

Max Flow Vulnerability of Undirected Planar Networks

Lorenzo Balzotti^{1,*}, Paolo G. Franciosa²

¹Department of Computer, Control, and Management Engineering “Antonio Ruberti”, Sapienza University of Rome, Via Ariosto 25, 00185, Rome, Italy.

²Department of Statistical Science, Sapienza University of Rome, p.le Aldo Moro 5, 00185 Rome, Italy.

Abstract

In the real world, planar networks commonly model utility distribution networks such as gas pipelines, water distribution, city’s electric grid, and transportation networks like railways and highways. Evaluating the vulnerability of these networks is crucial on several point of view: in a military context in order to prevent terrorist attack, or for maintenance to prevent wear and tear or natural disasters from inflicting extremely severe damage. This work aims to evaluate the vulnerability of undirected planar network studying the decrease of maximum st -flow when an edge or a vertex is removed, that is, we propose algorithms for computing the *max flow vitality* of edges and vertices.

Our algorithms are applied to real world scenarios, and we prove that high vitality values are well approximated in time close to the time currently required to compute st -max flow $O(n \log \log n)$. All our algorithms work in $O(n)$ space.

Keywords

planar networks, undirected networks, max flow, vitality, vulnerability

1. Introduction

The design and construction of networks are important topics in today’s world. Establishing the vulnerability of a network is crucial to prevent severe damage. There is no universal definition of vulnerability, as the same network can be assessed using different metrics, resulting in it being deemed vulnerable or robust depending on the metric used [2, 3].

In this paper, we focus on undirected planar networks and study vulnerability with respect to the maximum flow. It’s worth noting that planar networks find commonly application in urban science for representing, either directly or with high accuracy, various infrastructure networks [4], as water distribution networks [5, 6] and lots of streets patterns [7, 8]. The cities structure are the subject of many studies [9, 10, 11, 12] based on their planar aspects; see [13, 14] for a complete bibliography.

We focus our study on the *st-maximum flow vitality*, that is the st -max flow decrease when an edge or a vertex is removed, where s and t are two fixed vertices. A practical metric for assessing the overall vulnerability of a network may involve considering the count of


SEBD 2024: 32nd Symposium on Advanced Database Systems, June 23-26, 2024, Villasimius, Sardinia, Italy

* A preliminary version of this paper appeared in [1].

*Corresponding author.

✉ lorenzo.balzotti@uniroma1.it (L. Balzotti); paolo.franciosa@uniroma1.it (P. G. Franciosa)

ORCID 0000-0001-6191-9801 (L. Balzotti); 0000-0002-5464-4069 (P. G. Franciosa)

 © 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

edges/vertices exhibiting high vitality, that is, elements whose deletion implies serious damage to the st -maximum flow value. Thus, if all edges and vertices have low vitality, the network can be deemed robust. We refer to [15, 16, 17] for surveys on several kind of robustness and vulnerability problems discussed by an algorithmic point of view.

In networks algorithms, the effect of edges removal/damage on the max flow value has been studied since 1963, only a few years after the pioneering paper by Ford and Fulkerson [18] in 1956. Wollmer [19] presented an algorithm for determining the most vital edge in a railway network, and other studies about max flow interdiction were carried out on the planar Russian railway [20, 21] reported in Figure 1. According to Schrijver [22], in those year the study of vitality and interdiction on maximum flow problem arose from the US Air Force's desire to disrupt the transportation capabilities of their adversaries.

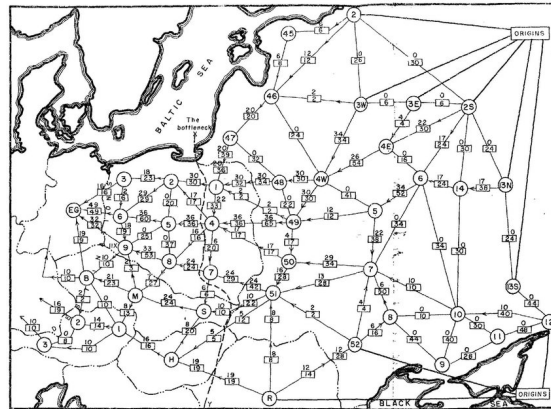


Figure 1: An illustrative real-world instance of a planar maximum flow problem is depicted in the schematic diagram of the railway network spanning the Western Soviet Union and Eastern European countries, as presented by Harris and Ross [20].

Ratliff and Sicilia [23] describe an algorithm for the k most vital edges, where the removal of these edge is simultaneous. Wood [24] showed that this problem is NP-hard in the strong sense, while its approximability has been studied in [25, 26]. Other interdiction problems studied in the literature include: shortest paths [27], minimum spanning tree [28, 29, 30], maximum matching [31, 32] and connectivity [33].

