Reliability Engineering and System Safety 206 (2021) 107307

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Estimation and improvement of transportation network robustness by exploiting communities

Sebastian Wandelt [a], Xing Shi [b], Xiaoqian Sun [*,a]

[a] National Key Laboratory of CNS/ATM, School of Electronic and Information Engineering, Beihang University, Beijing, China
[b] China Aerospace Academy of Systems Science and Engineering, Beijing, China

ARTICLE INFO

Keywords:
Robustness
Complex networks
Transportation systems
Communities

ABSTRACT

Throughout the past years, researchers increasingly study the resilience of transportation systems through the lens of complex networks. This model simplification has helped to identify bottlenecks for all kinds of systems, e.g., subway, railway, and road networks. Nevertheless, for large networks, with ten thousand and more nodes, standard complex network-based robustness analysis methods do not scale up well. In this study, we propose to estimate and improve the robustness of transportation systems by exploiting the presence of communities in complex network representations. A community, by definition, is densely connected inside, but loosely connected to other components in the system. Accordingly, the community structure and the induced edges connecting communities can help to orchestrate a framework for better analysis and protection of our transportation systems. Experiments on twelve real-world transportation systems demonstrate the efficiency and scalability of our novel community-based framework.

1. Introduction

Transportation systems require high disruption tolerance to guarantee a continued service and prevent economic losses caused by failures or intentional attacks. Several recent studies model transportation systems as complex networks, where the infrastructure of the transportation system is mapped to nodes and edges. For instance, airports are represented as nodes and two airports are connected whenever there is a direct flight between them [12]. How to determine an optimal attack towards a network is an NP-hard problem, given its intrinsic combinatorics over sets of nodes. Accordingly, many heuristics have been proposed, in order to derive disruptive attacks [6,28,40,46,51]. These strategies, in general, compute a ranking of nodes, based on local or global topological properties; see [49] for a recent survey. For instance, higher-degree nodes are better connected and their failure leads to larger loss of connectivity. The best-known method to identify near-optimal node orders is based on betweenness centrality, measuring the fraction of shortest paths a node appears on. The computation of betweenness centrality has a high runtime complexity, despite existing approximation schemes [50]. Moreover, the centrality of nodes often significantly changes during an attack, which requires recomputation upon node failures. This leads to a worst-case time complexity of $O(N^2 * E)$, where N is the number of nodes and E is the number of edges in the network. On large networks with millions of nodes (e.g., the road network in NewYork [20]), the runtime for most existing methods is prohibitively long.

In this study, we leverage community structures for guiding the attack generation process. Communities are groups of nodes which are highly connected but with only few connections between the communities [30]. Recently, a strong connection between resilience and community structure was found [9]. Community detection has not been used for the analysis of transportation systems so far; this lack of usage is intriguing, given that geography and construction constraints often induce a high modularity on transportation networks. We propose novel network robustness analysis methods as follows: COMDis (Community Dismantling) and COMDisE (Community Dismantling Edges) are network attack strategies for transportation networks based on the community structure. Both scale up to large networks while leading to outstanding attack quality. COMDis/COMDisE dismantle the community graph by attacking the interconnection nodes/edges, respectively. Besides, we also devise a strategy for identifying the most vital inter-community edges by adding weights based on the sizes of communities in the community graph. Furthermore, we propose CHH (Community High-High-Degree) and CLL (Community

* Corresponding author.
E-mail address: qqian.sun@gmail.com (X. Sun).

https://doi.org/10.1016/j.ress.2020.107307
Received 24 February 2020; Received in revised form 22 August 2020; Accepted 7 November 2020
Available online 13 November 2020

Low-Low-Degree) for network strengthening, which aim to exploit the networks' weaknesses identified by COMDis/COMDisE, adding links preferably between communities. We select twelve transportation networks of different modes and sizes from distinct regions, covering various topology structures for evaluation. All novel methods significantly outperform state-of-the-art, providing excellent trade-offs between solution quality and computational resources required. Our study provides a new perspective on transportation network robustness with focusing on inter-community edges in a network. Notably, compared to [17], our goal is not the identification of networks' robustness against specific attack types, but rather the identification of critical nodes and edges in a specific network.

The remainder of this paper is structured as follows. Section 2 reviews the literature on complex network and transportation network robustness; while Section 3 introduces standard network attack and strengthening methods. In Section 4, we introduce our novel methodology for analyzing and improving network robustness by leveraging communities. Section 5 reports our experimental results on twelve real-world transportation networks. Section 6 concludes the findings in our study and discusses some directions for future work.

2. Literature review

Literature on network robustness can be distinguished into two major categories: topology-based and operational robustness. The former, based on network topology, investigates an abstraction of the network into (usually unweighted) nodes and links. We review studies from this category first; see [49] for a recent review. Classical studies aim to identify highly-influential nodes, which establish the core of the network. The importance of nodes is measured with so-called network metrics, which quantify the importance based on local or global measures. For instance, the degree of a node measures its local connectivity with regard to neighbors, while the betweenness of anode measures the global fraction of shortest paths a node is located on. Advanced studies aim to dismantle a network algorithmically, with the aim to find subsets of interesting nodes. Collective influence (CI) [28] measures the importance of a node by aggregating degrees over all neighbors within a ball of size k. CoreHD [51] decycles the network by removing the node with the highest degree in 2-core networks first, followed by tree-breaking. AP greedily removes the most destructive articulation point in each iteration [46]; an articulation point is a node whose removal causes disconnection of a network.

Furthermore, several studies analyze transportation networks from a more operational-like perspective, by taking into account flows/congestions/capacities; see [25] for a recent review on resilience of urban critical infrastructure networks, including water, drainage, gas, transportation, electric, and communication networks. Cats et al. [7] integrated capacity degradation results per link into network robustness assessment, with an application of the urban public transport network of Amsterdam. Ouyang et al. [35] introduced four mathematical models and their solution algorithms to identify the optimal robustness-based and resilience-based protection strategies for critical infrastructure systems. Ouyang et al. [34] proposed a network-based approach to model the vulnerability of complementary transportation systems, the railway and airline systems in China were used as an example. Hong et al. [16] proposed three types of accessibility metrics based on departure time and investigated the vulnerability of the integrated metro and high-speed rail system in China under single high-speed train station failure and two real severe weather events. Hong et al. [15] proposed a methodology to analyze the vulnerability of public transit systems from the perspective of residential communities, the bus and subway systems in Wuhan, China were used as case studies. The vulnerability of complementary public transit systems with the consideration of passengers' intermodal transfer distance preference was studied as well [14]. Kermanshah and Derrible [22] provided a geographical and multi-criteria vulnerability assessment method to quantify the impacts of extreme earthquakes on road networks, two US cities, Los Angeles and San Francisco, were used as case studies. Muriel-Villegas et al. [29] provided a framework to derive the connectivity reliability and vulnerability of inter-urban transportation systems under network disruptions, with Antioquia, Colombia as a case study.

3. Measuring transportation network robustness with complex networks

Complex networks serve as models for many real-world systems, where nodes represent components and edges describe the interaction among components. The robustness of such a network can be evaluated based on removing nodes iteratively [6,40,46,48]: Once a node is attacked, the node itself and all the edges that connect it to its neighbors are removed from the network. In air transportation, once an airport is closed down, the airport is disconnected from the remaining network, since no aircraft can depart from or land at the closed airport. Edge removal, on the other hand, is common in transportation networks as well. In rail-based transportation, the closure of one track can disconnect two stations (if there exists no redundant track/connection between them). Even in air transportation systems, bad en-route weather can induce edge failures. This study investigates the robustness of a transportation network in presence of both types of failures.

To measure the vulnerability of network under specific attack, we compute the robustness value [42]: Given an order of node attack on a network with N nodes, the robustness value of the network is defined as $R = \frac{1}{N}\sum_{t=1}^{N} S(t)$, where $S(t)$ is the relative GCC (giant connected component) size after attacking t nodes. The computation of R depends on the attack order. The goal is finding the minimum R, which resembles a worst-case attack to a network. Computing the minimum R, however, is an NP-hard problem. Therefore, there is a need to generate good attacks using heuristics.

