מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Thesis for the degree
Doctor of Philosophy

עבודת גמר (תזה) לתואר
דוקטור לפילוסופיה

Submitted to the Scientific Council of the
Weizmann Institute of Science
Rehovot, Israel

מוגשת למועצה המדעית של
מכון ויצמן למדע
רחובות, ישראל

By
Havana Rika

מאת
הוואנה ריקה

כיסוי טרמינלים על-ידי פאות בגרפים מישוריים: דילול, שיכון ועוד

Terminal Face Cover in Planar Graphs: Sparsifiers, Embeddings and More

Advisor:
Prof. Robert Krauthgamer

מנחה:
פרופ' רוברט קראוטגמר

May 2020

סיוון התש"פ

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

Thesis for the degree Doctor of Philosophy

Terminal Face Cover in Planar Graphs: Sparsifiers, Embeddings and More

Havana Rika

under the supervision of

Prof. Robert Krauthgamer

May 31, 2020

Those who know, do.
Those that understand, teach.

Aristotle

Acknowledgements

I would like to express my deepest gratitude to Robi, for his uncompromised and caring guidance, his endless patience, and his positive attitude and support. I would also like to thank the wonderful faculty members and administrative staff, who made Weizmann feel like my second home. And finally, to my friends and family, I couldn't go without you all, thank you!

Declaration

I declare that the thesis summarizes my own research. Parts of the research were performed in collaboration with other researchers, and have already been published in refereed journals and conferences, as described below. The results presented in Section 2 are joint work with Robert Krauthgamer, and were published in SIAM Journal on Discrete Mathematics [KR20]. The results presented in Section 3 are joint work with Robert Krauthgamer and James R. Lee, and were published in proceeding of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA) [KLR19]. The results presented in Section 4 are joint work with Lee-Ad Gottlieb and Robert Krauthgamer, and can be found on the arXiv [GKR20].

Abstract

The important family of planar graphs has been studied extensively, and the planarity properties are often leveraged algorithmically to achieve better bounds such as running time or accuracy. Usually these bounds are expressed as a function of the number of vertices or the edges of the graph, but they can be refined by the number of "relevant" faces. More formally, given a planar graph G with k vertices designated as terminals, let γ be the minimum number of faces that are incident to all the terminals. The parameter γ represents the topology of the terminals in G, and it has a long history in the study of cuts and (multicommodity) flows, shortest paths, and the Steiner tree problem. Our work presents new results for graph sparsification, graph embeddings and the flow-cut gap with respect to the parameter γ.

In the (vertex) cut-sparsification problem, we are given a graph G with k terminals, and we wish to compress it into a graph H (of small size but with the same terminals) which maintains the value of the minimum cut between every bipartition of the k terminals. We improve a known upper bound on the size of H as a function of k, and furthermore refine the bound to depend on γ. Both bounds are near optimal due to a lower bound by Karpov, Pilipczuk and Zych-Pawlewicz (2019), and they are achieved using new insights about the structure of terminal min-cuts (which hold also for general graphs). In addition, we present a duality between cut sparsifiers and distance sparsifiers for planar graphs with certain γ.

The flow-cut gap problem is defined as the worst-case ratio between the multicommodity-flow and the sparsest-cut values in a given G with k terminals (under arbitrary edge capacities and demands between terminals). We obtain an exponential improvement to the flow-cut gap as a function of γ; this actually follows from a stochastic embedding of G into dominating trees with an optimal expected distortion. Three powerful tools that are used to prove these bounds are, the peeling lemma of Lee and Sidiropoulos (2009), the tree-cut operation of Klein (2006), and a special tree embedding of Sidiropoulos (2010). For vertex-capacitated graphs, we provide the first bound on the flow-cut gap with respect to γ, which holds in the more general setting of submodular vertex capacities.

In a separate line of work, we study an important version of the Traveling Salesman Problem (TSP) called the (rooted) orienteering problem in Euclidean space: Given n points P in $\mathbb{R}^d$, a root point $s \in P$ and a budget $B > 0$, the goal is to find a path that starts from s, has total length at most B, and visits as many points of P as possible. This problem is known to be NP-hard, and we provide a $(1-\epsilon)$ approximation algorithm, that improves the previous running time, due to Chen and Har-Peled (2008), by a factor of $n^{O(d\sqrt{d})}$. Our algorithm reduces the rooted orienteering problem to a multi-path version of k-TSP (a tour of minimum length that visits at least k points), and approximates each of these k-TSP paths by a special parameter called excess.

Contents

1 Introduction

The important family of planar graphs has been studied extensively, and the planarity properties are often leveraged algorithmically to achieve better bounds such as running time or accuracy. Usually these bounds are expressed as a function of the number of vertices or the edges of the graph, but they can be refined by the number of "relevant" faces. More formally, given a planar graph G with a fixed drawing and k vertices designated as terminals, let the *terminal face cover*, denoted by γ, be the minimum number of faces that are incident to all the terminals. The parameter γ represents the topology of the terminals in G, and it has a long history in the study of cuts and (multicommodity) flows [OS81, MNS85, CW04, CSW13, Fil20], shortest paths [Fre95, CX00], and the Steiner tree problem [EMV87, Ber90, KNvL19]. Our work presents new results in the fields of graph compression, graph embeddings and the flow-cut gap with respect to the parameter γ.

Graph compression is motivated by massively big graphs, such as social networks, communication networks and roadmaps, that one wishes to store and communicate more efficiently. There are 3 main paradigms of compression. The first one is to compress the input graph into a small number of bits, i.e., to find a *succinct representation* of the input graph. Itai and Rodeh [IR82] introduced and study this problem; their results were extended and improved by Turan [Tur84] and by Naor [Nao90], who found the optimal number of bits for representing a planar labeled and unlabeled graphs (another notion of compression can be found in [FM95]). Later on, Benczúr and Karger [Kar93, BK96] defined and studied the *edge-sparsification* paradigm, where the input graph G is being compressed into a smaller graph H with the *same* vertex set but *sparse*, i.e. with fewer edges, that maintains certain features (quantities) of G, like distances, cuts, or connectivity. This paradigm has led to important notions, from spanners [PU89] to cut and spectral sparsifiers [BK96, ST11]. The third paradigm is *vertex-sparsification*, which assumes that we are interested only in a small portion of the vertex set of G. Given G and k terminals, we aim to compress it by producing a graph H that contains the *same* terminals but with fewer non-terminal vertices, and maintains (exactly or approximately) some features of the terminals in G. The main advantage of edge and vertex sparsification, is that once the compressed graph H is computed in a preprocessing step, further processing (via classic graph algorithms) can be performed on H instead of on G, using less resources like running time and memory, or achieving better accuracy when the solution is approximate. Our work studies the (vertex) cut-sparsification problem for general and planar graphs, which asks to minimize the size of H while maintaining the minimum cut value between every two subsets of the terminals in G. See Section 1.1 for further discussion and results.

The *flow-cut gap* expresses the relationship between flows and cuts in a given graph.

A spacial case is the well-known max-flow min-cut (single commodity) theorem by Ford and Fulkerson [FF56], which states that in every weighted graph with fixed source s and sink t, the maximum amount of flow from s to t equals the total weight of edges in a minimum (s,t)-cut, i.e. the smallest total weight of the edges whose removal disconnects s from t. This theorem can be proved by the duality of linear programs, where the max-flow and the min-cut problems are formulated as a primal-dual pair of linear programs. A more general case is the multicommodity-flow and the sparsest-cut problems, where the input consists of a weighted graph G with k source-sink pairs (s_i,t_i) and their corresponding demands d_i. The objective in the multicommodity-flow problem is to maximize the fraction α such that all the k demands $\alpha \cdot d_i$ can be routed simultaneously in the graph (without exceeding edge capacities). In the sparsest-cut problem, the objective is to find a cut with minimum *sparsity*, defined as the ratio between the total capacity of edges in the cut, and the total demand separated by this cut. It is known that the sparsest-cut upper bounds the multicommodity-flow in every input, but (unlike the single commodity case) equality is not always achieved. Thus, the flow-cut gap is defined to be the ratio between these two quantities, i.e. the sparsest-cut and the multicommodity-flow on a given input. The main motivation for bounding the gap is to approximate the sparsest-cut problem, which is known to be NP-hard, by solving the multicommodity-flow problem (and the approximation factor is the flow-cut gap). The gap can also be motivated by its equivalence to the stretch in an ℓ_1 embeddings [AR98, LLR95, GNRS04], since bounding the gap immediately implies new embedding results. Our work studies the (multicommodity) flow-cut gap in planar graphs, which involves new optimal results about tree embedding. See Section 1.2 for further discussion and results.

A separate line of our work deals with the *Traveling Salesman Problem* (TSP), which is a fundamental problem in combinatorial optimization, computer science and operations research. In this problem, the input is a list of cities (aka sites) and their pairwise distances, and the goal is to find a (closed) tour of minimum length that visits all the sites. This problem is a prototype for planning routes in almost any context, from logistics to manufacturing, and is therefore studied extensively. In its full generality, TSP is known to be NP-hard to solve and even to approximate, but if we restrict ourselves to inputs where the distances satisfy triangle inequality, i.e., consider metric or Euclidean TSP, then the problem remains NP-hard, but it is no longer hard to approximate (up to a certain constant factor) [GGJ76, Pap77, Tre00]. The Euclidean TSP is a special case of a metric TSP, in which the points (cities) lie in a d-dimensional Euclidean space $\mathbb{R}^d$, and the distance between every two points x and y is defined to be the Euclidean distance between them, i.e. $(\sum_{i=1}^{d}(x_i-y_i)^2)^{1/2}$. For $d=2$ Arora [Aro98] and Mitchell [Mit99] presented a polynomial time approximation scheme (PTAS) based on dynamic programming that finds a tour of length $(1+\epsilon)\mathrm{OPT}$, where OPT stands for the length of the shortest tour. Arora's algorithm can also be extended to the more general case of $d > 2$. One important variant of TSP, out of many others, is the *orienteering* problem, which asks to maximize the number of sites visited when the tour length is constrained by a given budget. This problem models scenarios where the "salesman" has limited resources, such as gasoline, time or battery-life. Clearly, this problem is NP-hard, but it can be approximated up to some constant factor [BCK+07, CH08]. Our work presents an approximation algorithm for the orienteering problem in the Euclidean space. See Section 1.3 for further discussion and results.

1.1 Cut-Sparsifiers of Planar Graphs

Let us define the notion of (vertex) cut-sparsification, and discuss state of the art results for it. Given a (large) undirected graph G with k terminal vertices T and edge costs $c : E \to \mathbb{R}_+$, let $\text{mincut}_G(S)$ be the value of the minimum cut between $S \subset T$ and $\bar{S} = T \setminus S$ in G, i.e. the minimum total weight of edges whose removal from G disconnects every terminal in S from every terminal in $\bar{S}$. A (q, s)-*cut sparsifier* of G is a graph H with the same k terminals, where the size (number of vertices) of H is at most s and its quality is at most q, i.e.

$$\forall S \subset T, \qquad \text{mincut}_G(S) \le \text{mincut}_H(S) \le q \cdot \text{mincut}_G(S).$$

Clearly, there is a tradeoff between the quality $q \ge 1$ and the size $s \ge k$ of cut-sparsifiers, but most of the work that has been done so far addresses two extreme cases as follows. The first case is when we restrict attention to $s = k$, i.e. the vertices of the sparsifier H are only the terminals. In that case, we wish to minimize the quality q, and it is known that $\Omega(\frac{\sqrt{\log k}}{\log\log k}) \le q \le O(\frac{\log k}{\log\log k})$ for general graphs [CLLM10, EGK+14, MM16], and $q = O(1)$ for planar graphs [EGK+14]. The second case is when we restrict attention to quality $q = 1$, i.e. the sparsifier preserves all the terminal min-cuts (minimum cuts that separate terminals) exactly, which is known in the literature as a mimicking network (introduced by [HKNR98]). In this case we wish to minimize the size s, and it is known that $2^{\Omega(k)} \le s \le 2^{2^k}$ for general graphs [HKNR98, KR13, KR14], and that $2^{\Omega(k)} \le s \le O(k^2 2^{2k})$ for planar graphs [KR13, KPZ19].

