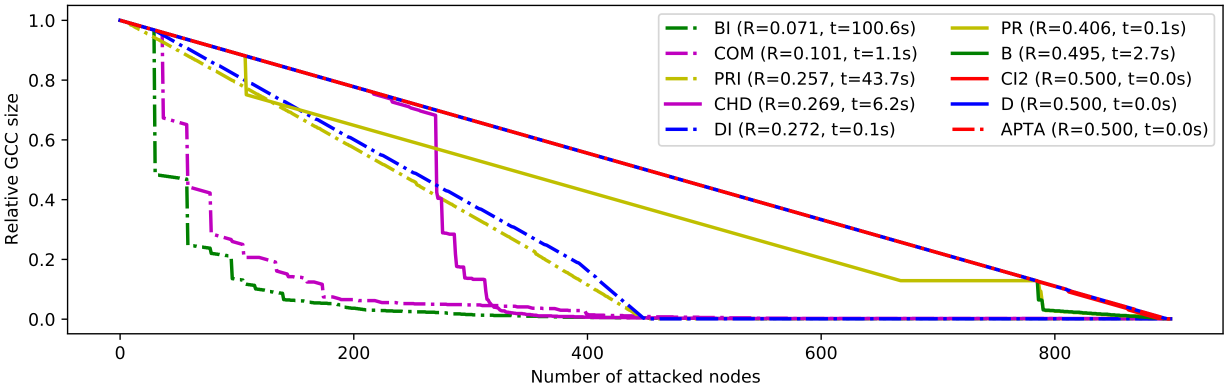

Figure 4: Comparison of robustness curves for a 30x30 grid network. Note that the curves for B, CI2, D, and APTA are hardly distinguishable, given that they have insignificant differences of the R values.

identifies horizontal and vertical cuts through the network, despite the existence of true communities in the grid network.

Figure 4 reports the robustness curves, obtained R values, and required runtime for all ten methods. It can be seen that these methods induce a wide range of different robustness traces and R values accordingly. The minimum R value is obtained by using BI. This is consistent with the results in the literature which showed that BI is the best attacking method. COM achieves competitive performance, but its R value is 0.101, compared to 0.071. The remaining methods yield significantly worse R values; some of the methods which cannot distinguish a sufficient number of node types yield essentially useless attacks, leading to maximum R values of 0.5. Another import insight from this grid network example is that the methods which yield better R values usually incur the longest runtime. For instance, BI already requires more than one minute of computation for this small network consisting of 900 nodes only. COM seems like a good trade-off, but as we will show in later experiments, COM does not scale up well for specific larger networks either. These results highlight the importance of understanding the classification of methods into fast, low-quality and slow, high-quality, as suggested in Figure 1 in the Introduction.

3.2. From random failures to targeted attacks

In the following, we describe the transformation of random failure traces to more effective targeted attacks. In general, it is known that random failures are less effective to dismantle real-world networks, compared to targeted attacks; this insight is not surprising, given the informed decision making in targeted attack generation (Zanin et al. 2018). In Figure 5, we visualize this effect on the 30x30 grid network. The figure is generated by evaluating 100,000 randomly-generated attacks on the network. The median R value is approximately 0.34; with few attacks reaching R values smaller than 0.3 and larger than 0.38, respectively. In order to compare the effectiveness range of random failures to those of targeted attacks, the R values for selected targeted attacks are shown as vertical lines.

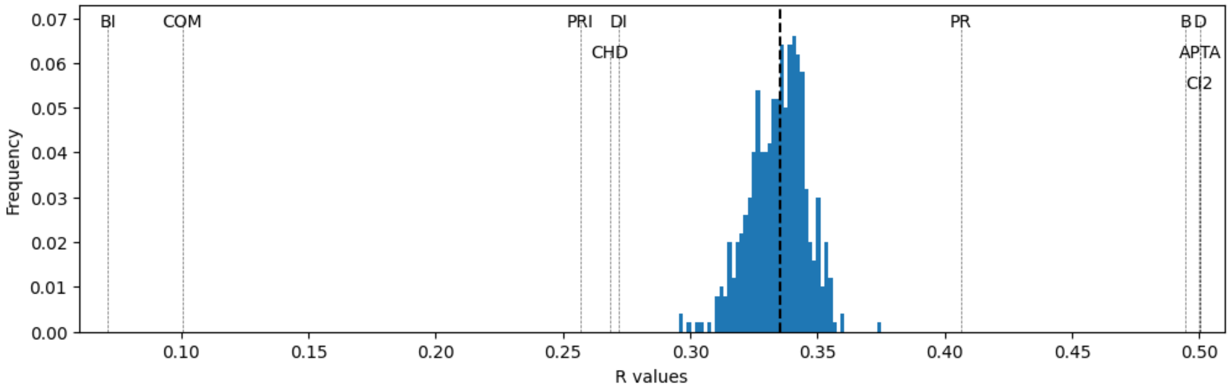

Figure 5: Effectiveness of random attacks to the 30x30 grid network, as indicated by the relative frequency distribution over all R values for 100,000 randomly-generated attacks.

The transformation process is inspired by recent work on explosive node percolation. In a first step, we compute a score for each node as induced by the reversely-constructed node neighborhood, as shown in Algorithm 1. The algorithm computes the score for node *u* based on the currently rebuilt network *G* and the corresponding disjoint-set data structure based on the UnionFind algorithm. UnionFind as an algorithm which is based on a disjoint-set data structure that keeps track of items forming an equivalence relation under updates. The name UnionFind comes from its two efficient core operations over items: Union (merging the equivalent sets of two items) and Find (identifying the equivalent set of an item). BuiltNeighbor(u) represents the currently neighborhood of *u*.

In the next step, the scores obtained by Algorithm 1 are used to convert one random attack into a targeted attack. The overall process is described in Algorithm 2. The random attack *R* is traversed from the end to the front; adding nodes back to the (dismantled) network, gradually building up the reversed attack *L*. This process is repeated as long as all largest sets in UF are of size at most one. Afterwards, the random attack is processed in batches of length S, appending the top-ranked S nodes to *L*, while maintaining the UnionFind-based data structure updated, i.e., merging the components correspondingly to edge additions. Finally, the list *L* is reversed and the targeted attack *A* is obtained. The batch size has an important impact here: It nicely controls the accuracy of this algorithm and allows to explore interesting trade-offs between quality and required computational resources, when transforming a single random failure trace into a targeted attack. Figure 6 visualizes the process of computing the scores iteratively.