3.1. Node attack

Below, we review eight standard node attack strategies. The first four strategies are based on node centrality and the other four strategies are specifically designed for the network dismantling problem, as proposed in recent literature.

1. **D**: Nodes are attacked with a decreasing degree, i.e., number of neighbors. The initial network's node degree is used.
2. **DI**: DI removes the node with the highest degree first as well, but the degree is recomputed in each iteration.
3. **B**: Betweenness centrality measures the fraction of all-pairs-shortest-paths going through a node. Nodes are attacked according to decreasing betweenness. The initial network's node betweenness is used.
4. **BI**: BI removes nodes with the highest betweenness as well, but the betweenness is recomputed in each iteration.
5. **Min-sum**: Min-sum algorithm [6] determines an attack by decycling and tree-breaking.
6. **CI**: Collective influence (CI) [28] measures the importance of a node by aggregating degrees over all neighbors within a ball of size k.
7. **CoreHD**: CoreHD [51] decycles the network by removing the node with the highest degree in 2-core networks first, followed by tree-breaking.
8. **AP**: An articulation point is a node whose removal causes disconnection of a network. AP greedily removes the most destructive articulation point in each iteration [46].

All eight methods above have been systematically evaluated in Wandelt et al. [49]. In general, BI provides the most destructive attack, but has a worst-case time complexity of $O(N^2 * E)$. Such computational costs prevent the application to large networks with millions of nodes.

AP runs in almost linear time, much faster than CI and CoreHD on large networks; AP was also shown to outperform existing methods in terms of dismantling quality [46]. Note that AP ends when there is no more articulation point in the network. In order to obtain complete attacks, AP falls back to degree-based attacks (D) when no more articulation points exist in the network (as proposed by the authors of [46]).

3.2. Edge attack

Below, we describe five standard edge attack strategies as present in the literature. We terminate the attack process when the network is broken down into components of subcritical sizes. We keep attacking edges until the relative size of GCC is smaller than 10% compared to the original network. Then, we compute the Q value (the fraction of edges that need to be removed to cut GCC into 10%) to evaluate robustness under edge attack.

1. **RE**: RE selects one edge randomly from the network and removes this edge. RE keeps removing edges at random until GCC $< 10\%$.
2. **HDE**: In transportation networks, the edges between two hubs are more critical than other edges. For each edge (i,j), we compute the maximum degree of i and j and remove edges in descending order of the values.
3. **HDEI**: An interactive version of HDE, which recomputes the maximum degree of two nodes at the end of each edge and removes the edge with the highest degree.
4. **EB**: The betweenness of an edge is the sum of the fraction of all-pairs shortest paths that pass through this edge. Edges are removed in descending order of betweenness.
5. **EBI**: An interactive version of EB, which updates the edge betweenness values after each edge removal.

EB and EBI are the most commonly used strategies in the literature, given their definition based on shortest path criticality. Nevertheless, the computational complexity of EB/EBI makes the application to larger networks infeasible.

3.3. Network strengthening

Methods for enhancing the robustness of complex networks add new edges between node pairs; the selection of node pairs depends on the underlying strategy. Most commonly, node pairs are selected according to their degree, mimicking network growth models [19]. Two frequently-used strategies in the literature are as follows:

1. **HH**: Select nodes n_1 and n_2 proportional to their degree (roulette wheel method) and add edge (n_1,n_2) to the network, if it does not exist yet.
2. **LL**: Select nodes n_1 and n_2 inversely proportional to their degree and add edge (n_1,n_2) to the network, if it does not exist yet.

The goal of these two strategies is similar, but the effect on the network is rather different. HH preferably makes existing hubs (highly-connected nodes) even stronger, while LL tends to connect nodes which are not well connected yet.

4. Community-based robustness framework

The goal of our novel robustness framework is to handle large networks up to millions of nodes within reasonable runtime. Communities provide an excellent foundation for such a robustness framework, given that almost-linear time algorithms are known to reach acceptable modularity scores (linear in the number of edges or alternatively $O(N\log^2(N))$); see [24]. Furthermore, the network dismantling problem has a highly similar goal as community detection: the identification of weak, loosely-connected parts in the network. If some of these weak parts are disrupted, the overall network would be severely affected.

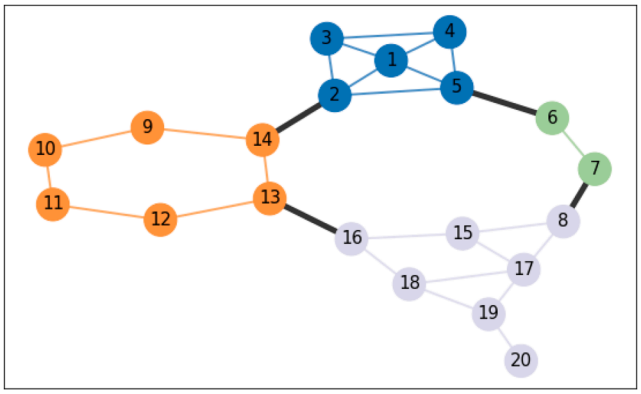

Fig. 1. Toy network with four communities detected by Louvain algorithm and represented by distinct colors. Inter-community edges are denoted in black color.

Communities are groups of highly-connected nodes, which have few connections between groups [30]. Fig. 1 shows four communities in a toy network: There is only one inter-community edge between each pair of connected communities; and within a community nodes are highly connected, e.g., the community with nodes 1, 2, 3, 4 and 5. Since transportation networks usually come with geographical restrictions, we focus on algorithms for disjoint communities in this study. Below, we introduce three classes of algorithms.

1. **Clustering algorithms**: Partitioning-based clustering methods divide the network into k clusters [2,10,27]. However, often the number of communities is not known in advance. In such cases hierarchical clustering methods are more suitable, merging similar clusters and identify multi-level communities [11,32].
2. **Dynamic algorithms**: Dynamic algorithms consider the processes of exploring the network [18]. Three typical approaches include random walk [36,52,53], synchronization [1,4], and spin models in statistical mechanics [38,39].
3. **Modularity optimization algorithms**: Modularity measures the quality of a division obtained by certain community detection method. Given a division of k communities, we obtain a $k \times k$ matrix in which element e_{ij} is the fraction of edges that connect community i and j. The modularity is $M = \sum_i e_{ii} - \sum_{ijk} e_{ij}e_{ki}$ [30]. Modularity optimization algorithms aim at maximizing modularity M by using heuristics, such as FastGN [31], Danon et al. [8] and Louvain algorithm [3]. Louvain algorithm [3] outperforms most other methods, with a linear time complexity regarding the number of edges, offering a nice trade-off between division quality and speed [24].