We improve and refine the upper bound for planar graphs by developing two methods that decompose the terminal min-cuts into "more basic" subsets of edges. Our first decomposition method identifies (in every graph G, even non-planar) a subset of terminal min-cuts, which we call elementary, that generates all the others. Consequently, H is a cut-sparsifier of G if all the elementary terminal min-cuts in G are well-approximated by those in H (see Theorem 2.9 and Corollary 2.10). By restricting our analysis to elementary terminal min-cuts, we manage to improve our pervious bound from [KR13] by a factor of k, and prove that every planar G with

k terminals admits a mimicking network of size $O(k2^{2k})$ which is also a minor of G, and is near optimal by [KPZ19] (see Theorem 2.11 and Corollary 2.12).

We then refine the known bounds in terms of γ, and prove that every planar G with k terminals has at most $O((2k/\gamma)^{2\gamma})$ elementary terminal min-cuts (compared to $O(2^k)$ terminal cuts, see Theorem 2.18), and it admits a mimicking network of size $O(\gamma 2^{2\gamma} k^4)$ which is also a minor of G (see Corollary 2.21); which is near optimal with respect to γ by [KPZ19]. Our proof uses the special γ faces to further decompose the elementary terminal min-cuts of G into disjoint sets of edges, and then bounds the number of these sets by $O(2^\gamma k^2)$ (see Theorem 2.20).

For the special case of $\gamma = 1$, which was famously studied by Okamura and Seymour [OS81], we prove a duality between cut sparsification and distance sparsification when the sparsifier H is required to be a minor of G (see Theorems 2.38 and 2.39). This duality connects problems that were previously studied separately, implying new results, new proofs of known results, and equivalences between open gaps. In addition, we prove that such G admits a mimicking network of size $O(k^4)$ which is also minor of G; independently it was shown that such G also admits a mimicking network of size $O(k^2)$ which is planar but is *not* a minor of G [GHP17].

For further details and proofs, see Section 2, which essentially replicates our published paper [KR20].

1.2 Refined Flow-Cut Gap in Planar Graphs

We also study how the terminal face cover γ affects the flow-cut gap defined as follows. Given an undirected graph G with k terminals and edge costs $c : E(G) \to \mathbb{R}_+$, the *flow-cut gap*, denoted by $\mathsf{gap}(G)$, equals to the ratio between the sparsest-cut and the maximum concurrent multicommodity-flow quantities on G. For general graphs G it is known that $\mathsf{gap}(G) = \Theta(\log k)$ [AR98, LLR95], but when topological restrictions are placed on G, the gap is not settled yet. In particular, it has been conjectured that $\mathsf{gap}(G) = O(1)$ [GNRS04] for planar G, while the known bounds place the (worst-case) gap somewhere between 2 [LR10] and $O(\sqrt{\log k})$ [Rao99].

For the special case where G can be drawn in the plane with all its terminals on the same face, i.e. $\gamma = 1$, Okamura and Seymour proved that $\mathsf{gap}(G) = 1$ [OS81]. This setting can be generalized by considering planar graphs with terminal face cover $\gamma > 1$. The methods of Lee and Sidiropoulos [LS09] show that $\mathsf{gap}(G) \leq 2^{O(\gamma)}$, and the state of the art bound is $\mathsf{gap}(G) \leq 3\gamma$ [CSW13]. We significantly improve this by establishing that $\mathsf{gap}(G) \leq O(\log \gamma)$ (see Theorem 3.3). This is achieved by showing that the edge-weighted shortest-path metric induced on the terminals admits a stochastic embedding into dominating trees with terminal distortion $O(\log \gamma)$ (see Theorem 3.5), which is tight by [GNRS04].

The preceding results refer to the setting of *edge-capacitated* graphs. For *vertex-capacitated* G it can be significantly more challenging to control flow-cut gaps. While there is no exact vertex-capacitated version of the Okamura-Seymour Theorem, an approximate version holds, and the vertex-capacitated flow-cut gap is $O(1)$ on planar G with $\gamma = 1$ [LMM15]. We study the more general case of $\gamma > 1$, and bound the vertex-capacitated flow-cut gap by $O(\gamma)$ (it in fact holds also for submodular vertex capacities).

For further details and proofs, see Section 3, which essentially replicates our published paper [KLR19].

1.3 Faster Algorithms for Euclidean Orienteering and k-TSP

In a separate line of work (not related to the terminal face cover), we study an important version of the Traveling Salesman Problem (TSP) called the *(rooted) orienteering problem* in Euclidean space, which asks to maximize the number of sites visited when the tour length is constrained by a given budget. More formally, given n points P in $\mathbb{R}^d$, a root point $s \in P$ and a budget $\mathcal{B} > 0$, find a path that starts from s, has total length at most $\mathcal{B}$, and visits as many points of P as possible. This problem is known to be NP-hard, hence we study $(1-\delta)$-approximation algorithm for this problem. A $(1-\delta)$-approximate solution is a path satisfying these constraints (starts at s and has length at most $\mathcal{B}$) that visits at least $(1-\delta)k_{\text{opt}}$ points, where k_{opt} denotes the maximum possible, i.e. the number of points visited by an optimal path. The known Polynomial-Time Approximation Scheme (PTAS) for this problem, due to Chen and Har-Peled, runs in time $n^{O(d\sqrt{d}/\delta)}(\log n)^{(d/\delta)^{O(d)}}$ [CH08], and we improve this time bound to $n^{O(1/\delta)}(\log n)^{(d/\delta)^{O(d)}}$ (see Theorem 4.1). For fixed δ and small dimension d, the leading term in their running time is about $n^{O(d\sqrt{d}/\delta)}$, which we improve to $n^{O(1/\delta)}$. Thanks to this improvement, our algorithm is polynomial even for a moderately large dimension, roughly up to $d = O(\log\log n)$ instead of $d = O(1)$.

The algorithm of Chen and Har-Peled [CH08] reduces the orienteering problem into a multi-path version of rooted k-TSP (finding a tour of minimum length that visits at least k points), and thus the heart of their algorithm is a PTAS for the latter, where the approximation is actually with respect to a parameter called *excess*, which can be much smaller than the optimal tour length. This follows an earlier approach of Blum et al. [BCK$^+$07], who introduced the concept of excess-based approximation, and designed a reduction to a simpler (single-path version of) k-TSP. However, that earlier reduction increases the approximation ratio by a constant factor and cannot yield a PTAS. We improve over the algorithm of Chen and Har-Peled by reducing the rooted orienteering problem to a more complicated multi-path version that we call rooted (m,k)-TSP. This problem asks to find m paths that visit k points in total, when the input prescribes the endpoints of all these m paths, and the main challenge is to solve it fast and with good excess-based approximation (see Theorem 4.3).

For further details and proofs, see Section 4, which essentially replicates our paper [GKR20].

2 Refined Vertex Sparsifiers of Planar Graphs

2.1 Opening

A very powerful paradigm when manipulating a huge graph G is to *compress* it, in the sense of transforming it into a small graph H (or alternatively, into a succinct data structure) that maintains certain features (quantities) of G, like distances, cuts, or flows. The basic idea is that once the compressed graph H is computed in a preprocessing step, further processing can be performed on H instead of on G, using less resources like running time and memory, or achieving better accuracy when the solution is approximate. This paradigm has lead to remarkable successes, such as faster running time for fundamental problems, and the introduction of important concepts, from spanners [PU89] to cut and spectral sparsifiers [BK96, ST11]. In these examples, H is a subgraph of G with the *same* vertex set but *sparse*, and is sometimes called an edge sparsifier. In contrast, we aim to reduce the number of vertices in G, using so-called vertex sparsifiers.

In the vertex-sparsification scenario, G has k designated vertices called *terminals*, and the goal is to construct a small graph H that contains these terminals, and maintains some of their features inside G, like distances or cuts. Throughout, a *k-terminal network*, denoted $G = (V, E, T, c)$, is an undirected graph (V, E) with edge weights $c : E \to \mathbb{R}_+$ and terminals set $T \subset V$ of size $|T| = k$. As usual, a *cut* is a partition of the vertices, and its *cutset* is the set of edges that connect between different parts. Interpreting the edge weights as capacities, the *cost* of a cut $(W, V \setminus W)$ is the total weight of the edges in the respective cutset.

We say that a cut $(W, V \setminus W)$ *separates* a terminals subset $S \subset T$ from $\bar{S} \stackrel{\text{def}}{=} T \setminus S$ (or in short that it is *S-separating*), if all of S is on one side of the cut and $\bar{S}$ on the other side, i.e., $W \cap T$ equals either S or $\bar{S}$. We denote by $\text{mincut}_G(S)$ the minimum cost of an S-separating cut in G, where by a consistent tie-breaking mechanism, such as edge-weights perturbation, we assume throughout that the minimum is attained by only one cut, which we call the *minimum terminal cut* (of S).

Definition 2.1. *A network $H = (V_H, E_H, T, c_H)$ is a* cut sparsifier *of $G = (V, E, T, c)$ with* quality *$q \geq 1$ and* size *$s \geq k$ (or in short, a (q, s)-cut-sparsifier), if its size is $|V_H| \leq s$ and*

$$\forall S \subset T, \qquad \text{mincut}_G(S) \leq \text{mincut}_H(S) \leq q \cdot \text{mincut}_G(S). \tag{1}$$

In words, (1) requires that every minimum terminal cut in H approximates the corresponding one in G. Throughout, we consider only $S \neq \emptyset, T$ although for brevity we will not write it explicitly.

Two special cases are particularly important for us. One is quality $q = 1$, or a $(1, s)$-cut-sparsifier, which is known in the literature as a *mimicking network* and was introduced by [HKNR98]. The second case is a cut sparsifier H that is furthermore a minor of G, and then we call it a *minor cut sparsifier*, and similarly for a *minor mimicking network*. In all our results, the sparsifier H is actually a minor of G, which can be important in some applications; for instance, if G is planar then H admits planar-graph algorithms.

In known constructions of mimicking networks ($q = 1$), the sparsifier's size s highly depends on the number of constraints in (1) that are really needed. Naively, there are at most 2^k constraints, one for every minimum terminal cut (this can be slightly optimized, e.g., by symmetry of S and $\bar{S}$). This naive bound was used to design, for an arbitrary network G, a mimicking network whose size s is exponential in the number of constraints, namely $s \le 2^{2^k}$ [HKNR98]. A slight improvement, that is still doubly exponential in k, was obtained by using the submodularity of cuts to reduce the number of constraints [KR14]. For a planar network G, the mimicking network size was improved to a polynomial in the number of constraints, namely $s \le k^2 2^{2k}$ [KR13], and this bound is actually near-optimal, due to a very recent work showing that some planar graphs require $s = 2^{\Omega(k)}$ [KPZ19]. In this work we explore the structure of minimum terminal cuts more deeply, by introducing technical ideas that are new and different from previous work like [KR13].