In Figure 7, the results of the transformation is shown. Given the 30x30 grid network, a random attack is generated, whose R value is 0.3201; considerable worse than many targeted attack strategies (which reach R values of 0.071). After applying the transformation described in Algorithm 2, the attack's quality has improved significantly with a R value of 0.1248, now outperforming PRI, CHD, and DI. This is a reduction by a factor of approx. 66%. The inset of the upper right of Figure 7 reveals that while the random attack is essentially causing to the site-percolation to the grid, the transformed attack is cutting the grid into three components. Although this attack is not optimal on a grid network, this improvement is remarkable, given that the input of the transformation process was a purely random attack.

Algorithm 1 Function for node scores

```
Input: Graph G, Node u, UnionFind UF
Output: Score of the node n
$S \leftarrow \{\}$
for $v \in G.BuiltNeighbor(u)$ do
    $S.add(UF.FindSet(v))$
end for
$Score \leftarrow Sum\ of\ UF.SizeOfSet(s)\ \forall s \in S$
$Score \leftarrow Score * S.size()$
return Score
```

Algorithm 2 Node explosive percolation for random attacks

```
Input: Graph G, Random attack R, ReInsertStep st
Output: Attack A
$L \leftarrow [\,]$
$UF \leftarrow UnionFind(G)$
while $UF.LargestSetSize == 1$ do
    $u = popFromBack(R)$
    $UF.Union(u, G.BuiltNeighbor(u))$
    $L.append(u)$
end while
while $len(R) > 0$ do
    $Scores \leftarrow \{node : NEP(G, u, UF)\} \quad \forall u \in R$
    $R \leftarrow sort\ R\ by\ increasing\ order\ of\ Score$
    $Batch \leftarrow R[0 : min(size(R), st)]$
    $L+ = Batch$
    for $node \in Batch$ do
        $UF.Union(node, G.BuiltNeighbor(node))$
    end for
    $R = R[min(size(R), st) :]$
end while
$A \leftarrow reversed(L)$
return $A$
```

3.3. Framework for attack space exploration

This subsection presents the overall iterative framework for attack generation, based on the transformation step in the previous subsection. The transformation step has two degrees of freedom. First, it comes with a parameter describing the reinsertion step; a measure which can be used for balancing effectiveness and quality. Intuitively, the larger the reinsertion step is chosen, the faster the transformation can be computed. The attack quality, however, is likely to reduce with larger reinsertion batches. Accordingly, controlling this parameter is particularly important from a scalability point of view. Second, given the non-deterministic characteristic of the input (as random failure traces), a question arises: How many times does one need to iterate the attack generation algorithm, in order to obtain a strong attack to the network? In other words, when can we stop sampling, in expectation of future insignificant improvements? These two parameters, reinsertion step and termination condition need to be carefully chosen; the details are discussed below.

Algorithm 3 describes the individual steps in detail. The algorithm makes use of two termination criteria: T_1 and T_2. The parameter T_1 is used to control the maximum number of non-improvement iterations. The input variable T_2 controls the number of consecutive samplings with the same reinsertion step length. Variable L collects the existing random attacks throughout one iteration; after one round of sampling, the reinsertion step *st* is halved and the process repeated, until the termination criterion T_1 is satisfied.

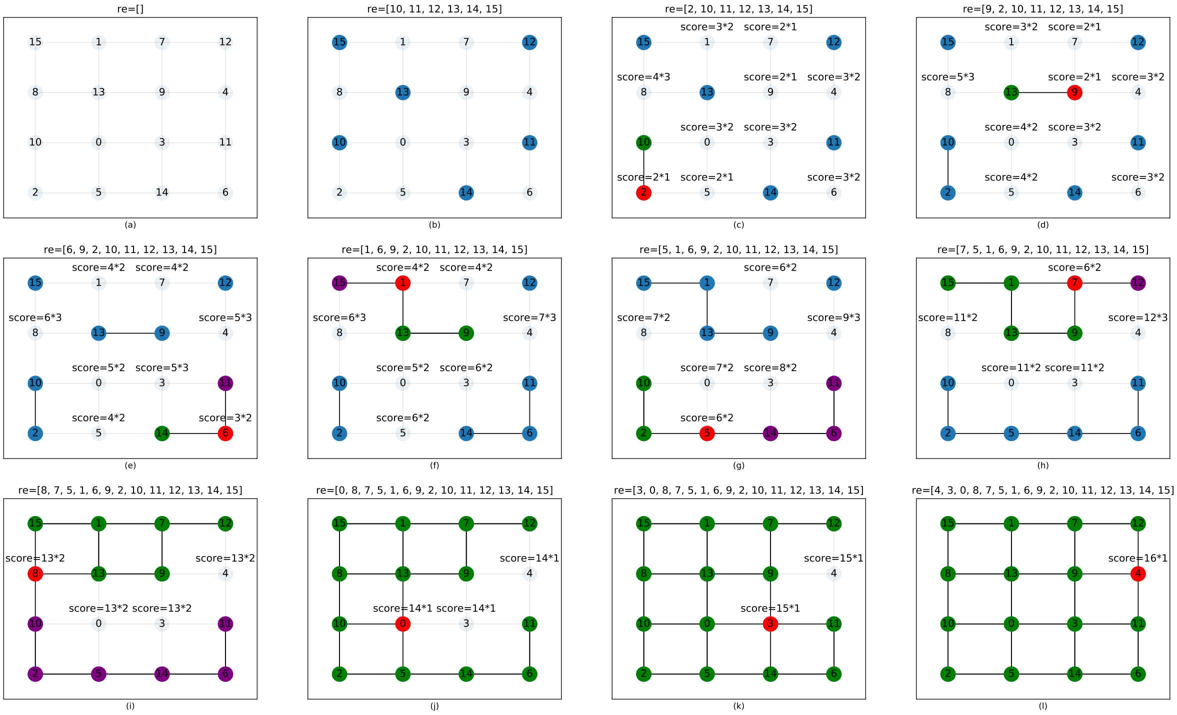
re=[]
re=[10, 11, 12, 13, 14, 15]
re=[2, 10, 11, 12, 13, 14, 15]
re=[9, 2, 10, 11, 12, 13, 14, 15]
(a)
(b)
(c)
(d)
re=[6, 9, 2, 10, 11, 12, 13, 14, 15]
re=[1, 6, 9, 2, 10, 11, 12, 13, 14, 15]
re=[5, 1, 6, 9, 2, 10, 11, 12, 13, 14, 15]
re=[7, 5, 1, 6, 9, 2, 10, 11, 12, 13, 14, 15]
(e)
(f)
(g)
(h)
re=[8, 7, 5, 1, 6, 9, 2, 10, 11, 12, 13, 14, 15]
re=[0, 8, 7, 5, 1, 6, 9, 2, 10, 11, 12, 13, 14, 15]
re=[3, 0, 8, 7, 5, 1, 6, 9, 2, 10, 11, 12, 13, 14, 15]
re=[4, 3, 0, 8, 7, 5, 1, 6, 9, 2, 10, 11, 12, 13, 14, 15]
(i)
(j)
(k)
(l)

Figure 6: Example for the computation of scores and ranking. L is the list of attacked nodes. During the construction of L, i.e., following subfigures from (a) to (l), nodes are appended to the front of L. Four node colors are used for indicating node status: red (currently added node), blue (inactive components), green / purple (individual components being connected by adding the red node. If the added node does not connect at least two individual components, then all nodes in the component are colored green).