Previous work A recent survey on exact st -max flow algorithms can be found in [34]. The best know algorithms for general graphs solve the problem in $O(mn)$ time [35, 36]. For undirected planar graphs a better result in $O(n \log \log n)$ is presented by Italiano *et al.* [37], while for planar directed graphs Borradaile and Klein [38] propose an $O(n \log n)$ time algorithm. The easier st -planar case (that is when s and t are in the same face of a planar embedding) is reduced by Hassin [39] to the computation of a single source shortest paths; that can be solved in $O(n)$ time [40]. Finally, Eisenstat and Klein [41] illustrate how to compute the max flow in undirected unweighted planar graphs in optimal $O(n)$ time.

There are a few results focused on max flow vitality. In [42] the vitality of each edge and vertex is computed in optimal $O(n)$ time in the st -planar case. The dynamic max flow algorithms

presented in [37, 43] allow also to compute edge and vertex vitality for undirected planar graphs, we remind to Theorem 3 for a more in-depth discussion.

Our contribution We present fast algorithms for computing an additive guaranteed approximation of the st -max flow vitality for both edges and vertices with capacities below a chosen threshold C . In Section 4 we explain how to use our results in real world scenarios.

Let G be an undirected planar graph, let $E(G)$ and $V(G)$ be its set of edges and vertices, respectively, and let s and t be two fixed vertices. Let us denote by MF_G or MF if no confusion arises, its st -max flow. Let $c : E(G) \rightarrow \mathbb{R}^+$ be the *edge capacity function*, we define the capacity $c(v)$ of a vertex v as the sum of the capacities of all edges incident on v .

Formally, the st -max flow vitality of a set $S \subseteq (V(G) \setminus \{s, t\}) \cup E(G)$, denoted by $vit(S)$, corresponds to $MF_G - MF_{G-S}$, where $G - S$ is the graph derived from G by deleting set S . Our main results are summarized in the following two theorems. We show that we can compute an approximated value of both edge and vertex vitalities with an additive error lower than $\delta > 0$, where δ is an arbitrarily fixed vitality. All our algorithms work in $O(n)$ space.

Theorem 1. *Let G be a planar graph with positive edge capacities. Then for any $C, \delta > 0$, we can compute a value $vit^\delta(e) \in (vit(e) - \delta, vit(e)]$ for each $e \in E(G)$ satisfying $c(e) \leq C$, in $O(\frac{C}{\delta}n + n \log \log n)$ time.*

Theorem 2. *Let G be a planar graph with positive edge capacities. Then for any $C, \delta > 0$, we can compute a value $vit^\delta(v) \in (vit(v) - \delta, vit(v)]$ for each $v \in V(G)$ satisfying $c(v) \leq C$, in $O(\frac{C}{\delta}n + n \log n)$ time.*

In the full version of the paper [1], results in the cases of integer and bounded capacities and are presented. Those results are obtained thanks to a better algorithm for computing distances in these particular cases [44, 45]. In what follows, we focus only on edge vitality, and we refer to the full version for results and approach regarding vertex vitality.

Outline Our procedure is split into two parts. We start from the approach introduced by Itai and Shiloach's [46] for computing the st -max flow in undirected planar graphs, see Section 2. They translate the max flow computation into finding non-crossing shortest paths in graph D , which is derived from the dual graph of G . We use this construction to prove that edge vitalities can be inferred by knowing some distances in D , see Proposition 1. In the second part, which is exposed in Section 3, we compute required distances by introducing a divide-and-conquer strategy, which consists in slicing D via non-crossing shortest paths, see Lemma 1.

In Section 4 we apply our results to a real world scenario, and in Section 5 conclusions and open problems are presented.

2. From vitalities to distances

The first part of our approach is based on the work by Itai and Shiloach [46], that computes the st -maximum flow of G via st -separating cycle in its dual graph G^* . They build an auxiliary graph D_G (if no confusion arises, denoted as D) where the st -separating cycles correspond to

non-crossing shortest paths between pairs vertices on the external face of D . Starting from D , we explain how computing edge vitalities of G by finding some distances in D , this is our main result of this section and it is exposed in Proposition 1. Now we describe into details the construction of D by Itai and Shiloach [46].

We observe that every vertex in G^* corresponds to a face in G . In Figure 2 on the left, there is graph G in black continuous edges, and graph G^* in red dashed edges. In [46], the authors first fix a shortest path π from v_s^* to v_t^* , where v_s^* to v_t^* are the vertices in G^* corresponding to faces in G containing s and t , respectively. Without loss of generality, we assume that $\pi = \{v_1^*, v_2^*, \dots, v_k^*\}$, with $v_1^* = v_s^*$ and $v_k^* = v_t^*$. Path π is represented in green in Figure 2 on the left. Then they “cut” G^* along π obtaining graph D_G (for convenience we refer to it as graph D) represented in Figure 2 on the right.