In general, a good network attack method requires to reduce the GCC-size quickly while removing a small fraction of nodes, leading to a small robustness value R. It is known that an attack based on betweenness centrality can get outstanding quality; given that high-betweenness nodes are exactly those nodes which lie on the majority of shortest paths. Community-based attack can be understood as a relaxation of the betweenness attack. Instead of finding the high-betweenness nodes directly, we assume that these nodes are in fact connecting different parts (=communities) of the networks, whose removal will break the weak components apart [9,47]. More generally, Stanley et al. [43] has shown that super nodes, i.e., groups of nodes contracted into one node, may be useful in a variety of domains dealing with large networks, allowing for otherwise difficult to perform analysis. In our context a super node could be considered as a community contracted into one node. Therefore, we design a novel attack to a network, by simply

Require: Network G, Community graph G_{com}
Ensure: The top edge to be attacked in G_{com}

```
Weights ← {}, Sizes ← {}
for ComNode ∈ G_com do
    Sizes[ComNode] ← the number of nodes in the Com
end for
for ComEdge ∈ G_com do
    Weights[ComEdge] ← 0
end for
APSP ← all pairs of the shortest paths in G_com
for s, sp in APSP do
    for t in sp do
        factor = Sizes[s]*Sizes[t]
        for e in sp[t] do
            Weights[e] ← Weights[e] + factor
        end for
    end for
end for
TopComEdge ← The top edge in Sorted(Weights)
return TopComEdge
```

Algorithm 1. IdentifyTopComEdge.

attacking the inter-community nodes in a specific order. We propose two scalable network dismantling/attacking methods, COMDis for node attack and COMDisE for edge attack, which both scale up to very large network instances. Both algorithms are based on community graph dismantling. Below we introduce the core parts of COMDis and COMDisE:

1. **Community detection**: When deciding how to attack the GCC, the community structure is determined first. For a scalable attack method, it is necessary to choose a community detection method which runs in linear time in the number of edges. More time-consuming community detection, specifically involving quadratic terms, would not scale up well. Given a network with E edges, the following three methods offer $O(E)$ time complexity: InfoMap [41], Label Propagation Algorithm (LPA) [37], and Louvain algorithm [3]. Since InfoMap and LPA detect overlapping communities, we selected Louvain algorithm for disjoint community detection in COMDis and COMDisE. After obtaining the community divisions, we create the community graph. In the community graph, all nodes inside a community are contracted into a single node. Intuitively, such a community graph makes the connections between communities explicit, while hiding the connections within communities. We create the community graph G_{com} in which each node (ComNode) is a community and each edge (ComEdge) indicates that there are connections between two communities.
2. **Community graph dismantling**: The community graph enables the identification of weak community connections, whose failures usually have severe effects on the network. We attack the network by dismantling the community graph in the following way. We separate two communities, i.e., cutting one edge in the created community graph in every iteration by removing the interconnection nodes (COMDis for node attack) or removing the interconnection edges directly (COMDisE for edge attack). For node attack, we need to decide which of the nodes to remove, in order to break the edge(s). This can be simply achieved by removing the node with the highest degree among the interconnection nodes iteratively; in our experiments this strategy led to best decisions in almost all cases. We keep attacking interconnection nodes on each edge in the community graph until all the edges in the community graph are removed. For edge attack, we directly remove these edges that connect two nodes with high degree; if multiple edges connect the two communities in the original network, all of them are removed.
3. **Top community edge identification**: The order of cutting edges in the community graph needs to be determined, which significantly affects the quality when there are many communities. To address this problem, we devised a strategy for identifying the most important edges in the community graph, which is shown in Algorithm 1. It should be noted that when attacking the community graph, we need to take into account that some of the nodes are more important than others, simply because a single node stands for a whole community of nodes in the original graph. Accordingly, we determine the weight of each ComEdge by considering both the number of nodes of each community in the original network (denoted by Sizes) and the number of shortest paths that pass through each community in the community graph. Intuitively, the edge in the community graph with the highest weight is the most important one and we attack it first.

Require: Connected Network G with N nodes and E edges
Ensure: Attack order

```
Attacks ← ∅
GCC ← the giant connected component of G
Coms ← communities detected by Louvain method
while |Coms| > 1 do
    G_com ← the community graph
    while G_com has edges do
        TopComEdge ← IdentifyTopComEdge(G,G_com)
        G_inter ← nodes and edges on TopComEdge
        while G_inter has no edge do
            v ← the highest degree node
            Remove v from G_inter and G
            Attacks ← Attacks ⋃ v
        end while
        Remove TopComEdge from G_com
    end while
end while
Remains ← remained nodes sorted by degree
Attacks ← Attacks ⋃ Remains
return Attacks
```

Algorithm 2. COMDis algorithm.

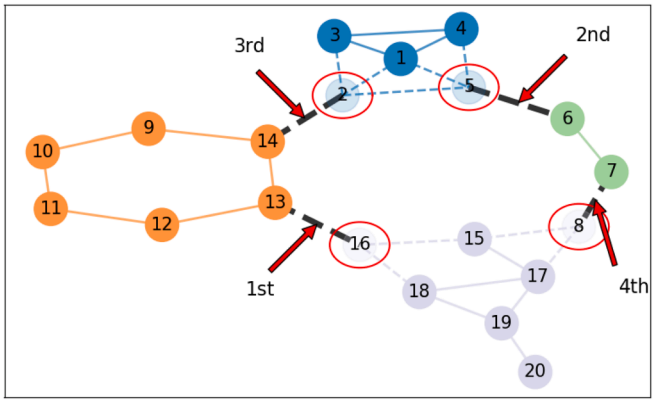

Fig. 2. Toy example network after removing nodes [16, 5, 2, 8] with COMDis. Arrow labels indicate the order of cutting edges in the community graph and the transparent nodes represent removed nodes (of COMDis for node attack).

Algorithm 2 presents our overall COMDis algorithm for attacking nodes based on community dismantling, using Algorithm 1 to determine

the order of attacking edge in the community graph. As long as the community graph still has edges, we identify the edge with the highest priority. For each of these edges we delete the highest-degree nodes, until the two communities are disconnected. Finally, once the community graph is completely dismantled, all remaining nodes are sorted by degree and appended to the attack, in order to ensure a full dismantling of the network. Please note that we have also implemented interactive variants of COMDis, which recompute the communities while attacking the network. These attempts, however, did not yield additional improvements. Our COMDisE algorithm only needs one modification with regard to COMDis: Instead of removing one node in Line 11 of Algorithm 2, COMDisE removes one edge whose two nodes have high degree.

Fig. 2 shows how COMDis works on the toy network example. Notably, COMDis obtains a robustness value ($R = 0.1925$) even smaller than BI ($R = 0.2125$). It can be seen that COMDis indeed cuts the vital interlink between two large communities in the community graph first and then proceeds to attack the interlink between the blue and green community instead of the interlinks attached to other large communities, which is more destructive to the whole network. This strategy results in a fast GCC reduction: The GCC is reduced to 50% after removing node 16 and node 5. For our COMDisE algorithm, Fig. 2 also shows the order of attacking edges, as indicated by red arrows.

Given an efficient community-based attacking strategy, the question arises how to strengthen a network against this kind of attack. Intuitively, the goal is to increase the number of inter-community connections, given that these additional edges would keep communities connected for a longer time under failures. Adding edges inside communities does not help here, since a community is already well-connected by definition. Traditional methods, i.e. HH and LL, do not consider the community structure; they add edges between nodes in arbitrarily chosen communities. This has a negative side-effect, specifically to transportation systems: rather distant nodes become connected by a single edge. As an example, in a subway network, HH/LL might suggest to connect the ends of a subway line; something that will very unlikely happen in practice (unless the line has a ring structure). In this study, we propose a novel edge-addition method which connects nodes between neighbor communities. Taking Fig. 2 as an example, there will be no edges between far-distance communities, we only strengthen the connections between directly-connected communities instead. The two algorithms HH and LL are extended as follows:

1. **CHH**: Select node n_1 proportional to the degree (roulette wheel method). Let community c_1 be the community of node n_1 and select community c_2 as a neighbor of c_1 in the community graph. Select node n_2 among the nodes in c_2 proportional to the degree and add edge (n_1, n_2) to the network, if it does not exist yet.
2. **CLL**: Select node n_1 inversely proportional to the degree. Let community c_1 be the community of node n_1 and select community c_2 as a neighbor of c_1 in the community graph. Select node n_2 among the nodes in c_2 inversely proportional to the degree and add edge (n_1, n_2) to the network, if it does not exist yet.