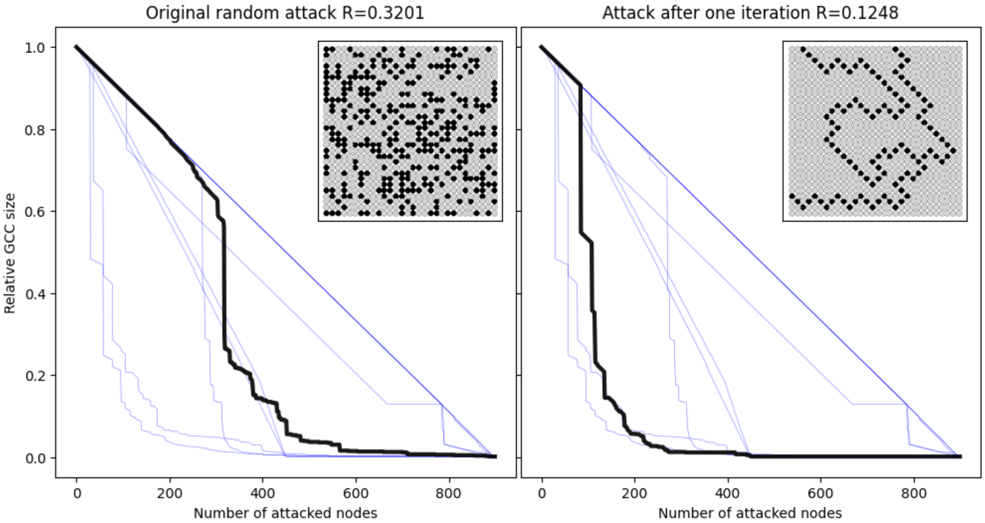

Figure 7: Visualization for the effect of transforming a random attack (left) to a targeted attack (right). The solid black line represents the evolution of the giant component. The inset in the upper right visualizes attacked nodes until the size of the giant component is less than 50%. Blue lines in both plots correspond to state-of-the-art methods for targeted attacks (see Figure 4 for reference).

Algorithm 3 Main algorithm

```
Input: Graph G, termination criteria $T_1, T_2$
Output: Attack $A_{best}$
$st \leftarrow \frac{|G|}{10}$
$A_{best} \leftarrow$ random attack for $G$
$nonimprov \leftarrow 0$
while $nonimprov < T_1$ do
  $L \leftarrow [\ ]$
  for $i \in \{1, ..., T_2\}$ do
    $A_{rand} \leftarrow$ random attack for $G$
    Let $A$ be the result of Algorithm 2 applied on $A_{rand}$
    $L.append(A)$
  end for
  $A \leftarrow$ best attack in $L$
  if $A$ is more effective than $A_{best}$ then
    $A_{best} = A$
  else
    $nonimprov = nonimprov + 1$
  end if
  $st \leftarrow \frac{st}{2}$
end while
return $A_{best}$
```

4. Experimental evaluation

This section reports the results of experiments for evaluating the efficiency and effectiveness of our proposed method. Section 4.1 describes the experimental setup, including the method selection and the evaluated networks. Section 4.2 performs a sensitivity analysis on the termination criteria, setting the baseline for additional experiments. Section 4.3 reports the results of an extensive comparison of state-of-the-art methods on real-world networks.

4.1. Experimental setup

In this section, we compare our novel methods against seven selected methods from the literature. These methods are listed as follows, together with the rationale for choosing them. We have included degree (D) and interactive degree (DI), given their prevalent usage in studies on network robustness; overall, the degree is probably one of the most-frequently used network metrics to assess the local importance of nodes in the network. Similarly, we have included betweenness (B) and interactive betweenness (BI), given their role for identifying the criticality of nodes based on global network properties. Notably, BI can be considered the reference baseline in terms of attack quality. We include experiments on collective influence (CI2), given its wide usage in many different network science domains. Moreover, we report results on articulation point targeted attacks (APTA), which has been shown in the literature to outperform Min-Sum; and also include community-based attacks (COM), which have been shown recently to be the best trade-off between attack quality and runtime on medium-sized networks. The experiments are executed on a broad range of network types; see Table 2 and Table 3 for an overview on the random and real-world networks in this study, respectively. The random networks and toy networks have been created with the generators available in the Python package networkx. The name of the network corresponds to the chosen parameters, e.g., for BA networks the name 120,3 indicates that the network was generated with 120 nodes and the number of links to attach for each node was set to three. The two tables report common network properties, as well as the best-known R values from state-of-the-art methods, in order to assess the overall robustness of these networks. Since our method is non-deterministic by design, we report the median time and R values over five experiments.

The real-world networks in this study have been regularly used for complex network robustness evaluation in related work; they come from a wide range of domains and cover networks from very fragile to rather robust. We briefly describe their domain of interest below; see Table 3 for complex network properties:

1. Network *dolphins* represents the Doubtful Sound community of bottlenose dolphins. The connectivity of individuals in the network follows a scale-free power-law distribution.
2. Network *polbooks* consists of books related to US politics.
3. Network *adjnoun* has anti-community structure, i.e., inter-community edges are denser than the intra-community edges

Table 2: Overview on random and toy networks in this study.