They double path π into the paths π_x and π_y , the first has vertices $\{x_1, \dots, x_k\}$ and the latter $\{y_1, \dots, y_k\}$. In order to split the graph G^* into two, π has to split the whole plane into two parts. To obtain this, they add two dummy vertices α and β , and they connect them to v_s^* and v_t^* , respectively. If α is inside the face containing s and β inside the face containing t , then the edges αv_s^* and βv_t^* can be extended to the infinity, splitting the whole plane where G^* is embedded into two arbitrarily parts A and B ; without loss of generality we say that A is the part “above” π and B the part “below” π . Now we can explain how to build graph D .

For any $i \in [k]$, where $[k] = \{1, \dots, k\}$, edges in G^* incident on each v_i^* from below π are moved to y_i and edges incident on v_i^* from above π are moved to x_i . For each $e^* \in \pi$ in G^* , in D the copy of e^* in π_x is denoted by e_x^* , and the copy of e^* in π_y is denoted by e_y^* . Note that every vertex v in G generates a face f_v^* in G^* , that still remains a face in D , still denoted by f_v^* . If e^* in G^* does not belong to π , then we still denote the corresponding edge in D by e^* . Similarly, if V_i^* in G^* does not belong to π , then we still denote the corresponding vertex in D by v_i^* (this happens when $i > k$).

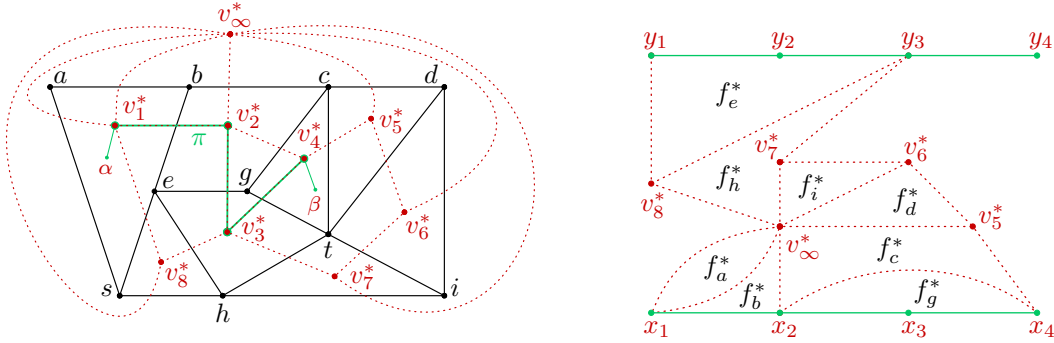


Figure 2: on the left graph G in black continuous line, G^* in red dashed lines, shortest path π from v_s^* (v_1^*) to v_t^* (v_k^*) in green, α and β are dummy vertices. On the right graph D .

For $i \in [k]$, we define $d_i = \text{dist}_D(x_i, y_i)$. Itai and Shiloach [46] proved that $MF = \min_{i \in [k]} d_i$, that is, the st -maximum flow is equal to the minimum st -separating cycle. We observe that the removal of an edge e in G corresponds to contracting vertices of e^* in G^* into one, and so some d_i 's may be shorten. For this reason, for $e^* \in V(D)$ and $i \in [k]$ we define $d_i(e^*) = \min\{d_i, \text{dist}_D(x_i, e^*) + \text{dist}_D(y_i, e^*)\}$. Note that $d_i(e^*)$ represents the distance in D from x_i to

y_i if all vertices of e^* are contracted into one. For $e \in E(G)$ we define MF_e as the st -maximum flow in graph $G - e$. It follows that $\text{vit}(e) = MF - MF_e$ and, trivially, $\text{vit}(e) > 0$ if and only if $MF_e < MF$. In the following proposition we translate the problem of computing edge vitality in G into computing distances in D .

Proposition 1. *For every edge e of G , if $e^* \notin \pi$ in G^* , then $MF_e = \min_{i \in [k]} \{d_i(e^*)\}$. If $e^* \in \pi$ in G^* , then $MF_e = \min_{i \in [k]} \{ \min\{d_i(e_x^*), d_i(e_y^*)\} \}$.*

3. Divide-and-conquer strategy

In this section we explain our divide-and-conquer strategy that leads to Lemma 1, which is the key to derive Theorem 1. We slice graph D along shortest non-crossing $x_i y_i$'s paths.