Our Approach. We take a closer look at the mimicking network size s of planar graphs, aiming at bounds that are more sensitive to the given network G. For example, we would like to "interpolate" between the very special case of an outerplanar G, which admits a mimicking network of size $s = O(k)$ [CSWZ00], and an arbitrary planar G for which $s \le 2^{O(k)}$ is known and optimal [KR13, KPZ19]. Our results employ a graph parameter $\gamma(G)$, defined next.

Definition 2.2 (Terminal Face Cover)**.** *The* terminal face cover $\gamma = \gamma(G)$ *of a planar k-terminal network G with a given drawing*[1] *is the minimum number of faces that are incident to all the k terminals, and thus $1 \le \gamma \le k$.*

This graph parameter $\gamma(G)$ is well-known to be important algorithmically. For example, it can be used to control the running time of algorithms for shortest-path problems [Fre91, CX00], for cut problems [CW04, Ben09], and for multicommodity flow problems [MNS85]. For the complexity of computing an optimal/approximate face cover $\gamma(G)$, see [BM88, Fre91].

When $\gamma = 1$, all the terminals lie on the boundary of the same face, which we may assume to be the outerface. This special case was famously shown by Okamura and

[1]We can let γ refer to the best drawing of G, and then our results might be non-algorithmic.

Seymour [OS81] to have a flow-cut gap of 1 (for multicommodity flows). Later work showed that for general gamma, the flow-cut gap is at most 3γ [LS09, CSW13], and after the current results were announced this bound was further improved to $O(\sqrt{\log \gamma})$ [KLR19, Fil20].

2.1.1 Main Results and Techniques

We provide new bounds for mimicking networks of planar graphs. In particular, our main result refines the previous bound so that it depends exponentially on $\gamma(G)$ rather than on k, This yields much smaller mimicking networks in important cases, for instance, when $\gamma = O(1)$ we achieve size $s = \text{poly}(k)$. See Table 1 for a summary of known and new bounds. Technically, we develop two methods to decompose the minimum terminal cuts into "more basic" subsets of edges, and then represent the constraints in (1) using these subsets. This is equivalent to reducing the number of constraints, and leads (as we hinted above) to a smaller sparsifier size s. A key difference between the methods is that the first one in effect restricts attention to a subset of the constraints in (1), while the second method uses alternative constraints.

Decomposition into Elementary Cutsets. Our first decomposition method identifies (in every graph G, even non-planar) a subset of minimum terminal cuts that "generates" all the other ones, as follows. First, we call a cutset *elementary* if removing its edges disconnects the graph into exactly two connected components (Definition 2.5). We then show that every minimum terminal cut in G can be decomposed into a disjoint union of elementary ones (Theorem 2.9), and use this to conclude that if all the elementary cutsets in G are well-approximated by those in H, then H is a cut sparsifier of G (Corollary 2.10).

Combining this framework with prior work on planar sparsifier [KR13], we devise the following bound that depends on $\mathcal{T}_e(G)$, the set of elementary cutsets in G.

- **Generic bound:** Every planar graph G has a mimicking network of size $s = O(k) \cdot |\mathcal{T}_e(G)|^2$; see Theorem 2.11.

Trivially $|\mathcal{T}_e(G)| \leq 2^k$, and we immediately achieve $s = O(k2^{2k})$ for all planar graphs (Corollary 2.12). This improves over the known bound [KR13] slightly (by factor k), and stems directly from the restriction to elementary cutsets (which are simple cycles in the planar-dual graph).

Using the same generic bound, we further obtain mimicking networks whose size is *polynomial* in k (but inevitably exponential in γ), starting with the base case $\gamma = 1$ and then building on it, as follows.

- **Base case:** If $\gamma(G) = 1$, then $|\mathcal{T}_e(G)| \leq O(k^2)$ and thus G has a mimicking network of size $s = O(k^4)$;[2] see Theorem 2.15 and Corollary 2.17.
- **General case, first bound:** If $\gamma(G) \geq 1$, then $|\mathcal{T}_e(G)| \leq (2k/\gamma)^{2\gamma}$ and thus G has a mimicking network of size $s = O(k(2k/\gamma)^{4\gamma})$; see Theorem 2.18 and Corollary 2.19.

The last bound on $|\mathcal{T}_e(G)|$ is clearly wasteful (for $\gamma = k$, it is roughly quadratically worse than the trivial bound). To avoid over-counting of edges that belong to multiple elementary cutsets, we devise a better decomposition.

Further Decomposition of Elementary Cutsets. Our second method decomposes each elementary cutset even further, in a special way such that we can count the underlying fragments (special subsets of edges) without repetitions, and this yields our main result.

- **General case, second bound.** When $\gamma(G) \geq 1$, there are $O(2^\gamma k^2)$ subsets of edges, such that every elementary cutset in G can be decomposed into a disjoint union of some of these subsets. Thus, G admits a mimicking network of size $O(\gamma 2^{2\gamma} k^4)$; see Theorem 2.20 and Corollary 2.21.

Additional Results. First, all our cut sparsifiers are also approximate flow sparsifers, by straightforward application of the known bounds on the flow-cut gap, see Section 2.4.4. Second, our decompositions easily yield a succinct data structure that stores all the minimum terminal cuts of a planar graph G. Its storage requirement depends on $|\mathcal{T}_e(G)|$, which is bounded as above, see Section 2.5 for details.

Finally, we show a duality between cut and distance sparsifiers (for certain graphs), and derive new relations between their bounds, as explained next.

2.1.2 Cuts vs. Distances

Although in several known scenarios cuts and distances are closely related, the following notion of distance sparsification was studied separately, with no formal connections to cut sparsifiers [Gup01, CXKR06, BG08, KNZ14, KKN15, GR16, CGH16, Che18, Fil18, FKT19].

Definition 2.3. *A network $H = (V_H, E_H, T, c_H)$ is called a (q, s)*-distance-approximating minor (abbreviated DAM) *of $G = (V, E, T, c)$, if it is a minor of*

[2] The generic bound implies $s = O(k^5)$, but we can slightly improve it in this case.

Graphs	Size	Minor	Reference
General	$2^{2^k} \approx 2^{\binom{(k-1)}{(k-1)/2}}$	no	[HKNR98, KR14]
Planar	$O(k^2 2^{2k})$	yes	[KR13]
Planar	$O(k 2^{2k})$	yes	Corollary 2.12
Planar $\gamma = \gamma(G)$	$O(\gamma 2^{2\gamma} k^4)$	yes	Corollary 2.21
Planar $\gamma(G) = 1$	$O(k^4)$	yes	Corollary 2.17
Planar $\gamma(G) = 1$	$O(k^2)$	no	[GHP17]
General	$2^{\Omega(k)}$	no	[KR13, KR14] lower bound
Planar	$2^{\Omega(k)}$	no	[KPZ19] lower bound

Table 1: Known and new bounds for mimicking networks.

G, its size is $|V_H| \le s$ and

$$\forall t, t' \in T, \qquad d_G(t,t') \le d_H(t,t') \le q \cdot d_G(t,t'), \tag{2}$$

where $d_G(\cdot,\cdot)$ is the shortest-path metric in G with respect to $c(\cdot)$ as edge lengths.

We emphasize that the well-known planar duality between cuts and cycles does *not* directly imply a duality between cut and distance sparsifiers. We nevertheless do use this planar-duality approach, but we need to break "shortest cycles" into "shortest paths", which we achieve by adding new terminals (ideally not too many).

- Fix $k, q, s \ge 1$. Then all planar k-terminal networks with $\gamma = 1$ admit a minor (q, s)-cut sparsifier if and only if all these networks admit an $(q, O(s))$-DAM; see Theorems 2.38 and 2.39.

This result yields new cut-sparsifier bounds in the special case $\gamma = 1$ (see Section 2.6.3). Notice that in this case of $\gamma = 1$ the flow-cut gap is 1 [OS81], hence the three problems of minor sparsification (of distances, of cuts, and of flows), all have the same asymptotic bounds and gaps.

This duality can be extended to general $\gamma \ge 1$ (including $\gamma = k$), essentially at the cost of increasing the number of terminals, as follows. If for some functions $q(\cdot)$ and $s(\cdot)$, all planar k-terminal networks with given γ admit a $(q(k), s(k))$-DAM, then all networks in this class admits also a minor $(q(\gamma 2^\gamma k^2), s(\gamma 2^\gamma k^2))$-cut sparsifier. For $\gamma = k$, we can add only $k2^k$ new terminals instead of $k^3 2^k$. We omit the proof of this extension, as applying it to the known bounds for DAM yields alternative proofs for known/our cut-sparsifier bounds, but no new results. For example, using the reduction together with the known upper bound of $(1, k^4)$-DAM, we get that every planar k-terminal network with $\gamma(G) = k$ admits a minor mimicking network of size $O((k2^k)^4)$.

Comparison with previous techniques. Probably the closest notion to duality between cut sparsification and distance sparsification is Räcke's powerful method [Räc08], adapted to vertex sparsification as in [CLLM10, EGK+14, MM16]. However, in his method the cut sparsifier H is inherently randomized; this is acceptable if H contains only the terminals, because we can take its "expectation" $\bar{H}$ (a complete graph with expected edge weights), but it is calamitous when H contains non-terminals, and then each randomized outcome has different vertices. Another related work, by Chen and Wu [CW04], reduces multiway-cut in a planar network with $\gamma(G) = 1$ to a minimum Steiner tree problem in a related graph G'. Their graph transformation is similar to one of our two reductions, although they show a reduction that goes in one direction rather than an equivalence between two problems.

2.1.3 Related Work

Cut and distance sparsifiers were studied extensively in recent years, in an effort to optimize their two parameters, quality q and size s. The foregoing discussion is arranged by the quality parameter, starting with $q = 1$, then $q = O(1)$, and finally quality that grows with k.

Cut Sparsification. Let us start with $q = 1$. Apart from the already mentioned work on a general graph G [HKNR98, KR14, KR13], there are also bounds for specific graph families, like bounded-treewidth or planar graphs [CSWZ00, KR13, KPZ19]. For planar G with $\gamma(G) = 1$, there is a recent tight upper bound $s = O(k^2)$ [GHP17] (independent of our work), where the sparsifier is planar but is *not* a minor of the original graph.

We proceed to a constant quality q. Chuzhoy [Chu12] designed an $(O(1), s)$-cut sparsifier, where s is polynomial in the total capacity incident to the terminals in the original graph, and certain graph families (e.g., bipartite) admit sparsifiers with $q = 1 + \epsilon$ and $s = \mathrm{poly}(k/\epsilon)$ [AGK14].

Finally, we discuss the best quality known when $s = k$, i.e., the sparsifier has only the terminals as vertices. In this case, it is known that $q = O(\log k/ \log\log k)$ [Moi09, LM10, CLLM10, EGK+14, MM16], and there is a lower bound $q = \Omega(\sqrt{\log k})$ [MM16]. For networks that exclude a fixed minor (e.g., planar) it is known that $q = O(1)$ [EGK+14], and for trees $q = 2$ [GR16] (where the sparsifier is *not* a minor of the original tree).