Network type	Name	|N|	|L|	Density	Cluster. Coeff.	|Bridges|	|APs|	Best R value
BA preferential attachment	30,1	30	29	0.06667	0.000000	29	13	0.10667
BA preferential attachment	70,2	70	136	0.05631	0.132228	0	0	0.15837
BA preferential attachment	120,3	120	351	0.04916	0.123559	0	0	0.20326
BA preferential attachment	200,1	200	199	0.01000	0.000000	199	67	0.01865
BA preferential attachment	250,2	250	496	0.01594	0.063795	0	0	0.10690
BA preferential attachment	300,3	300	891	0.01987	0.086945	1	1	0.16103
Erdős-Rényi	30,0.15	30	72	0.16552	0.187063	1	1	0.30556
Erdős-Rényi	30,0.25	30	106	0.24368	0.272513	0	0	0.38333
Erdős-Rényi	80,0.15	80	462	0.14620	0.135194	0	0	0.41484
Erdős-Rényi	80,0.25	80	778	0.24620	0.252669	0	0	0.45609
Erdős-Rényi	200,0.05	200	985	0.04950	0.043690	0	0	0.37047
Erdős-Rényi	200,0.1	200	1886	0.09477	0.093220	0	0	0.44785
Grid-shaped	10x10	100	180	0.03636	0.000000	0	0	0.17140
Grid-shaped	20x20	400	760	0.00952	0.000000	0	0	0.10064
Grid-shaped	30x30	900	1740	0.00430	0.000000	0	0	0.07113
Tree-shaped	2,127	127	126	0.01575	0.000000	126	63	0.03863
Tree-shaped	4,341	341	340	0.00587	0.000000	340	85	0.01370
Tree-shaped	3,364	364	363	0.00549	0.000000	363	121	0.01513
Watts–Strogatz small-world	30,4	30	60	0.13793	0.468889	0	0	0.24556
Watts–Strogatz small-world	70,4	70	140	0.05797	0.419524	0	0	0.14408
Watts–Strogatz small-world	120,4	120	240	0.03361	0.416944	0	0	0.12722
Watts–Strogatz small-world	200,5	200	400	0.02010	0.455333	0	0	0.08008
Watts–Strogatz small-world	250,5	250	500	0.01606	0.432667	0	0	0.09650
Watts–Strogatz small-world	300,5	300	600	0.01338	0.438889	0	0	0.07896

Table 3: Overview on real-world networks in this study.

Network type	Name	|N|	|L|	Density	Cluster. Coeff.	|Bridges|	|APs|	Best R value
Small	dolphins	62	159	0.08408	0.258958	9	7	0.17742
Small	polbooks	105	441	0.08077	0.487527	0	0	0.18776
Small	adjnoun	112	425	0.06837	0.172840	10	9	0.23143
Small	Moscow_subway	178	211	0.01339	0.060861	96	96	0.05511
Small	celegans	297	2148	0.04887	0.292363	15	3	0.21048
Small	usair	332	2126	0.03869	0.625217	56	27	0.07674
Small	polblogs	1222	16714	0.02240	0.320255	139	89	0.15767
Small	Helsinki_bus	2010	2555	0.00127	0.096635	321	314	0.01725
Large	ca-HepPh	11204	117619	0.00187	0.621582	1178	1122	0.12155
Large	Reuters911	13308	148035	0.00167	0.368648	668	557	0.11164
Large	p2p-Gnutella25	22663	54693	0.00021	0.005314	9310	4273	0.11030
Large	cond-mat-2005	36458	171736	0.00026	0.656585	2865	3049	0.10889
Large	rec-amazon	91813	125704	0.00003	0.268215	41833	39958	0.00276
Large	road-luxembourg-osm	114599	119666	0.00002	0.000606	23076	22426	0.00120

4. Network *Moscow_subway* represents the subway system of Moscow, Russia.
5. Network *celegans* corresponds to the metabolic network of C. elegans
6. Network *usair* represents the United States airport network.
7. Network *polblogs* represents the United States political blogosphere network; each node represents a blog and each edge represents a hyperlink between two blogs.
8. Network *Helsinki_bus* represents the bus transportation system of Helsinki, Finland.
9. Network *ca-HepPh* covers scientific collaborations between authors and papers submitted to High Energy Physics - Phenomenology category at arxiv.org in the period from January 1993 to April 2003.
10. Network *Reuters911* is based on all stories released during 66 consecutive days by the news agency Reuters concerning the September 11 attack on the US.
11. Network *p2p-Gnutella25* is a snapshot of the Gnutella peer-to-peer file sharing network from August 2002, with nodes being hosts and links representing connections between hosts.
12. Network *cond-mat-2005* represents a collaboration network based on preprints in condensed matter archive arxiv.org.
13. Network *rec-amazon* represents the relationships between customers and items at amazon.com
14. Network *road-luxemburg-osm* represents the road network of Luxemburg based on data from Openstreetmap.

4.2. Sensitivity Analysis

In a first set of experiments, we perform parameter tuning on the proposed framework. The two parameters to be explored are a) the number of non-improvement iterations during reinsertion step reduction (T_1) and b) the number of random attack generations (T_2). The purpose of the sensitivity analysis is to identify combinations of both parameters

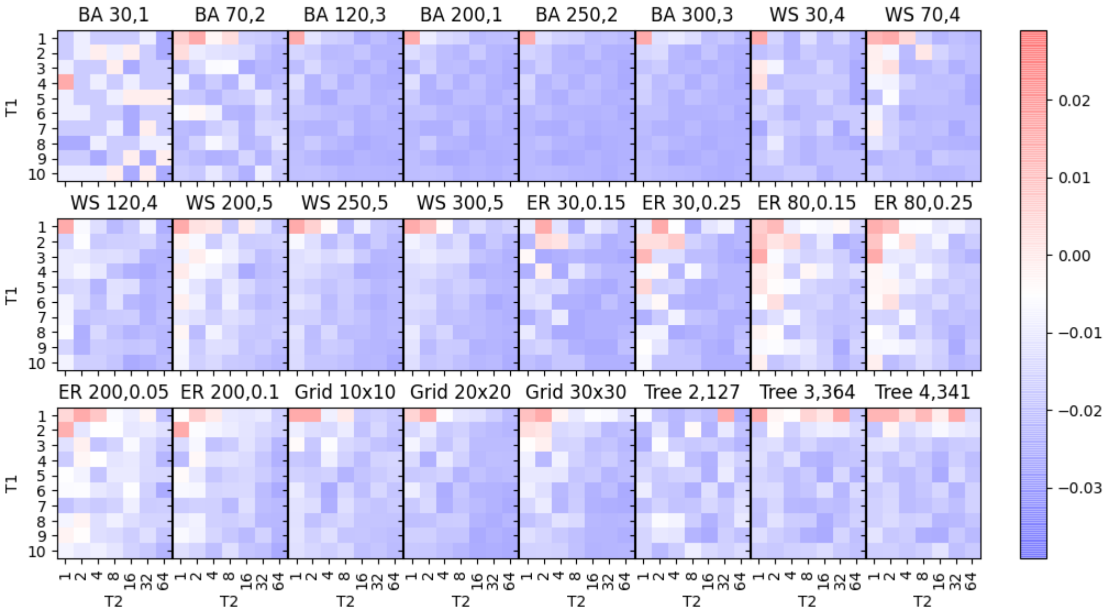

Figure 8: Sensitivity analysis regarding T_1 and T_2 with respect to quality on random and toy networks.