In summary, our framework makes a contribution to the literature as follows: To the best of our knowledge, this study is the first to include community structures in the analysis of transportation network robustness. Existing studies use topological measures which are either easy to compute, but inaccurate (e.g., D, DI) or accurate, yet computationally infeasible (e.g., BI). Recent studies [44,45,49] have found that there is a lack of methods which nicely address the sweet spot between accuracy and scalability. While betweenness approximation schemes can help to identify critical nodes, applying these schemes in an interactive fashion, i.e., recomputing node values after removal, cannot avoid quadratic runtime complexity [48,50]. In order to address this bottleneck, we leverage a community detection algorithm, which runs in linear time in the number of edges, to detect weakly connected subsystems. In terms of a provincial transportation network, such subsystems could be districts, cities, or other smaller components. All methods in this study build upon the fact that the links between the subsystems are the weak connections, whose failure inevitably leads to cascades, and thus, to considerable deterioration of system performance. The contributions of this study in light of the existing literature are summarized in Table 1. We propose scalable methods for the full network resilience analysis stack, containing node attacks (COMDis) and edge attacks (COMDisE). Furthermore, we exploit the identified weaknesses for two novel network strengthening methods (CHH and CLL), which aim to increase the connectivity between weakly connected subsystems. In addition to the methodological contribution, we discuss and evaluate the effectiveness and scalability of all methods on a wide range of twelve transportation networks with up to one million nodes; a scale unprecedented in the literature (see next section).

Table 1
Overview of network dismantling methods in this study (Our novel contribution is highlighted in bold).

Category	Method	Description	Time complexity
Node removal	D	Remove nodes by descending degree	$O(N)$
	DI	Interactive version of D	$O(N^2)$
	B	Remove node by descending betweenness	$O(NE)$
	BI	Interactive version of B	$O(N^2E)$
	AP	Attack the most destructive articulation point in each iteration	$O(E)$
	COMDis	Dismantle nodes based on community graph	$O(E)$
Edge removal	RE	Remove edges at random	$O(E)$
	HDE	Remove edges that connected high-degree nodes	$O(E)$
	HDEI	Interactive version of HDE	$O(E^2)$
	EB	Remove edges by descending edge betweenness	$O(E^2)$
	EBI	Interactive version of EB	$O(E^3)$
	COMDisE	Dismantle edges based on community graph	$O(E)$
Network strengthening	HH	Connect nodes by their degree	$O(E)$
	LL	Connect nodes by their degree (inversely)	$O(E)$
	CHH	Connect communities by node degree	$O(E)$
	CLL	Connect communities by node degree (inversely)	$O(E)$

5. Evaluation

We selected twelve real-world transportation networks with various topological structures, covering different geographical regions at three levels (i.e. cities, countries, and worldwide). We extracted the GCC of each network and removed duplicated edges between node pairs. Fig. 3 visualizes these twelve transportation networks; node locations are based on longitude/latitude for visualization purposes. Table 2 presents an overview on network properties, including the maximum degree (Maxdegree), density, transitivity, modularity, and the number of communities using Louvain algorithm (N_{coms}). It can be seen that the networks in our study cover a wide range of transportation modes, topologies, and various sizes. All experiments were conducted on a computer with four i7-6500U cores (2.50 GHz) and 16 GB RAM, running Ubuntu 16.04.4 LTS.

5.1. Robustness envelope

We analyze the efficiency and scalability of node attack strategies first. Fig. 4 shows the robustness curves of twelve transportation

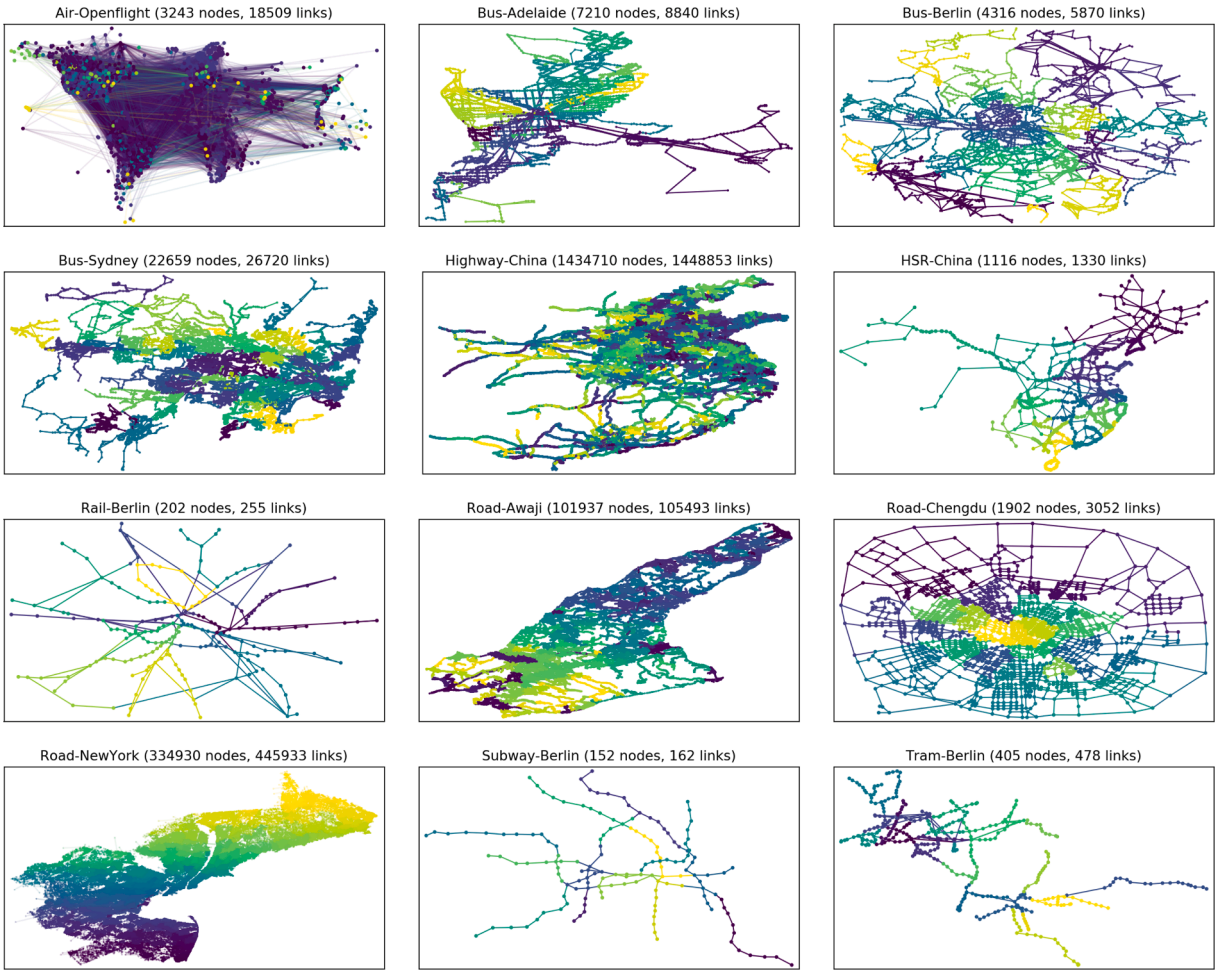

Fig. 3. Visualization of twelve transportation networks in this study. The color of nodes and edges represents the community; in case of inter-community edges, the color is chosen randomly among the communities of the two nodes.

Table 2
Overview on selected transportation networks.