Distance Sparsification. A separate line of work studied the tradeoff between the quality q and the size s of a distance approximation minor (DAM). For $q = 1$, every graph admits DAM of size $s = O(k^4)$ [KNZ14], and there is a lower bound of $s = \Omega(k^2)$ even for planar graphs [KNZ14]. Independently of our work, Goranci, Henzinger and Peng [GHP17] recently constructed, for planar graphs with $\gamma(G) = 1$, a $(1, O(k^2))$-distance sparsifier that is planar but *not* a minor of the original graph. Proceeding to quality $q = O(1)$, planar graphs admit a DAM with $q = 1 + \epsilon$ and $s = O(k \log k/\epsilon)^2$ [CGH16], and certain graph families, such as trees and outerplanar graphs, admit a DAM with $q = O(1)$ and $s = O(k)$ [Gup01, BG08, CXKR06, KNZ14]. When $s = k$ (the sparsifier has only the terminals as vertices), then known quality is $q = O(\log k)$ for every graph [KKN15, Che18, Fil18]. Additional tradeoffs and lower bounds can be found in [CXKR06, KNZ14, CGH16].

2.1.4 Preliminaries

Let $G = (V, E, T, c)$ be a k-terminal network, and denote its k terminals by $T = \{t_1, \dots, t_k\}$. We assume without loss of generality that G is connected, as otherwise we can construct a sparsifier for each connected component separately. For every $S \subset T$, let $\operatorname{argmincut}_G(S)$ denote the *argument* of the minimizer in $\operatorname{mincut}_G(S)$, i.e., the minimum-cost cutset that separates S from $\bar{S} = T \setminus S$ in G. We assume that the minimum is unique by a perturbation of the edge weights. Throughout, when G is clear from the context, we use the shorthand

$$E_S \stackrel{\text{def}}{=} \operatorname{argmincut}_G(S). \tag{3}$$

Similarly, $CC(E_S)$ is a shorthand for the set of connected components of the graph $G \backslash E_S$. Define the *boundary* of $W \subseteq V$, denoted $\delta(W)$, as the set of edges with exactly one end point in W, and observe that for every connected component $C \in CC(E_S)$ we have $\delta(C) \subseteq E_S$. By symmetry, $E_S = E_{\bar{S}}$. And since G is connected and $S \neq \emptyset, T$, we have $E_S \neq \emptyset$ and $|CC(E_S)| \geq 2$. In addition, by the minimality of E_S, every connected component $C \in CC(E_S)$ contains at least one terminal.

Lemma 2.4 (Lemma 2.2 in [KR13]). *For every two subsets of terminals $S, S' \subset T$ and their corresponding minimum cutsets $E_S, E_{S'}$, every connected component $C \in CC(E_S \cup E_{S'})$ contains at least one terminal.*

2.2 Elementary Cutsets in General Graphs

In this section we define a special set of cutsets called elementary cutsets (Definition 2.5), and prove that these elementary cutsets generate all other relevant cutsets, namely, the minimum terminal cutsets in the graph (Theorem 2.9). Therefore, to produce a cut sparsifier, it is enough to preserve only these elementary cutsets (Corollary 2.10). In the following discussion, we fix a network $G = (V, E, T, c)$ and employ the notations E_S, $CC(E_S)$ and $\delta(W)$ set up in Section 4.1.3.

Definition 2.5 (Elementary Cutset)**.** *Fix $S \subset T$. Its minimum cutset E_S is called an* elementary cutset *if $|CC(E_S)| = 2$.*

Definition 2.6 (Elementary Component)**.** *A subset $C \subseteq V$ is called an* elementary component *if $\delta(C)$ is an elementary cutset for some $S \subset T$, i.e., $\delta(C) = E_{C \cap T}$ and $|CC(\delta(C))| = 2$.*

Although the following two lemmas are quite straightforward, they play a central role in the proof of Theorem 2.9.

Lemma 2.7. *Fix a subset $S \subset T$ and its minimum cutset E_S. The boundary of every $C \in CC(E_S)$ is itself the minimum cutset separating the terminals $T \cap C$ from $T \setminus C$ in G, i.e., $\delta(C) = E_{T \cap C}$.*

Proof. Assume toward contradiction that $\delta(C) \neq E_{T\cap C}$. Since both sets of edges separate between the terminals $T \cap C$ and $T \setminus C$, then $c(E_{T\cap C}) < c(\delta(C))$. Let us replace the edges $\delta(C)$ by the edges $E_{T\cap C}$ in the cutset of E_S and call this new set of edges E'_S, i.e. $E'_S = (E_S \setminus \delta(C)) \cup E_{T\cap C}$. It is clear that $c(E'_S) < c(E_S)$. We will prove that E'_S is also a cutset that separates between S and $\bar{S}$ in the graph G, contradicting the minimality of E_S.

Assume without loss of generality that $T \cap C \subseteq S$, and consider $E_S \setminus \delta(C)$. By the minimality of E_S all the neighbors of C contain terminals of $\bar{S}$, therefore the cutset $E_S \setminus \delta(C)$ separates the terminals $S \setminus (T \cap C)$ from $\bar{S} \cup (T \cap C)$ in G. Now consider $E'_S = (E_S \setminus \delta(C)) \cup E_{T\cap C}$ and note that the connected component $C_{C+N_S(C)}$ contains all the terminals $T \cap C$ and some terminals of $\bar{S}$. This cutset E'_S clearly separates $T \cap C$ from all other terminals, and also separates $S \setminus (T \cap C)$ from $\bar{S} \cup (T \cap C)$. Altogether this cutset separates between S and $\bar{S}$ in G, and the lemma follows. $\square$

Lemma 2.8. *For every $S \subset T$, at least one component in $CC(E_S)$ is elementary.*

Proof. Fix $S \subset T$. Lemma 2.7 yields that $\delta(C) = E_{C\cap T}$ for every $C \in CC(E_S)$, thus it left to prove that there exists $C' \in CC(E_S)$ such that $|CC(\delta(C'))| = 2$. For simplicity, we shall represent our graph G as a bipartite graph $\mathcal{G}_S$ whose its vertices and edges are $CC(E_S)$ and E_S respectively, i.e. we get $\mathcal{G}_S$ by contracting every $C \in CC(E_S)$ in G into a vertex v_C. Let $V_1(\mathcal{G}_S) = \{v_C : C \cap T \subseteq S\}$ and $V_2(\mathcal{G}_S) = \{v_C : C \cap T \subseteq \bar{S}\}$ be the partition of $V(\mathcal{G}_S)$ into two sets. By the minimality of E_S the graph $\mathcal{G}_S$ is connected, and each of $V_1(\mathcal{G}_S)$ and $V_2(\mathcal{G}_S)$ is an independent set.

For every connected component $C \in CC(E_S)$, it is easy to see that $|CC(\delta(C))| = 2$ if and only if $\mathcal{G}_S \setminus \{v_C\}$ is connected. Since $\mathcal{G}_S$ is connected, it has a spanning tree and thus $\mathcal{G}_S \setminus \{v_{C'}\}$ is connected for every leaf $v_{C'}$ of that spanning tree, and the lemma follows. $\square$

Theorem 2.9 (Decomposition into Elementary Cutsets)**.** *For every $S \subset T$, the minimum cutset E_S can be decomposed into a disjoint union of elementary cutsets.*

The idea of the proof is to iteratively decrease the number of connected components in $CC(E_S)$ by uniting an elementary connected component with all its neighbors (while recording the cutset between them), until we are left with only one connected component — all of V.

Proof of Theorem 2.9. We will need the following definition. Given $S \subset T$ and its minimum cutset E_S, we say that two connected components $C, C' \in CC(E_S)$ are *neighbors with respect to* E_S, if E_S has an edge from C to C'. We denote by $N_S(C) \subseteq CC(E_S)$ the set of neighbors of C with respect to E_S. Observe that removing $\delta(C)$ from the cutset E_S is equivalent to uniting the connected component C with all its neighbors $N_S(C)$. Denoting this new connected component by $C_{C+N_S(C)}$ we get that $CC(E_S \setminus \delta(C)) = \Big(CC(E_S) \setminus \big(\{C\} \cup N_S(C)\big)\Big) \cup \{C_{C+N_S(C)}\}$.

Let E_S be a minimum cutset that separates S from $\bar{S}$. By Lemma 2.8, there exists a component $C \in CC(E_S)$ that is elementary, and by Lemma 2.7, $\delta(C) = E_{T\cap C}$. Assume without loss of generality that $T \cap C \subseteq S$ (rather than $\bar{S}$), and unite C with all its neighbors $N_S(C)$. Now, we would like to show that this step is equivalent to "moving" the terminals in C from S to $\bar{S}$. Clearly, the new cutset $E_S \setminus \delta(C)$ separates the terminals $S' = S \setminus (T \cap C)$ from $T \setminus S' = \bar{S} \cup (T \cap C)$, but to prove that

$$E_{S'} = E_S \setminus \delta(C), \tag{4}$$

we need to argue that this new cutset has *minimum cost* among those separating S' from $\bar{S}'$. To this end, assume to the contrary; then $E_{S'}$ must have a strictly smaller cost than $E_S \setminus \delta(C)$, because both cutsets separate S' from $T \setminus S'$. Now similarly to the proof of Lemma 2.7, it follows that $E_{S'} \cup \delta(C)$ separates S from $\bar{S}$, and has a strictly smaller cost than E_S, which contradicts the minimality of E_S.

Using (4), we can write $E_S = \delta(C) \uplus E_{S'}$ and continue iteratively with $E_{S'}$ while it is non-empty (i.e., $|CC(E_{S'})| > 1$). Formally, the theorem follows by induction on $|CC(E_S)|$.

□

To easily examine all the elementary cutsets in a graph G, we define

$$\mathcal{T}_e(G) \stackrel{\text{def}}{=} \{S \subset T \;:\; |CC(E_S)| = 2\}.$$

Using Theorem 2.9, the cost of every minimum terminal cut can be recovered, in a certain manner, from the costs of the elementary cutsets of G, and this yields the following corollary.

Corollary 2.10. *Let H be a k-terminal network with same terminals as G. If $\mathcal{T}_e(G) = \mathcal{T}_e(H)$ and*

$$\forall S \in \mathcal{T}_e(G), \qquad \text{mincut}_G(S) \leq \text{mincut}_H(S) \leq q \cdot \text{mincut}_G(S), \tag{5}$$

then H is a cut-sparsifier of G of quality q.