that lead to an interesting trade-off between attack quality and runtime efficiency. Intuitively speaking, the increment of both parameters individually is expected to increase the runtime linearly; the non-determinism of T_1 makes accurate predictions for the actual runtime infeasible. We compare instances of our method with $T_1 \in 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ and $T_2 \in 1, 2, 4, 8, 16, 32, 64$ against the baseline attack obtained by interactive betweenness (BI). Notably, BI is known as the best attacking method in the literature, regarding the obtained attack quality (Wandelt et al. 2018). In Figure 8 we report the results of these experiments regarding the absolute difference between obtained R values from our method and those obtained by BI. Cells highlighted in blue color indicate results better than BI and cells marked in red color indicate results worse than BI. Only for very small values of T_1 and T_2, our method is outperformed by BI. The reason is that too small values of the parameters lead to early terminations, and, accordingly, to an insufficient exploration of the attack space. With $T_1 > 3$ and $T_2 > 3$, our method consistently outperforms BI, by R differences of 0.02 and more. This result is remarkable; especially, since it holds across these various types of networks, which come with rather different topologies. In addition, it should be noted that those cells highlighted in white color still represent instances of our algorithm which perform on par with BI; which is striking, given the simplicity and randomness-based structure of our method.

Figure 9 reports the runtime differences between selected parameter combinations and BI. A large impact of the parameter choice on the runtime can be observed; the maximum runtime being two orders of magnitude longer than the ones with shorter runtime. The variants with $T_1 \in \{1, 4\}$ and $T_4 \in \{1, 4\}$ reveal runtimes of less than a second for

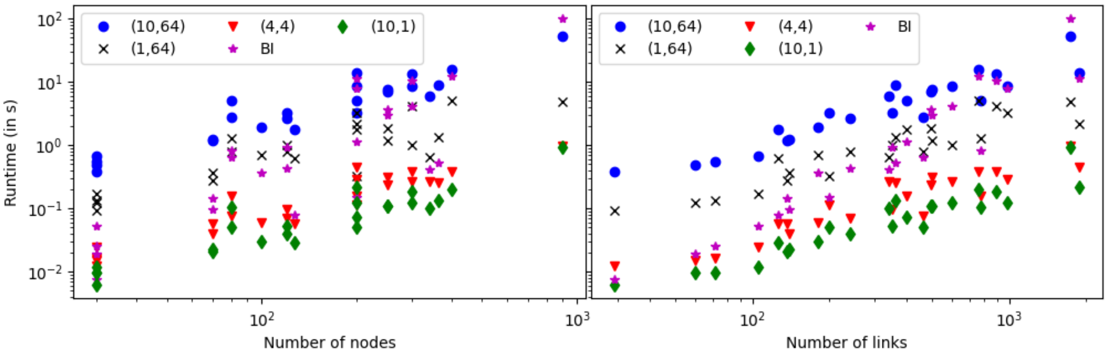

Figure 9: Sensitivity analysis regarding T_1 and T_2 with respect to runtime on random networks.

these networks with up to 1000 nodes; being more than two orders of magnitude faster than BI on the largest network. Accordingly, combining the insights from Figure 8 and Figure 9, we recommend to use $T_1 \in \{4\}$ and $T_2 \in \{4\}$, i.e., four iterations for non-improvement and four iterations for random exploration, as an interesting sweet spot regarding quality and computation time. In the remainder of this section, we use this combination for our experiments.

4.3. Comparison on real-world networks

In the next set of experiments, we compare our method against a set of state-of-the art method on real-world networks. We have split these experiments into two parts: smaller and larger networks. The rationale is that for small networks more methods can be compared, particularly, we include interactive betweenness as a baseline reference. Figure 10 reports the results of experiments on small real-world networks. It can be seen that the two top-ranked methods according to the quality are BI and and our method in all eight cases, often with minor mutual differences between R values for their attacks. COM and APTA often obtain good results as well; but their performance is significantly worse than BI and our method, depending on the network type. Another interesting observation can be made regarding the runtime: While BI is competitive for execution on smaller networks, its computation complexity increased significantly with the number of nodes and links in the network. For instance, BI needs 20 minutes to compute the results for network polblogs, while the results of our method can be obtained in eight seconds. The other methods can be computed in the same amount of time or less; particularly D and CI2 stand out regarding the required time, but also lead to rather bad attacks. Overall, we conclude our method indeed addresses an interesting and important sweet spot between attack quality and computational resources. In addition, with choosing less-conservative parameters for T_1 and T_2, the attack efficacy can be improved further, outperforming BI, at the cost of longer runtime.

For the next experiments, we investigate the performance on larger real-world networks. Accordingly, we need to discard methods which do not scale up to network scales with hundreds of thousands of nodes. This includes betweenness-based methods. The time complexity of BI is at least cubic in the number of nodes. Accordingly, once