From now on, for $i \in [k]$ we fix a shortest $x_i y_i$ path p_i , and we consider that $\{p_i\}_{i \in [k]}$ is a set of pairwise single-touch non-crossing shortest paths, where two paths are *single-touch* if their intersection is still a path. All the p_i 's can be computed in $O(n \log \log n)$ time by the algorithm in [37], and their lengths in $O(n)$ time by the algorithm in [44].

Let $U = \bigcup_{i \in [k]} p_i$, see Figure 3(a). The single-touch property assures us that U is a forest. Given an ab path p and a bc path q , we define $p \circ q$ as the (possibly not simple) ac path obtained by the union of p and q . Each p_i 's splits D into two parts as shown in the following definition and in Figure 3(b).

Definition 1. *For every $i \in [k]$, we define Left_i as the subgraph of D bounded by the cycle $\pi_y[y_1, y_i] \circ p_i \circ \pi_x[x_i, x_1] \circ l$, where l is the leftmost $x_1 y_1$ path in D . Similarly, we define Right_i as the subgraph of D bounded by the cycle $\pi_y[y_i, y_k] \circ r \circ \pi_x[x_k, x_i] \circ p_i$, where r is the rightmost $x_k y_k$ path in D .*

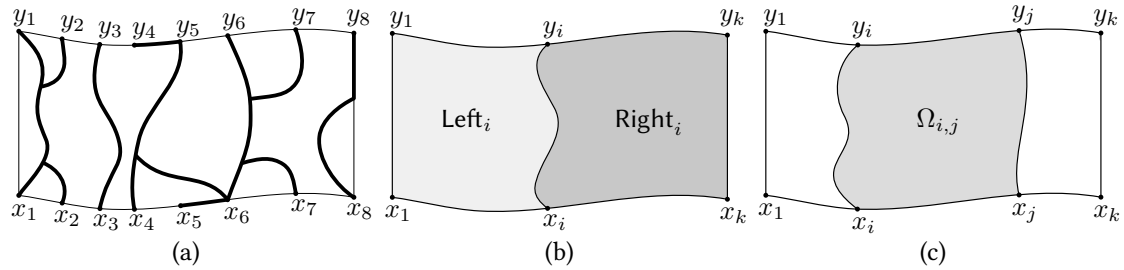


Figure 3: in (a) the graph U in bold and in (b) subgraphs Left_i and Right_i are highlighted. In (c) subgraph $\Omega_{i,j}$, for some $i < j$.

Based on Definition 1, for every $i, j \in [k]$, with $i < j$, we define $\Omega_{i,j} = \text{Right}_i \cap \text{Left}_j$, see Figure 3(c). Our divide-and-conquer strategy is based on these slices $\Omega_{i,j}$. Now we have to bucket the indices $\{1, 2, \dots, k\}$ so that these slices have some special distance properties.

For every $r \in \mathbb{N}$, we define $L_r = (\ell_1^r, \dots, \ell_{z_r}^r)$ as the ordered list of indices in $[k]$ such that $d_j \in [MF + \delta r, MF + \delta(r+1))$, for all $j \in L_r$, and $\ell_j^r < \ell_{j+1}^r$ for all $j \in [z_r - 1]$. If no confusion arises, we omit the superscript r ; thus we write ℓ_i in place of ℓ_i^r .

Now we can formulate properly our slicing strategy in the following lemma. Indeed, we note that every edge in D is always contained in a slice. The following lemma is the key of our slicing strategy. In particular, Lemma 1 can be applied for computing distances required in Proposition 1.

Lemma 1. *Let $r > 0$ and let $L_r = (\ell_1, \ell_2, \dots, \ell_z)$. Let e^* be a vertex in D and let $i \in [z - 1]$ satisfy $e^* \subseteq \Omega_{\ell_i, \ell_{i+1}}$. Then*

$$\min_{\ell \in L_r} d_\ell(e^*) > \min\{d_{\ell_i}(e^*), d_{\ell_{i+1}}(e^*)\} - \delta.$$

Moreover, if $e^ \subseteq \text{Left}_{\ell_1}$ (resp., $e^* \subseteq \text{Right}_{\ell_z}$) then $\min_{\ell \in L_r} d_\ell(e^*) > d_{\ell_1}(e^*) - \delta$ (resp., $\min_{\ell \in L_r} d_\ell(e^*) > d_{\ell_z}(e^*) - \delta$).*

An application of Lemma 1 is in Figure 4. By Proposition 1, we need $d_i(e^*)$ for all $i \in [k]$ and consider the minimum value. This can be obtained by $2k$ SSSP instances in D (one with source x_i and one with source y_i for every $i \in [k]$). But we can compute $\min_{\ell \in L_r} d_\ell(e^*)$ with only 4 SSSP instances focused on the slice of the graph D containing e^* ; in a way, we obtain the minimum among more indices by paying the additive error δ . In Figure 4, it holds $\min_{\ell \in L_r} d_\ell(e^*) \geq \min\{d_{\ell_3}(e^*), d_{\ell_4}(e^*)\} - \delta$.