Proof. Given G and H as above, we only need to prove (1). To this end, fix $S \subset T$. Observe that for every $\varphi \subseteq \mathcal{T}_e(G) = \mathcal{T}_e(H)$, the set $\cup_{S' \in \varphi} \text{argmincut}_G(S')$ is S-separating in G if and only if for every $t \in S$ and $t' \in \bar{S}$ there exists $S' \in \varphi$ such that without loss of generality $t \in S'$ and $t' \notin S'$. Thus, if φ is a partition of S, i.e. $S = \biguplus_{S' \in \varphi} S'$, then $\cup_{S' \in \varphi} \text{argmincut}_G(S')$ is S-separating in G. Since the same arguments hold also for H, we get the following:

$$\bigcup_{S' \in \varphi} \text{argmincut}_G(S') \text{ is } S\text{-separating in } G \Leftrightarrow \bigcup_{S' \in \varphi} \text{argmincut}_H(S') \text{ is } S\text{-separating in } H.$$

By Theorem 2.9, there exists $\varphi_G \subseteq \mathcal{T}_e(G)$ such that $S = \biguplus_{S' \in \varphi_G} S'$ and $\text{argmincut}_G(S) = \biguplus_{S' \in \varphi_G} \text{argmincut}_G(S')$, thus

$$\text{mincut}_H(S) \leq \sum_{S' \in \varphi_G} \text{mincut}_H(S') \leq q \cdot \sum_{S' \in \varphi_G} \text{mincut}_G(S') = q \cdot \text{mincut}_G(S).$$

Applying Theorem 2.9 to H together with an analogous argument yields that $\text{mincut}_G(S) \leq \text{mincut}_H(S)$, which proves (1) and the corollary follows. □

2.3 Mimicking Networks for Planar Graphs

We now present an application of our results in Section 2.2. We begin with a bound on the mimicking network size for a planar graph G as a function of the number of elementary cutsets (Theorem 2.11). We then obtain an upper bound of $O(k2^{2k})$ for every planar network (Corollary 2.12), which improves the previous work [KR13, Theorem 1.1] by a factor of k, thanks to the use of elementary cuts. The underlying reason is that the previous analysis in [KR13] considers all the 2^k possible terminal cutsets, and each of them is a collection of at most k simple cycles in the dual graph G^*. We can consider only the elementary cutsets by Corollary 2.10, and each of them is a simple cycle in G^* by Definition 2.5. Thus, we consider a total of 2^k simple cycles, saving a factor of k over the earlier naive bound of $k2^k$ simple cycles.

Theorem 2.11. *Every planar network G, in which $|CC(E_S \cup E_{S'})| \leq \alpha$ for all $S, S' \in \mathcal{T}_e(G)$, admits a minor mimicking network H of size $O(\alpha \cdot |\mathcal{T}_e(G)|^2)$.*

The proof of this theorem appears in Section 2.3.1. It is based on applying the machinery of [KR13], but restricting the analysis to elementary cutsets.

Corollary 2.12. *Every planar network G admits a minor mimicking network of size $O(k2^{2k})$.*

Proof. Apply Theorem 2.11, using an easy bound $\alpha = O(k)$ from Lemma 2.4, and a trivial bound on the number of elementary cutsets $|\mathcal{T}_e(G)| \leq 2^k$. $\square$

2.3.1 Proof of Theorem 2.11

Given a k-terminal network G and $\alpha > 0$ such that $|CC(E_S \cup E_{S'})| \leq \alpha$ for every $S, S' \in \mathcal{T}_e(G)$, we prove that it admits a minor mimicking network H of size $O(\alpha|\mathcal{T}_e(G)|^2)$. Let $\hat{E} = \cup_{S\in\mathcal{T}_e(G)} E_S$, and construct H by contracting every connected component of $G \setminus \hat{E}$ into a single vertex. Notice that edge contractions can only increase the cost of any minimum terminal cut, and that in our construction edges of an elementary cutset of G are never contracted. Thus, the resulting H is a minor of G, that maintains all the elementary cutsets of G, and by Corollary 2.10 H maintains all the terminal mincuts of G. We proceed to bound the number of connected components in $G \setminus \hat{E}$, as this will clearly be the size of our mimicking network H. The crucial step here is to use the planarity of G by employing the dual graph of G denoted by G^* (for basic notions of planar duality see Section 2.7).

Loosely speaking, the elementary cutsets in G correspond to cycles in the dual graph G^*, and thus we consider the dual edges of $\hat{E}$, which may be viewed as a subgraph of G^* comprising of (many) cycles. We then use Euler's formula and the special structure of this subgraph of cycles; more specifically, we count its meeting vertices, which turns out to require the aforementioned bound of α for two sets of terminals S, S'. This gives us a bound on the number of faces in this subgraph, which in turn is exactly the number of connected components in the primal graph (Lemma 2.14). Observe that removing edges from a graph G can disconnect it into (one or more) connected components. The next lemma characterizes this behavior in terms of the dual graph G^*. Let $V_m(G)$ be all the vertices in the graph G with degree ≥ 3, and call them *meeting vertices* of G. The following lemma bounds the number of meeting vertices in two elementary cuts by $O(\alpha)$.

Lemma 2.13. *For every two subsets of terminals $S, S' \in \mathcal{T}_e(G)$, the dual graph $G^*[E_S^* \cup E_{S'}^*]$ has at most 2α meeting vertices.*

Proof Sketch. For simplicity denote by $G_{SS'}^*$ the graph $G^*[E_S^* \cup E_{S'}^*]$. By our assumption, the graph $G \setminus (E_S \cup E_{S'})$ has at most α connected components. By Lemma 2.48 every connected component in $G \setminus (E_S \cup E_{S'})$ corresponds to a face in $G_{SS'}^*$.

Therefore, $G^*_{SS'}$ has at most α faces. Let $V_{SS'}, E_{SS'}$ and $F_{SS'}$ be the vertices, edges and faces of the graph $G^*_{SS'}$. Note that the degree of every vertex in that graph is at least 2. Thus, by the degree-sum formula (the total degree of all vertices equals to twice the number of edges), $2|E_{SS'}| \geq 2|V_{SS'} \setminus V_m(G^*_{SS'})| + 3|V_m(G^*_{SS'})|$ and so $|E_{SS'}| \geq |V_{SS'}| + \frac{1}{2}V_m(G^*_{SS'})$. Together with Euler formula we get that $\alpha \geq |F_{SS'}| \geq |E_{SS'}| - |V_{SS'}| \geq \frac{1}{2}|V_m(G^*_{SS'})|$, and the lemma follows. $\square$

Lemma 2.14. *The dual graph $G^*[\hat{E}^*]$ has at most $O(\alpha|\mathcal{T}_e(G)|^2)$ faces. Thus, $G \setminus \hat{E}$ has at most*
$O(\alpha|\mathcal{T}_e(G)|^2)$ connected components.

Proof Sketch. For simplicity denote by $\hat{G}^*$ the graph $G^*[\hat{E}^*]$, and let $E_m(\hat{G}^*)$ be all the edges in $\hat{G}^*$ that are incident to meeting vertices. Fix an elementary subset of terminals $S \in \mathcal{T}_e(G)$. By Lemma 2.13 there are at most 2α meeting vertices in $G^*[E^*_S \cup E^*_{S'}]$, for every $S' \in \mathcal{T}_e(G)$. Summing over all the different S' in $\mathcal{T}_e(G)$ we get that there are at most $2\alpha|\mathcal{T}_e(G)|$ meeting vertices on the cycle E^*_S in the graph $\hat{G}^*$. Since the degree of every vertex in $G^*(E^*_S)$ is 2, we get that

$$|E^*_S \cap E_m(\hat{G}^*)| \leq 2|V(G^*(E^*_S)) \cap V_m(\hat{G}^*)| \leq 4\alpha|\mathcal{T}_e(G)|.$$

Again summing over at most $|\mathcal{T}_e(G)|$ different elementary subsets S we get that $|E_m(\hat{G}^*)| \leq 4\alpha|\mathcal{T}_e(G)|^2$. Plugging it into Euler formula for the graph $G^*[\hat{E}^*]$, together with the inequality $|E(\hat{G}^*) \setminus E_m(\hat{G}^*)| \leq |V(\hat{G}^*) \setminus V_m(\hat{G}^*)|$ by the fact that the two sides represent the edges and vertices of a graph consisting of vertex-disjoint paths (because its maximum degree is at most 2), we get the following

$$\begin{aligned}
|F(\hat{G}^*)| &= |E(\hat{G}^*)| - |V(\hat{G}^*)| + 1 + |CC(\hat{G}^*)| \\
&\leq |E_m(\hat{G}^*)| - |V_m(\hat{G}^*)| + 1 + |CC(\hat{G}^*)| \\
&\leq 4\alpha|\mathcal{T}_e(G)|^2 + 1 + |CC(\hat{G}^*)|.
\end{aligned}$$

Since $|\mathcal{T}_e(G)| \geq k$ it left to bound $|CC(\hat{G}^*)|$ by k. Assume towards contradiction that $|CC(\hat{G}^*)| \geq k+1$, thus there exists a connected component W in $\hat{G}^*$ that does not contains a terminal face of G^*. By the construction of $\hat{E}^*$, W contains at least one elementary shortest cycle that separates between terminal faces of G^* in contradiction. Finally, Lemma 2.48 with $M = \hat{E}$ yields that $|CC(G \setminus \hat{E})| = |F(G^*[\hat{E}^*])| = O(\alpha|\mathcal{T}_e(G)|^2)$ and the lemma follows. $\square$

Recall that we construct our mimicking network H by contracting every connected component of $G \setminus \hat{E}$ into a single vertex. By Lemma 2.14 we get that H is a minor of G of size $O(\alpha|\mathcal{T}_e(G)|^2)$ and Theorem 2.11 follows.

2.4 Mimicking Networks for Planar Graphs with Bounded $\gamma(G)$

In this section, the setup is that $G = (V, E, T, c)$ is a planar k-terminal network with terminal face cover $\gamma = \gamma(G)$. Let $f_1, \ldots, f_\gamma$ be faces that are incident to all the terminals, and let k_i denote the number of terminals incident to face f_i. We can in effect assume that $\sum_{i=1}^{\gamma} k_i = k$, because we can count each terminal as incident to only one face, and "ignore" its incidence to the other $\gamma - 1$ faces (if any).

Our goal is to construct for G a mimicking network H, and bound its size as a function of k and $\gamma(G)$. Our construction of H is the same as in Theorem 2.11, and the challenge is to bound its size. The implications to flow sparsifiers are discussed in Section 2.4.4.

All terminals are on one face. We start with the basic case $\gamma = 1$, i.e., all the terminals are on the same face, which we can assume to be the outerface. The idea is to apply Theorem 2.11. The first step is to characterize all the elementary cutsets, which yields immediately an upper bound on their number. The second step is to analyze the interaction between any two elementary cutsets.

Theorem 2.15. *In every planar k-terminal network G with $\gamma(G) = 1$, the number of elementary cutsets is $|\mathcal{T}_e(G)| \leq \binom{k}{2}$.*

The proof, appearing in Section 2.4.1, is based on two observations that view the outerface as a cycle of vertices: (1) every elementary cutset disconnects the outerface's cycle into two paths, which we call intervals (see Definition 2.22); and (2) every such interval can be identified by the terminals it contains. It then follows that every elementary cutset E_S is uniquely determined by two terminals, leading to the required bound.

The next lemma bounds the interaction between any two elementary cutsets. Its proof appears at the end of Section 2.4.1.

Lemma 2.16. *For every planar k-terminal network G with $\gamma(G) = 1$, and for every $S, S' \in \mathcal{T}_e(G)$, there are at most 4 connected components in $G \setminus (E_S \cup E_{S'})$.*

Corollary 2.17. *Every planar k-terminal network G with $\gamma(G) = 1$ admits a minor mimicking network of size $s = O(k^4)$.*

Proof. Apply Theorem 2.11, with $\alpha = 4$ from Lemma 2.16 and $|\mathcal{T}_e(G)| \leq k^2$ from Theorem 2.15. $\square$

All terminals are on γ faces - First bound. Our first (and weaker) bound for the general case $\gamma \geq 1$ follows by applying Theorem 2.11. To this end, we bound the number of elementary cutsets by $(2k/\gamma)^{2\gamma}$ in Theorem 2.18, whose proof is in Section 2.4.2, and then conclude a mimicking network size of $O(k(2k/\gamma)^{4\gamma})$ in Corollary 2.19.