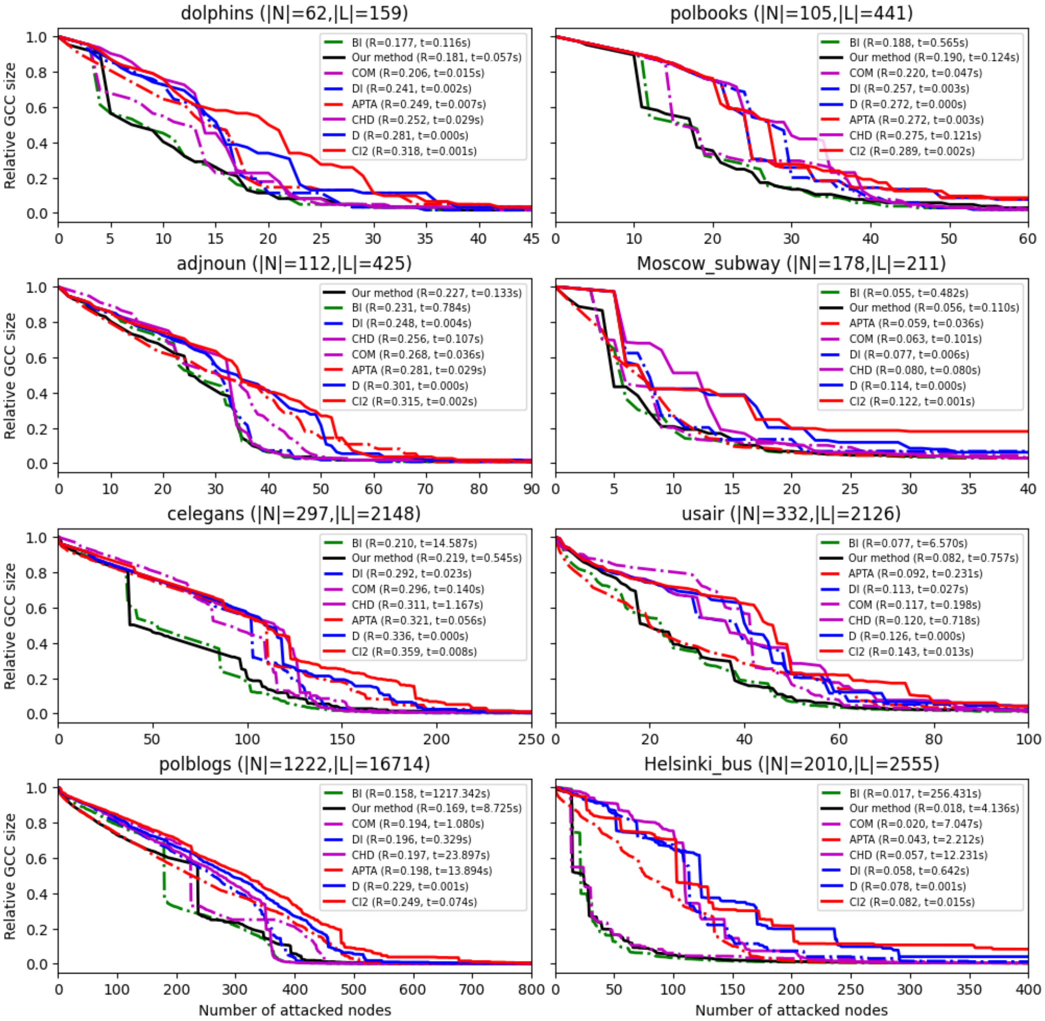

Figure 10: Experimental results on small to medium-sized real-world networks.

the number of nodes is increased by a factor of two, the runtime of BI is expected to be at least eight times higher. The networks in this subsection have up to 10^5 nodes and more, which leads to a lower bound of 10^{15} atomic steps (where each atomic step takes constant time). The results of our experiments on larger real-world networks are shown in Figure 11. For three out of six networks, our method finds the best attack (ca-HepPh, Reuters911, and cond-mat-2005). In the remaining three cases, the attacks by COM (2x) and APTA (1x) are slightly better than those obtained by our method. The runtime in these cases, however, is 1-2 orders of magnitude faster, highlighting the scalability of our method to very large networks. For instance, for the network road-luxemburg-osm, which contains more than 100,000 nodes, our method requires about two minutes to compute its strong attack, while APTA requires 1.5 hours and COM almost four hours. Finally, it should be noted that the performance of APTA and COM is largely network dependent; e.g., COM performs worst among all methods on the network Reuters911 and the gap between APTA and

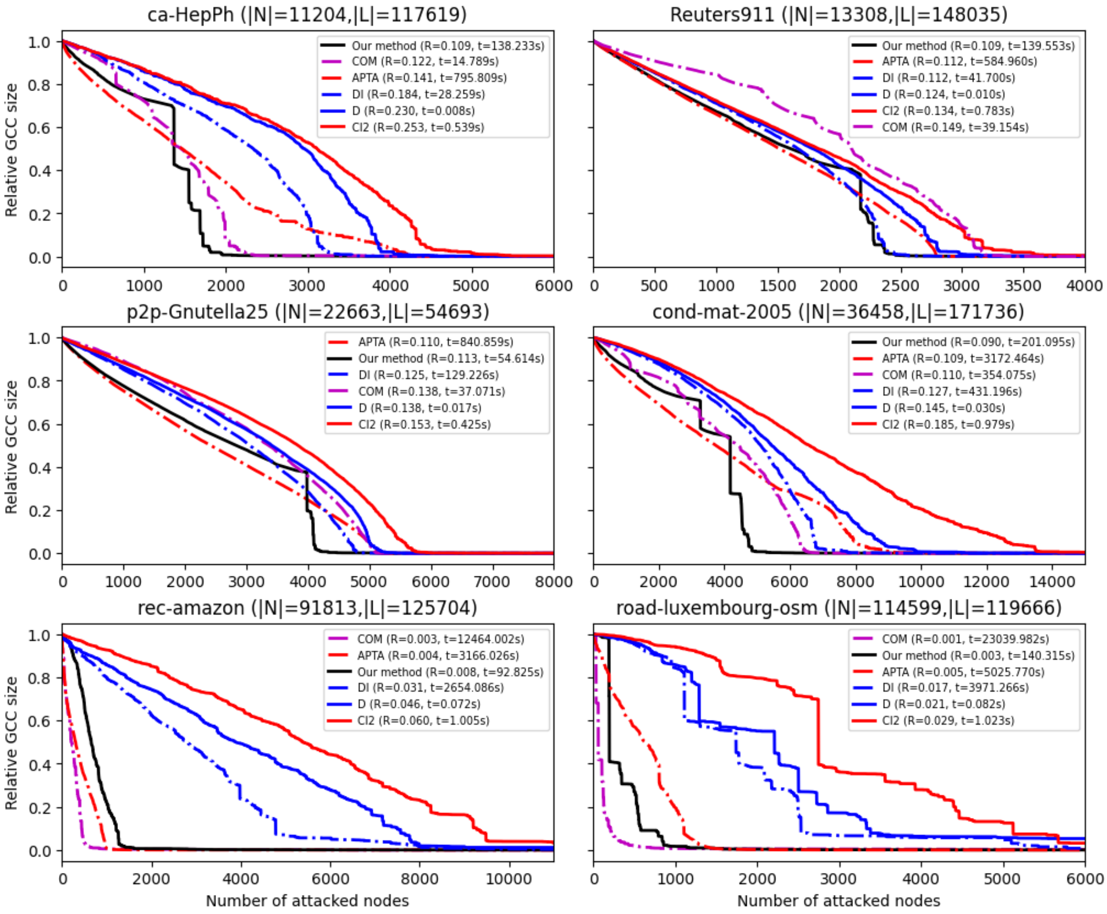

Figure 11: Experimental results on large real-world networks.

our method is significant for ca-HepPh. Accordingly, the generality of the random-attack based framework should be recognized based on these results.