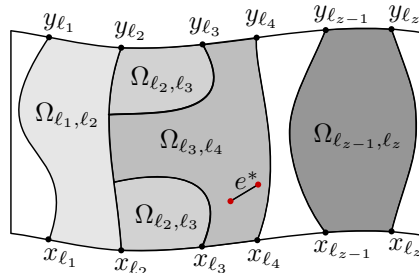


Figure 4: by Lemma 1, it holds that $\min_{\ell \in L_r} d_\ell(e^*) \geq \min\{d_{\ell_3}(e^*), d_{\ell_4}(e^*)\} - \delta$.

In the full version of this work [1] it is proved how to easily derive Theorem 1 from Lemma 1, moreover, the discussion is extended to vertex vitality, see Theorem 2.

4. Practical utilization in real world scenario

In this section we explain how to use the results stated in Theorem 1 and Theorem 2 to real world scenario. In particular we describe how to determine C and δ according to the distribution of edge capacities. Note that in real world applications one is usually interested in determining edge and/or vertices with high vitality, that is, edge or vertices whose deletion implies serious damage to the st -maximum flow value. This is strictly linked to evaluating the robustness/vulnerability of the network. Before arguing with our theorems, we report here the best algorithm for computing edge vitalities stated in [37, 43].

Theorem 3 ([37, 43]). *Let G be a planar graph with positive edge capacities. Then it is possible to compute the vitality of h single edges or the vitality of a set of h edges in $O(\min\{\frac{hn}{\log n} + n \log \log n, hn^{2/3} \log^{8/3} n + n \log n\})$ time.*

We underline that the previous algorithm can compute the exact vitality of h edges, but in order to obtain information about all edges, we need to apply it with $h = n$.

Returning to the discussion of our theorems, we observe that the capacities may be not bounded by any function of n . Despite this, we now introduce some procedures to make C/δ constant. In this way, the time complexity of Theorem 1 becomes $O(n \log \log n)$, that is the best current time bound for computing the st -max flow [37]. Let's start with the following crucial remark, where $c_{max} = \max_{e \in E(G)} c(e)$.

Remark 1. [Bounding capacities]. We can bound all edge capacities higher than MF to MF , obtaining a new bounded edge capacity function. This change has no impact on the st -max flow value or the vitality of any edge/vertex. Thus w.l.o.g., we can assume that $c_{max} \leq MF$.

After Remark 1, from now on we assume that $c_{max} \leq MF$. We apply the result in Theorem 1 to two real world scenarios. In the first scenario we assume that the distribution of capacities is general, i.e., there is no function that relates a capacity value to the number of edges having that capacity value. In the second scenario we study what happens when the distribution of edge capacities can be described by a power-law; but actually, this can be applied in every case there are few edges (outlier) with high capacity. Take into account that the power-law distribution is very common in real world planar networks, especially in distribution networks.

In Figure 5 we outline the following discussion. The figure shows how to determine constants C and δ in two different scenarios (the edge capacities are synthetic).

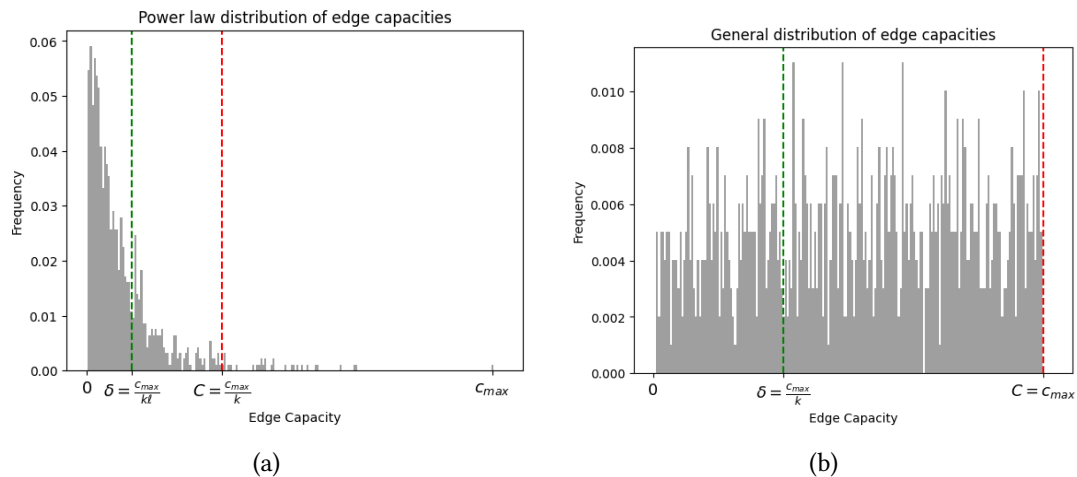


Figure 5: how to determine optimal C and δ according to the distribution of edge capacities.