Theorem 2.18. *In every planar k-terminal network G with $\gamma = \gamma(G)$, the number of different elementary cutsets is $|\mathcal{T}_e(G)| \leq 2^{2\gamma}\,(\Pi_{i=1}^{\gamma} k_i^2) \leq (2k/\gamma)^{2\gamma}$.*

Corollary 2.19. *Every planar k-terminal network G with $\gamma = \gamma(G)$ admits a mimicking network of size $s = O\left(k(2k/\gamma)^{4\gamma}\right)$.*

Proof. Apply Theorem 2.11, using $\alpha = O(k)$ from Lemma 2.4 and $|\mathcal{T}_e(G)| \leq (2k/\gamma)^{2\gamma}$ from Theorem 2.18. □

All terminals are on γ faces - Second bound. Our second (and improved) result for the general case $\gamma \geq 1$ follows by a refined analysis of the elementary cutsets. While our bound of $(2k)^{2\gamma}$ on the number of elementary cutsets is tight, it leads to a wasteful mimicking network size (for example, plugging the worst-case $\gamma = k$ into Corollary 2.19 is inferior to the bound in Corollary 2.12). The reason is that this approach over-counts edges of the mimicking network, and we therefore devise a new proof strategy that decomposes each elementary cutset even further, in a special way that lets us to count the underlying fragments (special subsets of edges) without repetitions. We remark that the actual proof works in the dual graph G^*, and decomposes a simple cycle into (special) paths.

Theorem 2.20 (Further Decomposition of Elementary Cutsets)**.** *Every planar k-terminal network G with $k_1, \ldots, k_\gamma$ as above, has $p = 2^\gamma\left(1 + \sum_{i,j=1}^{\gamma} k_i k_j\right) \leq O(2^\gamma k^2)$ subsets of edges $E_1, \ldots, E_p \subset E$, such that every elementary cutset in G can be decomposed into a disjoint union of some of these E_i's, and each of E_i contains exactly 2 edges from the boundaries of the faces $f_1, \ldots, f_\gamma$.*

We prove this theorem in Section 2.4.3. The main difficulty is to define subsets of edges that are contained in elementary cutsets and are also easy to identify. We implement this identification by attaching to every such subset a three-part label. We prove that each label is unique, and count the number of different possible labels, which obviously bounds the number of such "special" subsets of edges.

Corollary 2.21. *Every planar k-terminal network G with $\gamma = \gamma(G)$ admits a minor mimicking network of size $s = O(\gamma 2^{2\gamma} k^4)$.*

A slightly weaker bound of $O(2^{2\gamma} k^5)$ on the mimicking network size follows easily from Theorem 2.11 by replacing elementary cutsets with our "special" subsets of edges.

To this end, it is easy to verify that all arguments about elementary cutsets hold also for the "special" subsets of edges. This includes the bound $\alpha = O(k)$, because if every two elementary cutsets intersect at most $O(k)$ times, then certainly every two "special" subsets (which are subsets of elementary cutsets) intersect at most $O(k)$ times. We can thus apply Theorem 2.11 with $\alpha = O(k)$ and "replacing" $|\mathcal{T}_e(G)|$ with $O(2^\gamma k^2)$ that we have by Theorem 2.20. The stronger bound in Corollary 2.21 follows by showing that $\alpha = O(\gamma)$ for "special" subsets of edges.

Proof of Corollary 2.21. Given a planar k-terminal network G with $\gamma(G) = \gamma$, use Theorem 2.20 to decompose the elementary terminal cutsets of G into $p = O(2^\gamma k^2)$ subsets of edges $E_1, \ldots, E_p$ as stated above. Since each of E_i has exactly two edges from the boundaries of the faces $f_1, \ldots, f_\gamma$, then for every E_i and E_j there are at most $O(\gamma)$ connected components in $G \setminus (E_i \cup E_j)$ that contain terminals. Let E_S and $E_{S'}$ be elementary cutsets of G such that $E_i \subseteq E_S$ and $E_j \subseteq E_{S'}$. By Lemma 2.4, each connected component in $G \setminus (E_S \cup E_{S'})$ must contain at least one terminal. Thus, each connected component in $G \setminus (E_i \cup E_j)$ must contain at least one terminal, which bound the number of its connected components by $O(\gamma)$. Apply Theorem 2.11 with $|\mathcal{T}_e(G)| = O(2^\gamma k^2)$ and $\alpha = O(\gamma)$ and the corollary follows. $\square$

2.4.1 Proof of Theorem 2.15 and Lemma 2.16

In this section we prove Theorem 2.15, which bounds the number of elementary cutsets when $\gamma = 1$. We start with a few definitions and lemmas. Let $G = (V, E, T, c)$ be a connected planar k-terminal network, such that the terminals $t_1, \ldots, t_k$ are all on the same face in that order. Assume without loss of generality that this special face is the outerface f_∞. We refer to this outerface as a clockwise-ordered cycle $\langle v_1^\infty, v_2^\infty, \ldots, v_l^\infty \rangle$, such that for every two terminals t_i, t_j if $v_x^\infty = t_i$ and $v_y^\infty = t_j$ then $i < j$ if and only if $x < y$.

Definition 2.22. *An* interval *of f_∞ is a subpath $\hat{I} = \langle v_i^\infty, v_{i+1}^\infty, \ldots, v_j^\infty \rangle$ if $i \leq j$ and in the case where $i > j$ $\hat{I} = \langle v_i^\infty, \ldots, v_l^\infty, v_1^\infty, \ldots, v_j^\infty \rangle$. Denote its vertices by $V(\hat{I})$ or, slightly abusing notation, simply by $\hat{I}$.*

Two trivial cases are a single vertex $\langle v_i^\infty \rangle$ if $i = j$, and the entire outerface cycle f_∞ if $i = j - 1$.

Definition 2.23. *Given $W \subseteq V$, an interval $\hat{I} = \langle v_i^\infty, v_{i+1}^\infty, \ldots, v_j^\infty \rangle$ is called* maximal with respect to W, *if $V(\hat{I}) \subseteq W$ and no interval in W strictly contains $\hat{I}$, i.e. $v_{i-1}^\infty, v_{j+1}^\infty \notin W$. Let $\mathcal{I}(W)$ be the set of all maximal intervals with respect to W, and let the* order *of $W \subseteq V$ be $|\mathcal{I}(W)|$.*

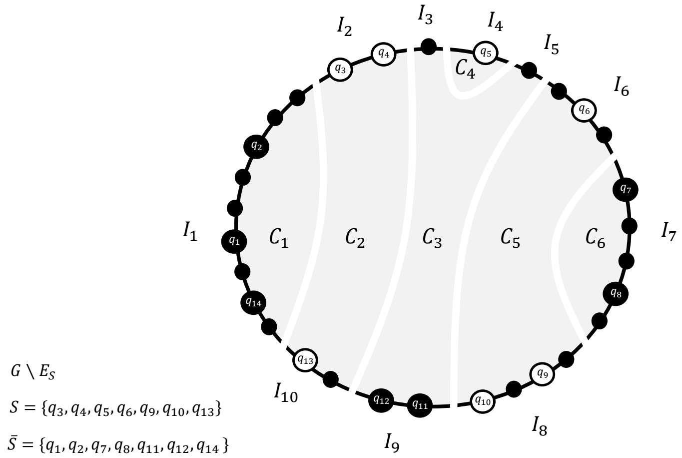


Figure 1: A minimum cutset E_S that partitions the graph G into 6 connected components $CC(E_S) = \{C_1, \ldots, C_6\}$, and partitions the outerface vertices into 10 intervals $\mathcal{I}(CC(E_S)) = \{I_1, \ldots, I_{10}\}$. Large nodes represent terminals, colored according to whether they lie in S or $\bar{S}$.

Observe that $\mathcal{I}(W)$ is a unique partition of $W \cap V(f_\infty)$, hence the order of W is well defined. Later on, we apply Definition 2.23 to connected components $C \in CC(E_S)$, instead of arbitrary subsets $W \subseteq V$. For example, in Figure 1, $\mathcal{I}(C_3) = \{I_3, I_5, I_9\}$, and the order of C_3 is $|\mathcal{I}(C_3)| = 3$.

Lemma 2.24. *For every subset $S \subset T$, $\uplus_{C \in CC(E_S)} \mathcal{I}(C)$ is a partition of $V(f_\infty)$.*

Proof. Fix $S \subset T$ and its minimum cutset E_S. Then $CC(E_S)$ is a partition of the vertices V into connected components. It induces a partition also of $V(f_\infty)$, i.e $V(f_\infty) = \uplus_{C \in CC(E_S)} \big(C \cap V(f_\infty)\big)$. By Definition 2.23, each $C \cap V(f_\infty)$ can be further partitioned into maximal intervals, given by $\mathcal{I}(C)$. Combining all these partitions, and the lemma follows. See Figure 1 for illustration. $\square$

Lemma 2.25. *For every $S \subset T$ and its minimum cutset E_S, if $C \in CC(E_S)$ is an elementary connected component in G, then $|I(C)| = 1$.*

Proof. Since C is elementary, there are exactly two connected components C and C' in $CC(\delta(C))$. By Lemma 2.4 each of C and C' contains at least one terminal. Since all the terminals are on the outerface, each of C and C' contains at least one interval. Assume toward contradiction that C contains at least two maximal intervals I_1 and I_3, then there must be at least two intervals I_2 and I_4 in C' that appear on the outerface in an alternating order, i.e. I_1, I_2, I_3, I_4. Let v_i be a vertex in the interval I_i, and denote by P_{13} and P_{24} a path that connects between v_1, v_3 and between

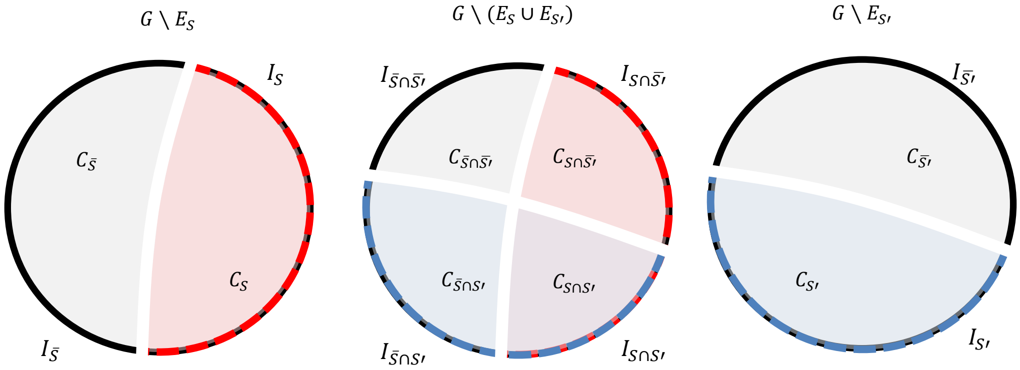


Figure 2: The union of two elementary cutsets, $E_S \cup E_{S'}$, disconnects G into (at most) 4 connected components, and the outerface into 4 intervals.