Network	MaxDegree	Density	Transitivity	Modularity	N_{coms}
Air-Openflight [33]	243	0.0035	0.2474	0.6421	25
Bus-Adelaide [23]	15	0.0003	0.0425	0.9207	46
Bus-Berlin [23]	16	0.0006	0.1653	0.9347	44
Bus-Sydney [23]	13	0.0001	0.0444	0.9556	69
Highway-China	8	1e − 06	0.9976	0.0005	1044
HSR-China	9	0.0021	0.0632	0.8894	28
Rail-Berlin [23]	10	0.0126	0.0685	0.7553	14
Road-Awaji [21]	6	2e − 05	0.0001	0.9877	236
Road-Chengdu [13]	5	0.0017	0.0096	0.8828	23
Road-NewYork [20]	11	8e − 06	0.0520	0.9881	259
Subway-Berlin [23]	4	0.0141	0.0271	0.8062	13
Tram-Berlin [23]	8	0.0058	0.0959	0.8560	21

networks with different attack methods; the curves indicate how the relative GCC size decreases under attacks. On large networks (e.g., Highway-China), some methods do not finish computing in days, so we cannot report their results. BI provides the most effective attack on six networks among eight tested data sets. This shows the importance of (interactive) high betweenness for network connectivity. However, BI did not terminate on the four largest networks within 24 h, because it does not scale well. Moreover, the static method B gets the maximum R value on six networks but gets no results on Highway-China and Road-NewYork with millions of nodes. The interactive method DI is also much better than the static method D. AP gets the minimum R on Highway-China. For such a sparse network with many articulation points, the most destructive AP can reduce the connectivity of the network. Besides, AP also obtains good results on four networks with its R being ranked second. However, on Road-Chengdu, AP gets the maximum R value as Road-Chengdu has many cycles and few articulation points. Therefore, AP highly depends on the topology of the network and fails on networks without critical articulation points. COMDis offers the minimum R on five networks, the second smallest R on four networks and the third smallest R on the remaining three networks. In addition, COMDis outperforms BI on HSR-China and Road-Chengdu. We provide additional experimental results on the dismantling with Pagerank, eigenvector centrality, and random walk betweenness centrality in Fig. 10 in the Appendix A; while these methods work quite reasonably on smaller networks, either the quality deteriorates significantly or the runtime increase is not acceptable, when increasing the size of the network.

When considering edge removal instead of node removal, see Fig. 5, our COMDisE provides the most destructive attack. Please note that randomly attacking edges in a network can be (indirectly) considered as an informed method, since it preferably attacks the edges of high degree nodes in the network; the probability of hitting a node's edge just increases with the node degree. Therefore, we have included random edge

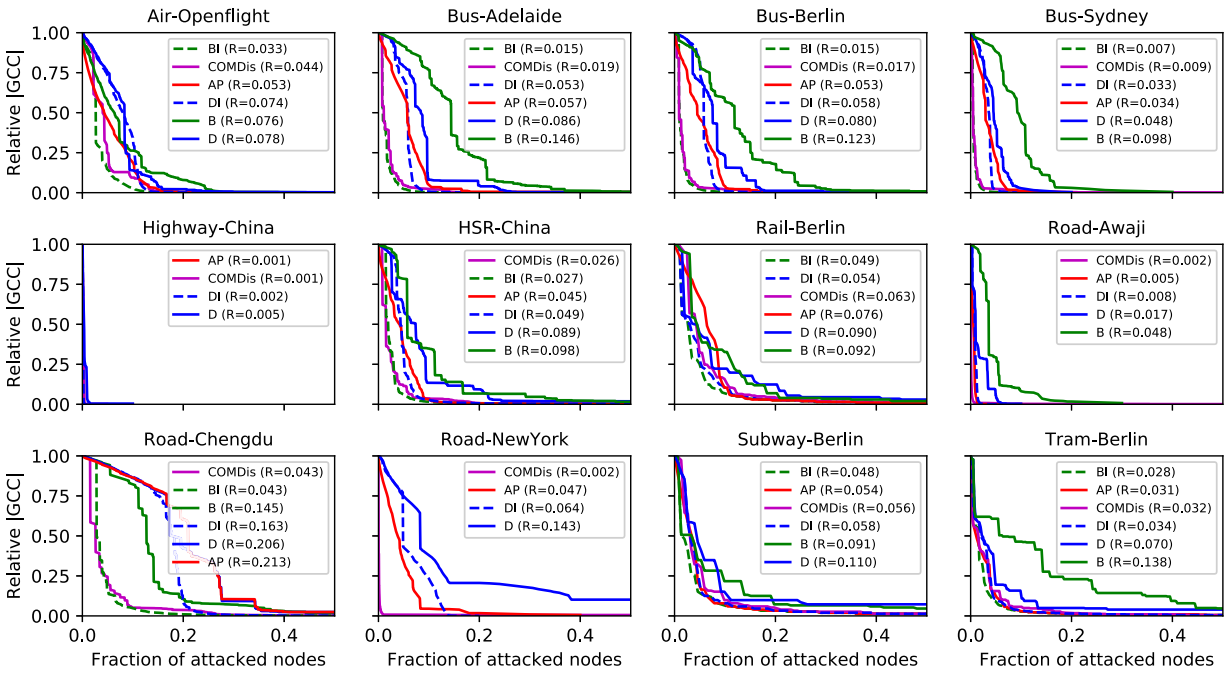

Fig. 4. Robustness envelopes under node attack. In each subplot, the *x*-axis represents the fraction of attacked nodes and the *y*-axis the effect on the system's performance. Curves closer to the lower-left origin of a plot indicate smaller *R* values and stronger effects in degrading a system's performance.

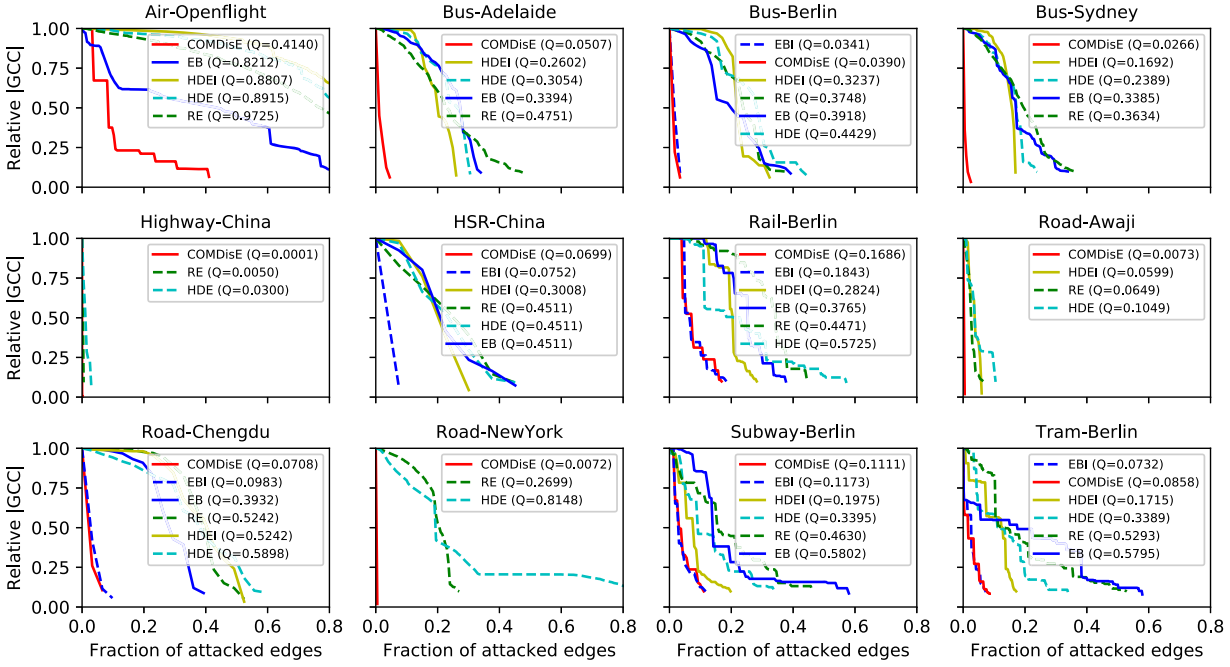

Fig. 5. Robustness envelopes under edge attack. In each subplot, the *x*-axis represents the fraction of attacked edges and the *y*-axis the effect on the system's performance. Curves closer to the lower-left origin of a plot indicate smaller *Q* values and stronger effects in degrading a system's performance.

removal in our experiments. For ten out of twelve data sets, COMDisE requires the minimum fraction of attacked edges to break GCC below 10%. On two small networks, Bus-Berlin and Tram-Berlin, COMDisE ranks second. On the largest network, COMDisE is still scalable and only takes 535 s to finish. On the six networks for which EBI obtains results, EBI also provides high-quality attacks. Similar to the results for node removal, interactive methods EBI and HDEI are superior to EB and HDE.