- *General distribution.* In this case we can set $C = c_{max}$ and $\delta = C/k$, for some small constant k , e.g., $k = 10, 50, 100$. So we have a good approximation of edges with high vitality. Indeed, if an edge has vitality comparable with MF , then we obtain its vitality with an additive error smaller than MF/k (because of Remark 1). On the other hand, we have a bad approximation for edges with low capacity; in Figure 5(a) they correspond to edges on the left of the green line. This is not a problem because “small capacity” corresponds to “small vitality”, indeed the vitality

of an edge is at most its capacity. With this settings the time complexity stated in Theorem 1 becomes $O(n \log \log n)$, that is the time currently required for the computation of the st -max flow.

• *Power-law distribution.* We cannot apply the same reasoning to the previous case, in fact if we choose $C = c_{max}$ and $\delta = C/k$ for small constant k , then we obtain a bad approximation of all edges on the left of the red line in Figure 5(b). They are absolutely the majority of edges. So we set $C = c_{max}/k$ and $\delta = C/\ell$, for some constant ℓ , and we compute the (exact) vitality of edges whose capacity is greater than C via Theorem 3. Note that this set of edges is small because of the power law distribution of capacities. The vitality of edges computed by our algorithm has an additive error smaller than $\frac{MF}{k\ell}$, because of Remark 1; it may be an acceptable error even for small values of k and ℓ . Also in this case, the overall time complexity is equal or close to $O(n \log \log n)$.

Since our algorithms provide approximate vitalities with an additive error, it is possible having as output $vit^\delta(e) = 0$ for each $e \in E(G)$. At first sight, it may appear as an useless result, but actually we verify the robustness of the network because all edges have vitality smaller than δ . The same reasoning can be applied also to vertex vitality, and consequently to Theorem 2 by observing that in a real-world planar network there are usually a few vertices with high degree (as also implied by Euler's formula). We conclude by a fast comparison among our results and the algorithm stated in Theorem 3. It's clear that we compute approximated vitalities while Theorem 3 allows to compute exact vitality, but if one is interested in understanding the robustness/vulnerability of the network, then it is necessary to evaluate the vitality (exact or approximated) of all edges and/or all vertices. Our algorithms obtain this in a time equal or close to $O(n \log \log n)$, while we need $O(n^{5/3} \log^{8/2} n)$ time by using the result in Theorem 3. Hence our results provide considerably faster algorithms.

5. Conclusions and open problems

We describe fast algorithms for computing an additive guaranteed approximation of the st -max flow vitality of both edges and vertices in undirected planar networks. The vulnerability of real world planar networks can be established via our algorithms under various edge capacity distributions.

We left open the problem of computing the exact vitality of all edges and vertices in the same time required for computing the max flow value, as is already known for the st -planar case.

References

- [1] L. Balzotti, P. G. Franciosa, How vulnerable is an undirected planar graph with respect to max flow, *Networks* 83 (2024) 570–586. doi:<https://doi.org/10.1002/net.22205>.
- [2] S. Freitas, D. Yang, S. Kumar, H. Tong, D. H. Chau, Graph vulnerability and robustness: A survey, *IEEE Transactions on Knowledge and Data Engineering* 35 (2023) 5915–5934. doi:[10.1109/TKDE.2022.3163672](https://doi.org/10.1109/TKDE.2022.3163672).