5. Conclusions

The pursue of an optimal attack on complex networks has a long history in graph theory and network science. The plethora of methods proposed by researchers has significantly improved our understanding of network robustness over time, and the contribution of steps like decycling, tree-breaking, articulation points, bridges, and communities. This study contributes to the literature by showing how random attacks can be turned into highly effective targeted attacks by a transformation process, based on explosive node percolation. An iterative framework is proposed for the scalable computation of network attacks, which gradually performs more detailed attack revisions until a set of termination criteria is satisfied. Our experimental results on random networks and several real-world networks of different sizes highlight the efficiency and efficacy of our method compared to the state-of-the-art. Particularly, our method is the only one that consistently works on all network types, independent of the topology. For smaller networks, it is shown that interactive betweenness centrality can be outperformed. Our method is significantly faster for large networks than the state-of-the-art, while not losing too much attack quality.

The structure of a network is a significant factor for deciding the robustness towards a targeted attack. For instance, if a network has few articulation points, then APTA is significantly less effective. CI, on the other hand, presumably works best in hierarchical networks and in networks with a short diameter (such that the central nodes can be distinguished as being central with a small ball size k). COM relies on the presence of communities in the network. Our novel method does not have such strong dependencies on the network structure: We begin network dismantling with a random attack. This initial attack – by definition - is not subject to any initial heuristic; the initial sequence does not rely on other local/global properties of the network. This iterative characteristic is the key novel scientific message of our study: While a single iteration based on one random attack might be still far away from being competitive, after a few iterations this deterioration disappears. This perspective is new in the literature; existing work builds iterative/adaptive attacks on the node level, i.e., remove nodes and then recompute a measure on the remaining network for a partial attack. Our iterative computation takes place on an attack level, where a given attack is gradually improved over time.

Our study opens up several avenues for future research. First of all, given the novel perspective on network dismantling, exploiting randomness in the input, hopefully inspires researchers to design other powerful non-deterministic methods for network dismantling. Specifically to our framework, future studies could consider the design of different termination criteria and investigate more informed methods to randomize the node explosive percolation process.

Acknowledgments

This study is supported by the National Natural Science Foundation of China (Grants No. 61861136005, No. 61851110763).

References

Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

Réka Albert, Hawoong Jeong, and Albert-László Barabási. Error and attack tolerance of complex networks. *Nature*, 406:378–382, 2000.

Réka Albert, István Albert, and Gary L Nakarado. Structural vulnerability of the north american power grid. *Physical Review E*, 69(2):025103, 2004.

Amirmasoud Almotahari and Anil Yazici. A computationally efficient metric for identification of critical links in large transportation networks. *Reliability Engineering & System Safety*, 209: 107458, 2021. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2021.107458. URL `https://www.sciencedirect.com/science/article/pii/S0951832021000260`.

Fabrizio Altarelli, Alfredo Braunstein, Luca Dall'Asta, and Riccardo Zecchina. Large deviations of cascade processes on graphs. *Physical Review E*, 87(6):062115, 2013a.

Fabrizio Altarelli, Alfredo Braunstein, Luca Dall'Asta, and Riccardo Zecchina. Optimizing spread dynamics on graphs by message passing. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(09):P09011, 2013b.

Jeff Ash and David Newth. Optimizing complex networks for resilience against cascading failure. *Physica A: Statistical Mechanics and its Applications*, 380:673–683, 2007.

Vincent Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics Theory and Experiment*, 2008, 04 2008. doi: 10.1088/1742-5468/2008/10/P10008.

Michael Boss, Helmut Elsinger, Martin Summer, and Stefan Thurner. Network topology of the interbank market. *Quantitative Finance*, 4(6):677–684, 2004.

Ulrik Brandes. A faster algorithm for betweenness centrality. *The Journal of Mathematical Sociology*, 25(2):163 – 177, 2001. doi: https://10.1080/0022250X.2001.9990249.

Ulrik Brandes. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 30(2):136 – 145, 2008. ISSN 0378-8733. doi: http://dx.doi.org/10.1016/j.socnet.2007.11.001.

Alfredo Braunstein, Luca Dall'Asta, Guilhem Semerjian, and Lenka Zdeborová. Network dismantling. *Proceedings of the National Academy of Sciences*, page 201605083, 2016.

Peter Brooker. Fear in a handful of dust: aviation and the icelandic volcano. *Significance*, 7(3):112–115, 2010.

Duncan S Callaway, Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Network robustness and fragility: Percolation on random graphs. *Physical review letters*, 85(25):5468, 2000.

Fu Chaoqi, Gao Yangjun, Zhong Jilong, Sun Yun, Zhang Pengtao, and Wu Tao. Attack-defense game for critical infrastructure considering the cascade effect. *Reliability Engineering & System Safety*, 216:107958, 2021. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2021.107958. URL `https://www.sciencedirect.com/science/article/pii/S0951832021004695`.

Reuven Cohen and Shlomo Havlin. *Complex networks: structure, robustness and function*. Cambridge University Press, 2010.

Reuven Cohen, Keren Erez, Daniel Ben-Avraham, and Shlomo Havlin. Resilience of the internet to random breakdowns. *Physical review letters*, 85(21):4626, 2000.

Nariman L. Dehghani, Soroush Zamanian, and Abdollah Shafieezadeh. Adaptive network reliability analysis: Methodology and applications to power grid. *Reliability Engineering & System Safety*, page 107973, 2021. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2021.107973. URL https://www.sciencedirect.com/science/article/pii/S095183202100483X.

Paul AC Duijn, Victor Kashirin, and Peter MA Sloot. The relative ineffectiveness of criminal network disruption. *Scientific reports*, 4:4238, 2014.

AbdelRahman Eldosouky, Walid Saad, and Narayan Mandayam. Resilient critical infrastructure: Bayesian network analysis and contract-based optimization. *Reliability Engineering & System Safety*, 205:107243, 2021. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2020.107243. URL https://www.sciencedirect.com/science/article/pii/S0951832020307432.

Changjun Fan, Zeng li, Yanghe Feng, Baoxin Xiu, Jincai Huang, and Zhong Liu. Revisiting the power of reinsertion for optimal targets of network attack. *Journal of Cloud Computing*, 9, 05 2020a. doi: 10.1186/s13677-020-00169-8.

Changjun Fan, Zeng Li, Yizhou Sun, and Yang-Yu Liu. Finding key players in complex networks through deep reinforcement learning. *Nature Machine Intelligence*, 2:1–8, 06 2020b. doi: 10.1038/s42256-020-0177-2.

Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41, 1977.

Robert Geisberger, Peter Sanders, and Dominik Schultes. Better approximation of betweenness centrality. In *Proceedings of the Meeting on Algorithm Engineering & Expermiments*, pages 90–100, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics. URL http://dl.acm.org/citation.cfm?id=2791204.2791213.

Zhidong He, Kumar Navneet, Wirdmer van Dam, and Piet Van Mieghem. Robustness assessment of multimodal freight transport networks. *Reliability Engineering & System Safety*, 207: 107315, 2021. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2020.107315. URL https://www.sciencedirect.com/science/article/pii/S0951832020308097.