5.2. Tradeoffs between runtime and quality

Although BI provides the most destructive attack on most data sets, it does not finish computing within days on larger networks (e.g., Road-NewYork). Static degree (D), on the contrary, can quickly get results but the quality is often the worst. We evaluate the trade-off between runtime and quality further. For each transportation network, we min-normalized R and T of each method, i.e., the actual R (or runtime) are divided through the minimum R (or runtime). Accordingly, $R = 1$ means the best attack quality and $T = 1$ means fastest method. We plot all these

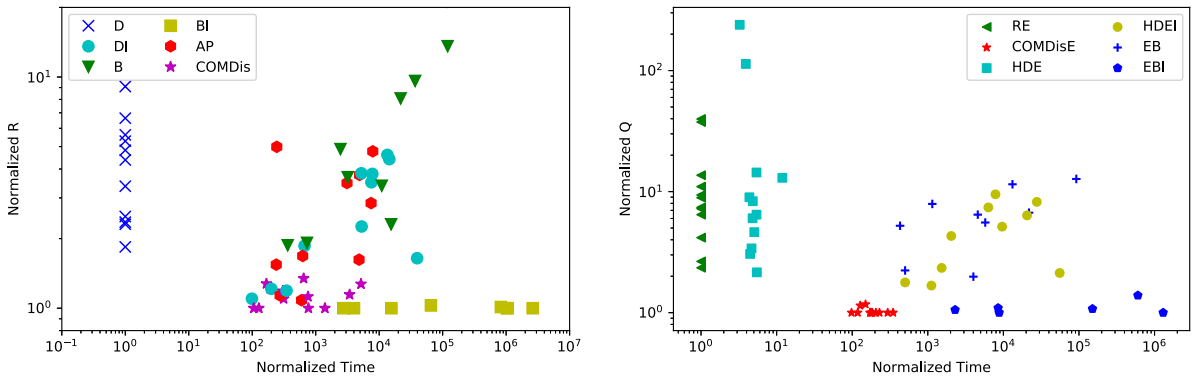

Fig. 6. Trade-offs between quality (obtained R value for node removal on the left and Q value for edge removal on the right) and computational resources (runtime) of different methods. The R values, Q values, and the runtime are min-normalized as fraction of R (or Q and time) against the minimum R (or Q and time) for the network. For all cases, smaller values are better.

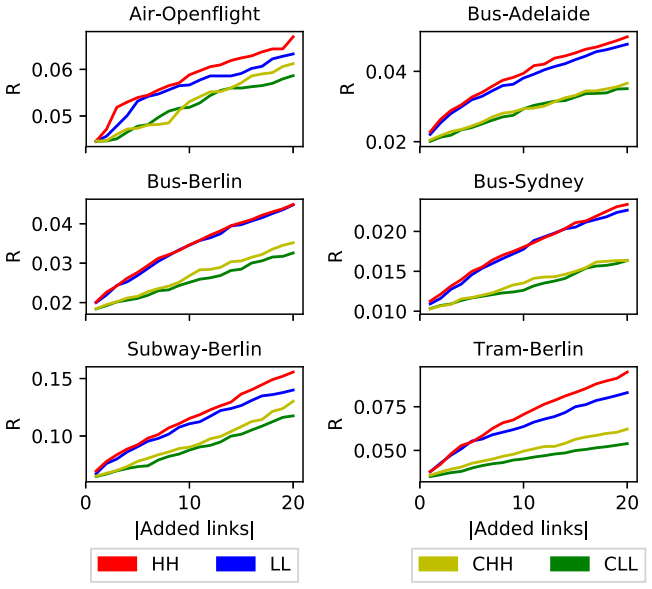

Fig. 7. The improvement of R regarding the number of added edges using four strengthening strategies on six transportation networks.

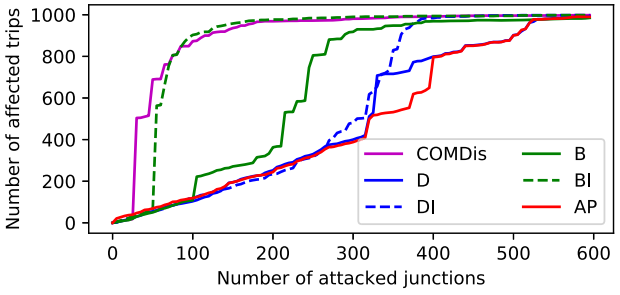

Fig. 8. Flow-based resilience of Road-Chengdu. The number of affected trips versus the number of attacked junctions.

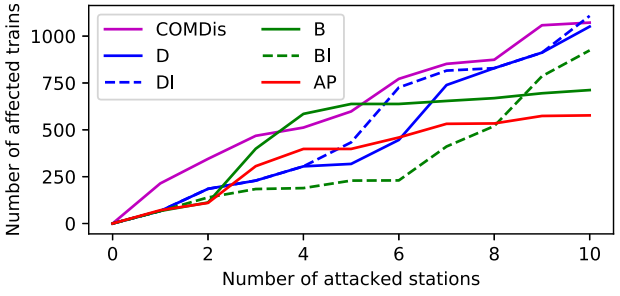

Fig. 9. Flow-based resilience of China-HSR. The number of affected trains versus the number of attacked stations.

normalized values of competitors on different networks in Fig. 6. We can see that method B based on static betweenness computation is slow with poor quality, as the time complexity is $O(N*E)$, where N is the number of nodes and E is the number of edges [5]. Besides, the attack effect of static method D is also bad. Compared to static methods, the interactive counterparts DI and BI are much slower but get better quality. Overall, these four methods based on node centrality cannot make good trade-offs. The (T,R) points of COMDis method are concentrated in the middle and lower parts in Fig. 6, which indicates that COMDis method offers nice trade-offs between runtime and quality, as our method is based on the (usually much smaller) community graph. In fact, our algorithm spends most of its time on community detection. AP is also fast but the quality is not as good as COMDis: The R value can be up to four times larger than the one obtained by COMDis. As for edge removal, COMDisE also provides a nice, scalable trade-off between quality and runtime. Although EBI can also get small Q, it does not scale up with the size of the network, given the high computational complexity: EBI does not finish the results on six larger networks within 24 h of computation.

5.3. Strengthening envelope

Fig. 7 presents the results for adding edges with different strengthening strategies. Given the intrinsic randomness in all strategies, we report the median R values after adding edges. The attacking strategy used for obtaining R is COMDis. LL often increases the robustness most, more than HH. The reason is that connecting additional hubs does not make these hubs more resilient to failures, but adding edges between nodes at the periphery of the network does. CHH/CLL usually obtain R values around half as high as HH/LL.