- [3] H. V. Nath, Vulnerability assessment methods – a review, in: D. C. Wyld, M. Wozniak, N. Chaki, N. Meghanathan, D. Nagamalai (Eds.), *Advances in Network Security and Applications*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 1–10. doi:10.1007/978-3-642-22540-6_1.
- [4] M. Barthélemy, Spatial networks, *Physics Reports* 499 (2011) 1–101. doi:https://doi.org/10.1016/j.physrep.2010.11.002.
- [5] V. Chauhan, *Planar Graph Generation with Application to Water Distribution Networks*, Ph.D. thesis, Clemson University, 2018.
- [6] M. Herrera, E. Abraham, I. Stoianov, A Graph-Theoretic Framework for Assessing the Resilience of Sectorised Water Distribution Networks, *Water Resources Management* 30 (2016) 1685–1699. doi:10.1007/s11269-016-1245-6.
- [7] S. Marshall, *Streets and Patterns*, Routledge, 2004. doi:10.4324/9780203589397.
- [8] M. Southworth, E. Ben-Joseph, *Streets and the Shaping of Towns and Cities*, Island Press, 2013.
- [9] M. Barthélemy, A. Flammini, Modeling Urban Street Patterns, *Phys. Rev. Lett.* 100 (2008) 138702. doi:10.1103/PhysRevLett.100.138702.
- [10] J. Buhl, J. Gautrais, N. Reeves, R. V. Solé, S. Valverde, P. Kuntz, G. Theraulaz, Topological patterns in street networks of self-organized urban settlements, *The European Physical Journal B-Condensed Matter and Complex Systems* 49 (2006) 513–522. doi:10.1140/epjb/e2006-00085-1.
- [11] A. Cardillo, S. Scellato, V. Latora, S. Porta, Structural properties of planar graphs of urban street patterns, *Phys. Rev. E* 73 (2006) 066107. doi:10.1103/PhysRevE.73.066107.
- [12] A. P. Masucci, D. Smith, A. Crooks, M. Batty, Random planar graphs and the London street network, *The European Physical Journal B* 71 (2009) 259–271. doi:10.1140/epjb/e2009-00290-4.
- [13] G. Boeing, A multi-scale analysis of 27,000 urban street networks: Every us city, town, urbanized area, and zillow neighborhood, *Environment and Planning B: Urban Analytics and City Science* 47 (2020) 590–608. doi:10.1177/2399808318784595.
- [14] M. P. Viana, E. Strano, P. Bordin, M. Barthelemy, The simplicity of planar networks, *Scientific reports* 3 (2013) 3495. doi:10.1038/srep03495.
- [15] D. L. Alderson, G. G. Brown, W. M. Carlyle, L. A. Cox Jr, Sometimes There Is No “Most-Vital” Arc: Assessing and Improving the Operational Resilience of Systems, *Military Operations Research* 18 (2013) 21–37.
- [16] L.-G. Mattsson, E. Jenelius, Vulnerability and resilience of transport systems – a discussion of recent research, *Transportation Research Part A: Policy and Practice* 81 (2015) 16–34. doi:https://doi.org/10.1016/j.tra.2015.06.002, resilience of Networks.
- [17] A. T. Murray, An overview of network vulnerability modeling approaches, *GeoJournal* 78 (2013) 209–221. doi:http://dx.doi.org/10.1007/s10708-011-9412-z.
- [18] L. R. Ford, D. R. Fulkerson, Maximal Flow Through a Network, *Canadian journal of Mathematics* 8 (1956) 399–404. doi:http://dx.doi.org/10.1007/978-0-8176-4842-8_15.
- [19] R. D. Wollmer, *Some Methods for Determining the Most Vital Link in a Railway Network*, Rand Corporation, 1963.
- [20] T. Harris, F. Ross, *Fundamentals of a Method for Evaluating Rail Net Capacities*, Technical

Report, Rand Corporation, 1955.