Yusoon Kim, Yi-Su Chen, and Kevin Linderman. Supply network disruption and resilience: A network structural perspective. *Journal of Operations Management*, 33:43–59, 2015.

Maksim Kitsak, Lazaros K. Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H. Eugene Stanley, and Hernan A. Makse. Identification of influential spreaders in complex networks. *Nature Physics*, 6:888–893, 2010. doi: 10.1038/nphys1746.

Xing Liu, Yi-Ping Fang, and Enrico Zio. A hierarchical resilience enhancement framework for interdependent critical infrastructures. *Reliability Engineering & System Safety*, 215: 107868, 2021. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2021.107868. URL https://www.sciencedirect.com/science/article/pii/S0951832021003872.

Flaviano Morone and Hernan A Makse. Influence maximization in complex networks through optimal percolation. *Nature*, 524, 2015.

Flaviano Morone, Byungjoon Min, Lin Bo, Romain Mari, and Hernán A Makse. Collective influence algorithm to find influencers via optimal percolation in massively large social media. *Scientific reports*, 6:30062, July 2016. ISSN 2045-2322. doi: https://10.1038/srep30062. URL http://europepmc.org/articles/PMC4960527.

Salomon Mugisha and Hai-Jun Zhou. Identifying optimal targets of network attack by belief propagation. *Phys. Rev. E*, 94:012305, Jul 2016. doi: 10.1103/PhysRevE.94.012305. URL https://link.aps.org/doi/10.1103/PhysRevE.94.012305.

Sai Munikoti, Kexing Lai, and Balasubramaniam Natarajan. Robustness assessment of hetero-functional graph theory based model of interdependent urban utility networks. *Reliability Engineering & System Safety*, 212:107627, 2021. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2021.107627. URL https://www.sciencedirect.com/science/article/pii/S095183202100168X.

Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

Mark EJ Newman. *Networks-An Introduction*. Oxford University Press, 2010.

Mark EJ Newman. Community detection and graph partitioning. *EPL (Europhysics Letters)*, 103(2):28003, 2013. URL http://stacks.iop.org/0295-5075/103/i=2/a=28003.

Shao-Meng Qin, Xiao-Long Ren, and Lin-Yuan Lü. Efficient network dismantling via node explosive percolation. *Communications in Theoretical Physics*, 71(6):764, jun 2019. doi: 10.1088/0253-6102/71/6/764. URL https://doi.org/10.1088/0253-6102/71/6/764.

Xiao-Long Ren, Niels Gleinig, Dirk Helbing, and Nino Antulov-Fantulin. Generalized network dismantling. *Proceedings of the National Academy of Sciences*, 116, 01 2018. doi: 10.1073/pnas.1806108116.

Bruno Requião da Cunha, Juan Carlos González-Avella, and Sebastián Gonçalves. Fast fragmentation of networks using module-based attacks. *PloS one*, 10(11):e0142824, 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0142824. URL http://europepmc.org/articles/PMC4646680.

Christian M Schneider, André A Moreira, José S Andrade, Shlomo Havlin, and Hans J Herrmann. Mitigation of malicious attacks on networks. *Proceedings of the National Academy of Sciences*, 108(10):3838–3841, 2011.

Xiaoqian Sun, Volker Gollnick, and Sebastian Wandelt. Robustness analysis metrics for worldwide airport network: A comprehensive study. *Chinese Journal of Aeronautics*, 30(2): 500–512, 2017. ISSN 1000-9361. doi: https://doi.org/10.1016/j.cja.2017.01.010. URL https://www.sciencedirect.com/science/article/pii/S1000936117300390.

Liang Tian, Amir Bashan, Da-Ning Shi, and Yang-Yu Liu. Articulation points in complex networks. *Nature communications*, 8: 14223, 2017.

Sebastian Wandelt, Xiaoqian Sun, Daozhong Feng, Massimiliano Zanin, and Shlomo Havlin. A comparative analysis of approaches to network-dismantling. *Scientific reports*, 8(1):1–15, 2018.

Sebastian Wandelt, Xing Shi, Xiaoqian Sun, and Massimiliano Zanin. Community detection boosts network dismantling on real-world networks. *IEEE Access*, PP:1–1, 06 2020. doi: 10.1109/ACCESS.2020.3002807.

Sebastian Wandelt, Xing Shi, and Xiaoqian Sun. Estimation and improvement of transportation network robustness by exploiting communities. *Reliability Engineering & System Safety*, 206: 107307, 2021. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2020.107307. URL `https://www.sciencedirect.com/science/article/pii/S0951832020308036`.

Zhixiao Wang, Ya Zhao, Jingke Xi, and Changjiang Du. Fast ranking influential nodes in complex networks using a k-shell iteration factor. *Physica A: Statistical Mechanics and its Applications*, 461:171 – 181, 2016. ISSN 0378-4371. doi: dx.doi.org/10.1016/j.physa.2016.05.048.

Gongyu Wu, Meiyan Li, and Zhaojun Steven Li. A gene importance based evolutionary algorithm (giea) for identifying critical nodes in cyber–physical power systems. *Reliability Engineering & System Safety*, 214:107760, 2021. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2021.107760. URL `https://www.sciencedirect.com/science/article/pii/S0951832021002891`.

Ramin Zahedi and Mohammad Khansari. A new immunization algorithm based on spectral properties for complex networks. *Journal of Statistical Mechanics Theory and Experiment*, October 2015. doi: 10.1109/IKT.2015.7288754.

Massimiliano Zanin and Fabrizio Lillo. Modelling the air transport with complex networks: A short review. *The European Physical Journal Special Topics*, 215(1):5–21, 2013.

Massimiliano Zanin, Xiaoqian Sun, and Sebastian Wandelt. Studying the topology of transportation systems through complex networks: Handle with care. *Journal of Advanced Transportation*, 2018:1–17, 08 2018. doi: 10.1155/2018/3156137.

Lenka Zdeborová, Pan Zhang, and Hai-Jun Zhou. Fast and simple decycling and dismantling of networks. *Scientific reports*, 6, 2016.

Jianhua Zhang, Ziqi Wang, Shuliang Wang, Wenchao Shao, Xun Zhao, and Weizhi Liu. Vulnerability assessments of weighted urban rail transit networks with integrated coupled map lattices. *Reliability Engineering & System Safety*, 214:107707, 2021. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2021.107707. URL `https://www.sciencedirect.com/science/article/pii/S0951832021002428`.

Hai-Jun Zhou. Spin glass approach to the feedback vertex set problem. *European Physical Journal B*, 86(11), 2013.