5.4. Robustness under flow

Transportation networks serve as the infrastructure for the transportation of people and goods. These flow characteristics are neglected in the study so far, given that complex networks are an abstraction to study mainly the topology. We have shown above that community-based resilience methods excel from the topological point of view. In order to analyze their performance under network flow, we perform two additional experiments. In the first experiment, we generated 1000 random trips over the network Road-Chengdu. A trip is affected, if there is no

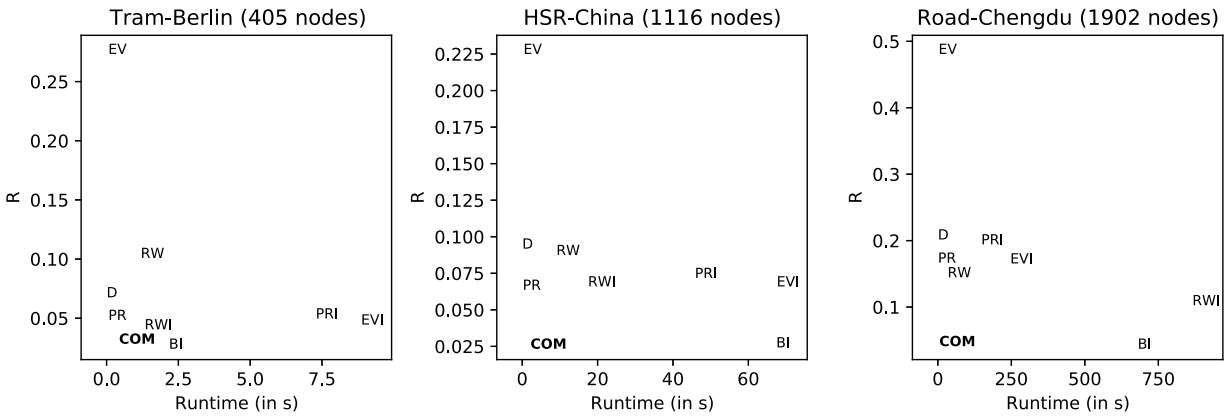

Fig. 10. Results on additional network metrics. Pagerank (PR: static, PRI: interactive), Eigenvector-centrality (EV: static, EVI: interactive), and Random-walk betweenness (RW: static, RWI: interactive); for comparison, we show degree (D), interactive betweenness (BI), and our novel method based on communities (COM). COM provides the best tradeoff between attack quality (smaller R is better) and fast runtime.

shortest road path from origin to destination of a trip. The results are shown in Fig. 8. COMDis is the best method, affecting the largest number of trips with a fewer number of junctions. Given the random distribution of trips, BI is strong as well. In the second experiment, we report the number of affected HSR trains in the network HSR-China, when using different node attack strategies. A train is affected, if its stations are not connected after an attack. The results are shown in Fig. 9. Again, COMDis is the best method, affecting the largest number of HSR trains with a small fraction of stations. The bad performance of BI can be explained by the fact that it often prefers very small, yet central stations, where the number of serving trains is lower.

6. Conclusions

Analyzing, understanding and improving the robustness of transportation networks is an important challenge in our society. In this study, we systematically evaluate the robustness of transportation networks by leveraging the community structure. We have proposed two effective and scalable network dismantling algorithms by separating communities, considering both node removal and edge removal on transportation networks. Our COMDis/COMDisE algorithms for attacking nodes/edges cut all interlinks in the community graph and remove interconnection nodes/edges on these community interlinks. We selected twelve transportation networks of various sizes, modes, and structures from different regions and compared COMDis/COMDisE against state-of-the-art methods. Our results show that COMDis/COMDisE obtain outstanding attack effects with a linear runtime regarding the number of edges in the network. For attacking nodes, BI offers good quality but the runtime is prohibitive for large networks. AP, on the other hand, highly depends on the number of articulation points in the network and often overestimates the robustness. For attacking edges, our COMDisE is outstanding and outperforms all other methods.

Our results provide a scalable method for network dismantling, shed light on the important roles of inter-community nodes/edges for robustness improvement and deepen our understanding of the modularity in transportation networks. This study is limited to single-modal transportation systems. In practice, however, multiple transportation modes interact with each other, competitively and cooperatively. Future studies investigating the robustness of such multi-layer systems, can possibly reveal new insights in overall transportation resilience under failures. Furthermore, it will be interesting to model robustness of transportation networks with more expressive formalism based on flows, taking into account capacities of nodes and edges; particularly in presence of real transportation passenger flow data. Our own preliminary experiments on randomly-generated flows indicate the potential of community-based methods in such a setup as well. Another interesting direction is to extend our community-based network strengthening algorithm by making it preserve the degree distribution of the network [26].

CRediT authorship contribution statement

Sebastian Wandelt: Conceptualization, Formal analysis, Resources, Writing - original draft, Supervision, Funding acquisition. **Xing Shi:** Software, Validation, Writing - original draft. **Xiaoqian Sun:** Conceptualization, Writing - original draft, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study is supported by the National Natural Science Foundation of China (Grants nos.61861136005, 61851110763).

Appendix A

References

[1] Arenas A, Díaz-Guilera A, Pérez-Vicente CJ. Synchronization reveals topological scales in complex networks. Phys Rev Lett 2006;96:114102. https://doi.org/10.1103/PhysRevLett.96.114102.
[2] Bezdek JC. Pattern recognition with fuzzy objetive function algorithms. Springer Science & Business Media.; 2013.
[3] Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. J Stat Mech 2008;2008(10):P10008. https://doi.org/10.1088/1742-5468/2008/10/p10008.
[4] Boccaletti S, Ivanchenko M, Latora V, Pluchino A, Rapisarda A. Detecting complex network modularity by dynamical clustering. Phys Rev E 2007;75:045102. https://doi.org/10.1103/PhysRevE.75.045102.
[5] Brandes U. A faster algorithm for betweenness centrality. J Math Sociol 2004;25. https://doi.org/10.1080/0022250X.2001.9990249.
[6] Braunstein A, Dall'Asta L, Semerjian G, Zdeborová L. Network dismantling. Proc Natl Acad Sci 2016;113(44):12368–73.
[7] Cats O, Koppenol G-J, Warnier M. Robustness assessment of link capacity reduction for complex networks: Application for public transport systems. Reliab Eng Syst Saf 2017;167:544–53. https://doi.org/10.1016/j.ress.2017.07.009.Special Section: Applications of Probabilistic Graphical Models in Dependability, Diagnosis and Prognosis
[8] Danon L, Díaz-Guilera A, Arenas A. Effect of size heterogeneity on community identification in complex networks. J Stat Mech 2006;2006(11):P11010. https://doi.org/10.1088/1742-5468/2006/11/p11010.

[9] Dong G, Fan J, Shekhtman LM, Shai S, Du R, Tian L, et al. Resilience of networks with community structure behaves as if under an external field. Proc Natl Acad Sci 2018;115(27):6911–5. https://doi.org/10.1073/pnas.1801588115.URL: https://www.pnas.org/content/115/27/6911

[10] Dunn JC. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. J Cybern 1973;3(3):32–57.

[11] Fortunato S. Community detection in graphs. Phys Rep 2010;486(3):75–174.

[12] Guimera R, Mossa S, Turtschi A, Amaral LN. The worldwide air transportation network: anomalous centrality, community structure, and cities' global roles. Proc Natl Acad Sci 2005;102(22):7794–9.

[13] Guo F., Zhang D., Dong Y., Guo Z.. Urban link travel speed dataset from a megacity road network2019;. URL: https://figshare.com/articles/Urban_link_travel_speed_dataset_from_a_megacity_road_network/7140209. 10.6084/m9.figshare.7140209.v4.

[14] Hong L, Yan Y, Ouyang M, Tian H, He X. Vulnerability effects of passengers' intermodal transfer distance preference and subway expansion on complementary urban public transportation systems. Reliab Eng Syst Saf 2017;158:58–72. https://doi.org/10.1016/j.ress.2016.10.001.Special Sections : Reliability and Safety Certification of Software-Intensive Systems, URL: http://www.sciencedirect.com/science/article/pii/S095183201630624X

[15] Hong L, Zhong X, Ouyang M, Tian H, He X. Vulnerability analysis of public transit systems from the perspective of urban residential communities. Reliab Eng Syst Saf 2019;189:143–56. https://doi.org/10.1016/j.ress.2019.04.018.URL: http://www.sciencedirect.com/science/article/pii/S0951832018301340

[16] Hong L, Ouyang M, Xu M, Hu P. Time-varied accessibility and vulnerability analysis of integrated metro and high-speed rail systems. Reliab Eng Syst Saf 2020;193:106622. https://doi.org/10.1016/j.ress.2019.106622.URL: http://www.sciencedirect.com/science/article/pii/S0951832019304582

[17] Iyer S, Killingback T, Sundaram B, Wang Z. Attack robustness and centrality of complex networks. PLoS One 2013;8:e59613. https://doi.org/10.1371/journal.pone.0059613.