---

v_2, v_4 correspondingly. Note that P_{13} is contained in C and P_{24} is contained in C'. Moreover note that these two paths must intersect each other, giving a contradiction, and the lemma follows. □

We are ready to prove Theorem 2.15. Recall that for every $S \subset T$, if E_S is an elementary cutset then by Lemma 2.25 each of S and $\bar{S}$ must be a single interval. Hence they must be of the form $\{t_i, t_{i+1}, \dots, t_j\}$ and $\{t_{j+1}, \dots, t_{i-1}\}$ allowing wrap-around. Thus, we can characterize S and $\bar{S}$ by the pairs (i, j) and $(j+1, i-1)$ respectively. There are at most $k(k-1)$ such different pairs, since $S \neq T$ and thus $j \neq i-1$. By the symmetry between S and $\bar{S}$, we should divide that number by 2 and Theorem 2.15 follows.

Proof of Lemma 2.16. Let E_S and $E_{S'}$ be two elementary minimum cutsets, and let C_S and $C_{\bar{S}}$ be the two elementary connected components in $CC(E_S)$. By Lemma 2.25, each of C_S and $C_{\bar{S}}$ contains exactly one maximal interval denoted by I_S and $I_{\bar{S}}$ respectively, and similarly denote $C_{S'}, C_{\bar{S}'}, I_{S'}$ and $I_{\bar{S}'}$ for $E_{S'}$. Since each of the cutsets E_S and $E_{S'}$ intersect the cycle of the outerface in exactly two edges, the cutset $E_S \cup E_{S'}$ intersects the cycle of the outerface in at most 4 edges. Therefore the graph $G \setminus (E_S \cup E_{S'})$ has at most 4 maximal intervals. By Lemma 2.4, every connected component in $CC(E_S \cup E_{S'})$ must contains at least one terminal. Since all the terminals lie on the outerface, any connected component that contains terminal must contains also an interval. Every interval is contained in exactly one connected component. Thus, there are at most 4 connected components in $CC(E_S \cup E_{S'})$, and the lemma follows. See Figure 2 for illustration. □

2.4.2 Proof of Theorem 2.18

In this section we prove Theorem 2.18, which bounds the number of elementary cutsets when $\gamma > 1$. Since we assume (by perturbation) that there is a one-to-one correspondence between $S \subseteq T$ and E_S, it suffices to bound the number of different ways that an elementary cutset can partition the terminals into S and $\bar{S}$. We achieve the latter by two observations, which are extensions of the ideas in Theorem 2.15. First, an elementary cutset can break each of the γ faces into at most two paths, which overall splits the terminals into at most 2γ subsets. As each subset (path) can lie either in S or in $\bar{S}$, there are at most $2^{2\gamma}$ different ways to partition T into S and $\bar{S}$ (this bound includes cases where two paths from the same face lie both in S or both in $\bar{S}$, which is equivalent to not breaking the face into two paths). Second, there are $\binom{k_i}{2}$ ways that the face f_i can be broken into 2 paths by elementary cutsets, which gives overall $\Pi_{i=1}^{\gamma}\binom{k_i}{2}$ ways to break all the γ faces simultaneously. Combining these two observations leads to the required bound.

We start with a few definitions. Let $G = (V, E, F, T, c)$ be a k-terminal network with γ faces $f_1, \dots, f_\gamma$, where each f_i contains the k_i terminals T_i (breaking ties arbitrarily), where $T = \cup_{i=1}^{\gamma} T_i$ and thus $k = \sum_{i=1}^{\gamma} k_i$. Denote the terminals in T_i by $t^i_1, \dots, t^i_{k_i}$, where the order is by a clockwise order around the boundary of f_i, starting with an arbitrary terminal; for simplicity, we shall write t_j instead of t^i_j when the face f_i is clear from the context. Let G^* be the dual graph of G. The graph G^* has k terminal faces $\{f_{t^i_j}\}$ that are dual to the terminals $\{t^i_j\}$ of G, and has γ special vertices $W = \{w_1, \dots, w_\gamma\}$ that are dual to the faces $f_1, \dots, f_\gamma$ of G (see Section 2.7 for basic notions of planar duality).

We label each $S \in \mathcal{T}_e(G)$ (and its elementary cycle E^*_S) by two vectors $\bar{x}, \bar{y}$, as follows. Since E^*_S is a simple cycle, it visits every vertex $w_i \in W$ at most once. If it does visit w_i, then exactly two cycle edges are incident to w_i. and these two edges naturally partition the faces around w_i into two subsets. Moreover, each subset appears as a contiguous subsequence if the faces around w_i are scanned in a clockwise order. In particular, the terminal faces $f_{t_1}, \dots, f_{t_{k_i}}$ are partitioned into two subsets, whose indices can be written as $\{t_{x_i}, \dots, t_{y_i-1}\}$ and $\{t_{y_i}, \dots, t_{x_i-1}\}$, for some $x_i, y_i \in [k_i]$, under the two conventions: (i) we allow wraparound, i.e., $t_{k_i+1} = t_1$ and so forth; (ii) if $x_i = y_i$, then we have a trivial partition of T_i, where one subset is T_i and the other is $\emptyset$. Observe that one of these subsets is contained in S and the other in $\bar{S}$, thus we can assume that $\{t_{x_i}, \dots, t_{y_i-1}\} \subseteq S$ and $\{t_{y_i}, \dots, t_{x_i-1}\} \subset \bar{S}$. If the cycle E^*_S does not visit w_i, then we simply define $x_i = y_i = 1$, which represents a trivial partitioning of T_i. The labels are now defined as $\bar{x} = (x_1, \dots, x_\gamma)$ and $\bar{y} = (y_1, \dots, y_\gamma)$.

We now claim that G^* has at most $2^{2\gamma}$ elementary cycles with the same label $(\bar{x}, \bar{y})$. To see this, fix $\bar{x}, \bar{y} \in [k_1] \times \dots \times [k_\gamma]$ and modify G^* into a plane graph $G^*_{\bar{x},\bar{y}}$ with at most 2γ terminal faces, as follows. For every w_i, create a single terminal face

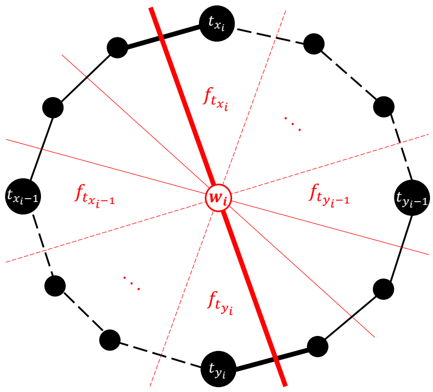


Figure 3: A simple cycle E_S^* separates the faces around w_i into two subsets. The primal graph G is shown in black, and its dual G^* in red. Thicker lines are used for edges of the elementary cutset E_S and of the cycle E_S^*. Dashed lines represent dual edges that are removed by merging faces, and primal edges that are contracted.

$f^i_{x_i}$ by "merging" faces around w_i, starting from $f_{t_{x_i}}$ and going in a clockwise order until $f_{t_{y_i-1}}$ (inclusive). Then merge similarly the faces from $f_{t_{y_i}}$ and until $f_{t_{x_i-1}}$ into a single terminal face $f^i_{y_i}$. If $x_i = y_i$, then the two merging operations above are identical, and thus (as an exception) create only one terminal face denoted $f^i_{x_i}$. Formally, a merge of two faces is implemented by removing the edge incident to w_i that goes between the relevant faces. Observe that removing these edges in G^* can be described in G as contracting the path around the boundary of the face f_i from the terminal t_{x_i} to t_{y_i-1}, and similarly from the terminal t_{y_i} to t_{x_i-1}, see Figure 3. It is easy to verify that the modified graph $G^*_{\bar{x},\bar{y}}$ is planar, and that every elementary cycle E_S^* in G^* with this label $(\bar{x}, \bar{y})$ is also an elementary cycle in $G^*_{\bar{x},\bar{y}}$ that separates the new terminal faces in a certain way. Usually, the new terminal faces are separated into $\{f^i_{x_i}\}_{i=1}^{\gamma}$ and $\{f^i_{y_i}\}_{i=1}^{\gamma}$, except that when $x_i = y_i$, we have only one new terminal face $f^i_{x_i}$, which should possibly be included with the y_i's instead of with the x_i'. Since $G^*_{\bar{x},\bar{y}}$ has at most 2γ terminal faces, it can have at most $2^{2\gamma}$ elementary cycles (one for each subset). This shows that for every label $(\bar{x}, \bar{y})$, there are at most $2^{2\gamma}$ different elementary cycles in G^*, as claimed.

Finally, the number of distinct labels $(\bar{x}, \bar{y})$ is clearly bounded by $\Pi_{i=1}^{\gamma} k_i^2$ and the above claim applies to each of them. By the inequality of arithmetic and geometric means $\Pi_{i=1}^{\gamma} k_i^2 \le (k/\gamma)^{2\gamma}$. Therefore, the total number of different elementary cycles in G^* is at most $(2k/\gamma)^{2\gamma}$, and Theorem 2.18 follows.

2.4.3 Proof of Theorem 2.20

In this section we prove Theorem 2.20, which actually decompose the elementary cutsets in a "bounded manner" when $\gamma > 1$. The idea is to consider the dual graph, which has γ special vertices, and elementary cycles. Since every elementary cycle is a simple cycle, it visits each of the γ vertices at most once, and thus we can decompose the elementary cycles into paths, such that the two endpoints of every path belong to the γ vertices. The challenging part is to count how many distinct paths are there.

We shall use the notation introduced in the beginning of Section 2.4.2. In particular, the graph G has terminals $T = \cup_{i=1}^{\gamma} T_i$, where $T_i = \{t_1^i, \dots, t_{k_i}^i\}$ are the terminals on the boundary of special face f_i, and for simplicity we omit i when it is clear from the context. The dual graph, denoted G^* has terminal faces $\{f_{t_j^i}\}$ and special vertices $W = \{w_1, \dots, w_\gamma\}$. Let $v_\infty \in V$ be the vertex whose dual face f_{v_∞} is the outerface of G^*.

Informally, the next definition determines whether f_v, the face dual to a vertex $v \in V$, lies "inside" or "outside" a circuit M^* in G^*. It works by counting how many times a path from v to v_∞ "crosses" M^* and evaluating it modulo 2 (i.e., its parity). The formal definition is more technical because it involves fixing a path, but the ensuing claim shows the value is actually independent of the path. Moreover, we need to properly define a "crossing" between a path Φ in G and a circuit in G^*; to this end, we view the path Φ as a sequence of faces in G^*, that goes from f_v to f_{v_∞} and at each step "crosses" an edge of G^*.

Definition 2.26 (Parity of a dual face). *Let f_v be the dual face to a vertex $v \in V$, and fix a simple path in G between v and v_∞, denoted Φ. Let M^* be a circuit in G^*, and observe that its edges $E(M^*)$ form a multiset. Define the* parity *of f_v with respect to M^* to be*

$$\mathrm{Par}(f_v, M^*) := \Big(\sum_{e \in E(\Phi)} \mathrm{Count}(e^*, E(M^*)) \Big) \bmod 2,$$

where $\mathrm{Count}(a, A)$ is the number of times an element a appears in a multiset A.

The next claim justifies the omission of the path Φ in the notation $\mathrm{Par}(f_v, M^*)$.