- [21] A. W. McMasters, T. M. Mustin, Optimal Interdiction of a Supply Network, *Naval Research Logistics Quarterly* 17 (1970) 261–268. doi:<http://dx.doi.org/10.1002/nav.3800170302>.
- [22] A. Schrijver, On the history of the transportation and maximum flow problems, *Mathematical programming* 91 (2002) 437–445. doi:<http://dx.doi.org/10.1007/s101070100259>.
- [23] H. D. Ratliff, G. T. Sicilia, S. Lubore, Finding the n Most Vital Links in Flow Networks, *Management Science* 21 (1975) 531–539. doi:<http://dx.doi.org/10.1287/mnsc.21.5.531>.
- [24] R. K. Wood, Deterministic Network Interdiction, *Mathematical and Computer Modelling* 17 (1993) 1–18. doi:[http://dx.doi.org/10.1016/0895-7177\(93\)90236-R](http://dx.doi.org/10.1016/0895-7177(93)90236-R).
- [25] D. S. Altner, Ö. Ergun, N. A. Uhan, The Maximum Flow Network Interdiction Problem: Valid Inequalities, Integrality Gaps, and Approximability, *Operations Research Letters* 38 (2010) 33–38. doi:[10.1016/j.orl.2009.09.013](https://doi.org/10.1016/j.orl.2009.09.013).
- [26] C. A. Phillips, The Network Inhibition Problem, in: *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, ACM, 1993, pp. 776–785. doi:[10.1145/167088.167286](https://doi.org/10.1145/167088.167286).
- [27] E. Israeli, R. K. Wood, Shortest-path network interdiction, *Networks* 40 (2002) 97–111. doi:<https://doi.org/10.1002/net.10039>.
- [28] L.-H. Hsu, R.-H. Jan, Y.-C. Lee, C.-N. Hung, M.-S. Chern, Finding the most vital edge with respect to minimum spanning tree in weighted graphs, *Information Processing Letters* 39 (1991) 277–281. doi:[https://doi.org/10.1016/0020-0190\(91\)90028-G](https://doi.org/10.1016/0020-0190(91)90028-G).
- [29] L.-H. Hsu, P.-F. Wang, C.-T. Wu, Parallel algorithms for finding the most vital edge with respect to minimum spanning tree, *Parallel Computing* 18 (1992) 1143–1155. doi:[https://doi.org/10.1016/0167-8191\(92\)90061-B](https://doi.org/10.1016/0167-8191(92)90061-B).
- [30] S. Banerjee, S. Saxena, Parallel algorithm for finding the most vital edge in weighted graphs, *Journal of Parallel and Distributed Computing* 46 (1997) 101–104. doi:<https://doi.org/10.1006/jpdc.1997.1383>.
- [31] R. Zenklusen, Matching interdiction, *Discrete Applied Mathematics* 158 (2010) 1676–1690. doi:<https://doi.org/10.1016/j.dam.2010.06.006>.
- [32] M. Dinitz, A. Gupta, Packing interdiction and partial covering problems, in: M. Goemans, J. Correa (Eds.), *Integer Programming and Combinatorial Optimization*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 157–168. doi:http://dx.doi.org/10.1007/978-3-642-36694-9_14.
- [33] R. Zenklusen, Connectivity interdiction, *Operations Research Letters* 42 (2014) 450–454. doi:<https://doi.org/10.1016/j.orl.2014.07.010>.
- [34] O. Cruz-Mejía, A. N. Letchford, A survey on exact algorithms for the maximum flow and minimum-cost flow problems, *Networks* 82 (2023) 167–176. doi:<https://doi.org/10.1002/net.22169>.
- [35] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, *Network Flows* (1988).
- [36] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, M. Reddy, Applications of Network Optimization, *Handbooks in Operations Research and Management Science* 7 (1995) 1–83.
- [37] G. F. Italiano, Y. Nussbaum, P. Sankowski, C. Wulff-Nilsen, Improved Algorithms for

- Min Cut and Max Flow in Undirected Planar Graphs, in: Proceedings of the 43rd ACM Symposium on Theory of Computing, ACM, 2011, pp. 313–322. doi:10.1145/1993636.1993679.
- [38] G. Borradaile, P. N. Klein, An $O(n \log n)$ Algorithm for Maximum st -Flow in a Directed Planar Graph, *Journal of the ACM* 56 (2009) 9:1–9:30. doi:10.1145/1502793.1502798.
- [39] R. Hassin, Maximum Flow in (s,t) Planar Networks, *Information Processing Letters* 13 (1981) 107. doi:10.1016/0020-0190(81)90120-4.
- [40] M. R. Henzinger, P. N. Klein, S. Rao, S. Subramanian, Faster Shortest-Path Algorithms for Planar Graphs, *Journal of Computer and System Sciences* 55 (1997) 3–23. doi:10.1006/jcss.1997.1493.
- [41] D. Eisenstat, P. N. Klein, Linear-Time Algorithms for Max Flow and Multiple-Source Shortest Paths in Unit-Weight Planar Graphs, in: Symposium on Theory of Computing Conference, STOC'13, ACM, 2013, pp. 735–744. doi:10.1145/2488608.2488702.
- [42] G. Ausiello, P. G. Franciosa, I. Lari, A. Ribichini, Max flow vitality in general and st -planar graphs, *Networks* 74 (2019) 70–78. doi:10.1002/net.21878.
- [43] J. Łącki, P. Sankowski, Min-Cuts and Shortest Cycles in Planar Graphs in $O(n \log \log n)$ Time, in: Algorithms - ESA 2011 - 19th Annual European Symposium, Proceedings, volume 6942 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 155–166. doi:10.1007/978-3-642-23719-5_14.
- [44] L. Balzotti, P. G. Franciosa, Non-Crossing Shortest Paths in Undirected Unweighted Planar Graphs in Linear Time, *Journal of Graph Algorithms and Applications* 26 (2022) 589–606. doi:10.7155/jgaa.00610.
- [45] L. Kowalik, M. Kurowski, Short Path Queries in Planar Graphs in Constant Time, in: Proceedings of the 35th Annual ACM Symposium on Theory of Computing, ACM, 2003, pp. 143–148. doi:10.1145/780542.780565.
- [46] A. Itai, Y. Shiloach, Maximum Flow in Planar Networks, *SIAM Journal on Computing* 8 (1979) 135–150. doi:10.1137/0208012.