[18] Javed MA, Younis MS, Latif S, Qadir J, Baig A. Community detection in networks: A multidisciplinary review. J Netw Comput Appl 2018;108:87–111. https://doi.org/10.1016/j.jnca.2018.02.011.URL: http://www.sciencedirect.com/science/article/pii/S1084804518300560

[19] Jiang Z, Liang M, Guo D. Enhancing network performance by edge addition. Int J Mod Phys C 2011;22(11):1211–26.

[20] Karduni A, Kermanshah A, Derrible S. A protocol to convert spatial polyline data to network formats and applications to world urban road networks. Sci Data 2016;3:160046. https://doi.org/10.1038/sdata.2016.46.

[21] Kawamata K, Oku K. Roadscape-based route recommender system using coarse-to-fine route search. J Inf Process 2019;27:392–403. https://doi.org/10.2197/ipsjjip.27.392.

[22] Kermanshah A, Derrible S. A geographical and multi-criteria vulnerability assessment of transportation networks against extreme earthquakes. Reliab Eng Syst Saf 2016;153:39–49. https://doi.org/10.1016/j.ress.2016.04.007.URL: http://www.sciencedirect.com/science/article/pii/S0951832016300217

[23] Kujala R, Weckström C, Darst RK, Mladenovic MN, Saramäki J. A collection of public transport network data sets for 25 cities. Sci Data 2018;5:180089.

[24] Lancichinetti A, Fortunato S. Community detection algorithms: a comparative analysis. Phys Rev E 2009;80:056117. https://doi.org/10.1103/PhysRevE.80.056117.

[25] Liu W, Song Z. Review of studies on the resilience of urban critical infrastructure networks. Reliab Eng Syst Saf 2020;193:106617. https://doi.org/10.1016/j.ress.2019.106617.URL: http://www.sciencedirect.com/science/article/pii/S0951832018309840

[26] Louzada V, Daolio F, Herrmann H, Tomassini M. Smart rewiring for network robustness. J Complex Netw 2013;1:150–9. https://doi.org/10.1093/comnet/cnt010.

[27] MacQueen J.. Some methods for classification and analysis of multivariate observations. vol. 1. 1967, p. 281–297.

[28] Morone F, Makse HA. Influence maximization in complex networks through optimal percolation. Nature 2015;524(7563):65–8. https://doi.org/10.1038/nature14604.

[29] Muriel-Villegas JE, Alvarez-Uribe KC, Patiño-Rodríguez CE, Villegas JG. Analysis of transportation networks subject to natural hazards - insights from a colombian case. Reliab Eng Syst Saf 2016;152:151–65. https://doi.org/10.1016/j.ress.2016.03.006.URL: http://www.sciencedirect.com/science/article/pii/S0951832016000727

[30] Newman MEJ. Detecting community structure in networks. Eur Phys J B 2004;38(2):321–30. https://doi.org/10.1140/epjb/e2004-00124-y.

[31] Newman MEJ. Fast algorithm for detecting community structure in networks. Phys Rev E 2004;69(6 Pt 2):066133.

[32] Newman MEJ, Girvan M. Finding and evaluating community structure in networks. Phys Rev E 2004;69:026113. https://doi.org/10.1103/PhysRevE.69.026113.

[33] Openflights. 2020. URL: https://openflights.org/data.html.

[34] Ouyang M, Pan Z, Hong L, He Y. Vulnerability analysis of complementary transportation systems with applications to railway and airline systems in china. Reliab Eng Syst Saf 2015;142:248–57. https://doi.org/10.1016/j.ress.2015.05.013.URL: http://www.sciencedirect.com/science/article/pii/S0951832015001623

[35] Ouyang M, Liu C, Xu M. Value of resilience-based solutions on critical infrastructure protection: Comparing with robustness-based solutions. Reliab Eng Syst Saf 2019;190:106506. https://doi.org/10.1016/j.ress.2019.106506.URL: http://www.sciencedirect.com/science/article/pii/S0951832018304927

[36] Pons P, Latapy M. Computing communities in large networks using random walks. Computer and information sciences - ISCIS 2005. Berlin, Heidelberg: Springer Berlin Heidelberg; 2005, ISBN 978-3-540-32085-2. p. 284–93.

[37] Raghavan UN, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 2007;76:036106. https://doi.org/10.1103/PhysRevE.76.036106.

[38] Reichardt J, Bornholdt S. Detecting fuzzy community structures in complex networks with a Potts model. Phys Rev Lett 2004;93(21):218701. https://doi.org/10.1103/PhysRevLett.93.218701.

[39] Reichardt J, Bornholdt S. Statistical mechanics of community detection. Phys Rev E 2006;74(1 Pt 2):016110. https://doi.org/10.1103/physreve.74.016110.

[40] Ren X-L, Gleinig N, Helbing D, Antulov-Fantulin N. Generalized network dismantling. Proc Natl Acad Sci 2019;116(14):6554–9. https://doi.org/10.1073/pnas.1806108116.URL: https://www.pnas.org/content/116/14/6554

[41] Rosvall M, Bergstrom CT. Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci 2008;105(4):1118–23. https://doi.org/10.1073/pnas.0706851105.

[42] Schneider CM, Moreira AA, Andrade JS, Havlin S, Herrmann HJ. Mitigation of malicious attacks on networks. Proc Natl Acad Sci 2011;108(10):3838–41.

[43] Stanley N, Kwitt R, Niethammer M, Mucha P. Compressing networks with super nodes. Sci Rep 2017;8. https://doi.org/10.1038/s41598-018-29174-3.

[44] Sun X, Gollnick V, Wandelt S. Robustness analysis metrics for worldwide airport network: A comprehensive study. Chinese Journal of Aeronautics 2017;30(2):500–12. https://doi.org/10.1016/j.cja.2017.01.010.URL: http://www.sciencedirect.com/science/article/pii/S1000936117300390

[45] Sun X, Wandelt S, Cao X. On node criticality in air transportation networks. Netw Spat Econ 2017;17(3):737–61. https://doi.org/10.1007/s11067-017-9342-5.

[46] Tian L, Bashan A, Shi D-N, Liu Y-Y. Articulation points in complex networks. Nat Commun 2017;8:14223. https://doi.org/10.1038/ncomms14223.

[47] Tulu MM, Hou R, Younas T. Identifying influential nodes based on community structure to speed up the dissemination of information in complex network. IEEE Access 2018;6:7390–401.

[48] Wandelt S, Sun X, Zanin M, Havlin S. QRE: quick robustness estimation for large complex networks. Future Gener Comput Syst 2017;83. https://doi.org/10.1016/j.future.2017.02.018.

[49] Wandelt S, Sun X, Feng D, Zanin M, Havlin S. A comparative analysis of approaches to network-dismantling. Sci Rep 2018;8(1):13513. https://doi.org/10.1038/s41598-018-31902-8.

[50] Wandelt S, Shi X, Sun X. Scalability of betweenness approximation algorithms: an experimental review. IEEE Access 2019;7:104057–71. https://doi.org/10.1109/ACCESS.2019.2927681.

[51] Zdeborová L, Zhang P, Zhou H-J. Fast and simple decycling and dismantling of networks. Scientific Reports 2016;6:37954. https://doi.org/10.1038/srep37954. Article

[52] Zhou H. Distance, dissimilarity index, and network community structure. Phys Rev E 2003;67:061901. https://doi.org/10.1103/PhysRevE.67.061901.

[53] Zhou H, Lipowsky R. Network brownian motion: a new method to measure vertex-vertex proximity and to identify communities and subcommunities. In: Bubak M, van Albada GD, Sloot PMA, Dongarra J, editors. Computational science - ICCS 2004. Berlin, Heidelberg: Springer Berlin Heidelberg; 2004. p. 1062–9.