Claim 2.27. *Fix $v \in V$, and let Φ and Φ' be two paths in G between v and v_∞. Then for every circuit M^* in G^*,*

$$\sum_{e \in E(\Phi)} \mathrm{Count}(e^*, E(M^*)) = \sum_{e \in E(\Phi')} \mathrm{Count}(e^*, E(M^*)) \pmod 2.$$

Proof. Fix a vertex $v \in V$ and its dual face f_v. Fix also a circuit M^*, and a decomposition of it into simple cycles. We say that a simple cycle in G^* (like one from the decomposition of M^*) *contains* the face f_v if that cycle separates f_v from the outerface f_{v_∞}. Let Φ be a path between v and v_∞. By the Jordan Curve Theorem, the path's dual edges $\{e^* : e \in E(\Phi)\}$ intersect a simple cycle in G^* an odd number of times if and only if that simple cycle contains the dual face f_v. By summing this quantity over the simple cycles in the decomposition of M^*, we get that

$$\sum_{e \in E(\Phi)} \text{Count}(e^*, E(M^*)) = 1 \pmod 2$$

if and only if f_v is contained in an odd number of these simple cycles. The latter is clearly independent of the path Φ, which proves the claim. $\square$

Given a circuit M^* in G^*, we use the above definition to partition the terminals T into two sets according to their parity, namely,

$$T_{odd}(M^*) \stackrel{\text{def}}{=} \{t \in T : \text{ Par}(f_t, M^*) = 1\},$$
$$T_{even}(M^*) \stackrel{\text{def}}{=} \{t \in T : \text{ Par}(f_t, M^*) = 0\}.$$

Given $S \in \mathcal{T}_e(G)$, recall that E_S^* is the shortest cycle which is S-separating in G^* (i.e. it separates between the terminal faces S^* and $\bar{S}^*$). Since E_S^* is an elementary cycle, it separates the plane into exactly two regions, which implies, without loss of generality , $T_{odd}(E_S^*) = S$ and $T_{even}(E_S^*) = \bar{S}$. Moreover, E_S^* is a simple cycle and thus goes through every vertex of W at most once. We decompose E_S^* into $|W \cap V(E_S^*)|$ paths in the obvious way, where the two endpoints of each path, and only them, are in W, and we let Π_S denote this collection of paths in G^*. There are two exceptional cases here; first, if $|W \cap V(E_S^*)| = 1$ then we let Π_S contain one path whose two endpoints are the same vertex (so actually a simple cycle). second, if $|W \cap V(E_S^*)| = 0$ then we let $\Pi_S = \emptyset$ (we will deal with this case separately later). Now define the set

$$\Pi \stackrel{\text{def}}{=} \bigcup_{S \in \mathcal{T}_e(G)} \Pi_S$$

be the collection of all the paths that are obtained in this way over all possible $S \in \mathcal{T}_e(G)$. Notice that if the same path is contained in multiple sets Π_S, then it is included in the set Π only once (in fact, this "overlap" is what we are trying to leverage).

Now give to each path $P \in \Pi$ a label that consists of three parts: (1) the two endpoints of P, say $w_i, w_j \in W$; (2) the two successive terminals on each of the faces f_i and f_j, which describe where the path P enters vertices w_i and w_j, say between t_{x-1}^i, t_x^i and between t_y^j, t_{y+1}^j; and (3) the set $T_{odd}(P \cup \Pi_{ij})$, where Π_{ij} is the shortest path (or any other fixed path) that agrees with parts (1) and (2) of the label and does not go through W, i.e., the shortest path between w_i and w_j that enters them between t^i_{x-1}, t^i_x and t^j_y, t^j_{y+1} and does not go through any other vertex in W. This includes the exceptional case $i = j$, in which P is actually a simple cycle.

We proceed to show that each label is given to at most one path in Π (which will be used to bound $|\Pi|$). Assume toward contradiction that two different paths $P, P' \in \Pi$ get the same label, and suppose $c(P') < c(P)$. Suppose P is the path between w_i to w_j in E^*_S for $S \in T_e(G)$, and P' is the path between the same endpoints (because of the same label) in $E^*_{S'}$ for another $S' \in T_e(G)$. By construction, the paths P and P' are simple, because E^*_S and $E^*_{S'}$ are elementary cycles, and only their endpoint vertices are from W.

The key to arriving at a contradiction is the next lemma. In these proofs, a path P is viewed as a multiset of edges $E(P)$, and the union and subtraction operations are applied to multisets. In particular, the union of two paths with the same endpoints gives a circuit.

Lemma 2.28. *The circuit $(E^*_S \setminus P) \cup P'$ is S-separating.*

To prove this lemma, we will need the following two claims.

Claim 2.29. *Let A, B and C be (the edge sets of) simple paths in G^* between the same $w_i, w_j \in W$. Then*

$$\forall t \in T, \qquad \mathrm{Par}(f_t, A \cup C) = \mathrm{Par}(f_t, A \cup B) + \mathrm{Par}(f_t, B \cup C) \pmod 2.$$

Proof. Fix $t \in T$ and a path Φ between t and v_∞. Since A, B and C are simple paths,

$$\sum_{e \in E(\Phi)} \mathrm{Count}(e^*, A \cup C) = |E^*(\Phi) \cap A| + |E^*(\Phi) \cap C|,$$
$$\sum_{e \in E(\Phi)} \mathrm{Count}(e^*, A \cup B) = |E^*(\Phi) \cap A| + |E^*(\Phi) \cap B|,$$
$$\sum_{e \in E(\Phi)} \mathrm{Count}(e^*, B \cup C) = |E^*(\Phi) \cap B| + |E^*(\Phi) \cap C|.$$

Summing the three equations above modulo 2 yields

$$\mathrm{Par}(f_t, A \cup C) + \mathrm{Par}(f_t, A \cup B) + \mathrm{Par}(f_t, B \cup C) = 0 \pmod 2,$$

which proves the claim. $\square$

Claim 2.30. *Let $\bigtriangleup$ be the symmetric difference between two sets. For every 3 simple paths A, B and C between $w_i, w_j \in W$,*

$$T_{odd}(A \cup C) = T_{odd}(A \cup B) \bigtriangleup T_{odd}(B \cup C).$$

Proof. Observe that $T_{odd}(A \cup B) \bigtriangleup T_{odd}(B \cup C)$ contains all $t \in T$ for which exactly one of $\mathrm{Par}(f_t, A \cup B)$ and $\mathrm{Par}(f_t, B \cup C)$ is equal to 1, which by Claim 2.29 is equivalent to having $\mathrm{Par}(f_t, A \cup C) = 1$. □

Proof of Lemma 2.28. To set up some notation, let $Q \stackrel{\text{def}}{=} E_S^* \setminus P$ be a simple path between w_i and w_j. Since E_S^* is a simple cycle that contains P, we can write $E_S^* = Q \cup P$.

The idea is to swap the path P in E_S^* with the other path P', which for sake of analysis is implemented in two steps. The first step replace P (in E_S^*) with Π_{ij}, which gives the circuit $(E_S^* \setminus P) \cup \Pi_{ij} = Q \cup \Pi_{ij}$. The second step replaces Π_{ij} with P', which results with the circuit $Q \cup P' = (E_S^* \setminus P) \cup P'$. Now apply Claim 2.30 twice, once to the simple paths $A = Q$, $B = P$ and $C = \Pi_{ij}$, and once to the simple paths $A = Q$, $B = \Pi_{ij}$ and $C = P'$, we get that

$$T_{odd}(Q \cup \Pi_{ij}) = T_{odd}(E_S^*) \bigtriangleup T_{odd}(P \cup \Pi_{ij}),$$
$$T_{odd}(Q \cup P') = T_{odd}(Q \cup \Pi_{ij}) \bigtriangleup T_{odd}(\Pi_{ij} \cup P').$$

By plugging the first equality above into the second one, and observing that $T_{odd}(\Pi_{ij} \cup P) = T_{odd}(\Pi_{ij} \cup P')$ because P and P' have the same label, we obtain that

$$T_{odd}(Q \cup P') = T_{odd}(E_S^*). \tag{6}$$

Finally, it is easy to verify that the circuit $Q \cup P'$ must separate between $T_{odd}(Q \cup P')$ and $T_{even}(Q \cup P')$. Using (6) and the fact that E_S^* is an elementary cycle, we know that $T_{odd}(Q \cup P') = T_{odd}(E_S^*) = S$, and thus $T_{even}(Q \cup P') = T \setminus S$. It follows that $Q \cup P'$ is S-separating, as required. □

Lemma 2.28 shows that the circuit $(E_S^* \setminus P) \cup P'$ is S-separating, while also having lower cost than E_S^*. This contradicts the minimality of E_S^*, and shows that the paths in Π have distinct labels. Thus, $|\Pi|$ is at most the number of distinct labels, and we will bound the latter using the following claim.

Claim 2.31. *Let $P \in \Pi$ be a path between w_i and w_j, and let $r \in [\gamma]$. Then*

$$\forall t, t' \in T_r, \qquad \mathrm{Par}(f_t, P \cup \Pi_{ij}) = \mathrm{Par}(f_{t'}, P \cup \Pi_{ij}),$$

where Π_{ij} is the shortest path with the same parts (1) and (2) of the label as P, and does not go through any other vertices of W.

Proof. Since $t, t' \in T_r$, their dual faces f_t and $f_{t'}$ share w_r on their boundary. P and Π_{ij} are simple paths in G^* with the same endpoints, and thus $P \cup \Pi_{ij}$ is a circuit in

G^*, which by construction does not go through any vertex w_r with $r \neq i, j$. Fix a path Φ in G between t and v_∞. We can extend it into a path Φ' between t' and v_∞, by taking a path $A_{t't}$ in G that goes around the face f_r between t' and t (both are on the face f_r, because $t, t' \in T_r$), and letting $\Phi' \stackrel{\text{def}}{=} A_{t't} \cup \Phi$.

Since P and Π_{ij} agree on the same parts (1) and (2) of the label, then $P \cup \Pi_{ij}$ have exactly two edges between some two successive terminals on each of the faces f_i and f_j. Thus, if $r \neq i, j$ then $|A_{t't} \cap (P \cup \Pi_{ij})| = 0$. If $r = i$ or $r = j$ but $i \neq j$ then $|A_{t't} \cap (P \cup \Pi_{ij})|$ is either 0 or 2. And if $r = i = j$ then $|A_{t't} \cap (P \cup \Pi_{ij})|$ is either 0, 2 or 4. Therefore, if we examine the parities of f_t and $f_{t'}$ with respect to $P \cup \Pi_{ij}$ using the paths Φ and $\Phi' = A_{t't} \cup \Phi$, respectively, we conclude that these parities are equal, as required. $\square$

We can now bound the number of possible labels of a path $P \in \Pi$. There are γ^2 possibilities for part 1 of the label, i.e., the endpoints $w_i, w_j \in W$ of P (note that we may have $i = j$). Given this data, there are $k_i k_j$ possibilities for part 2, i.e., between which two terminals the path P exits w_i and enters w_j. Furthermore, the number of possibilities for part 3 is the number of different subsets $T_{odd}(P \cup \Pi_{ij})$. By Claim 2.31 for every $r \in [\gamma]$ either $T_r \subseteq T_{odd}(P \cup \Pi_{ij})$ or $T_r \cap T_{odd}(P \cup \Pi_{ij}) = \emptyset$. Thus, the number of different subsets $T_{odd}(P \cup \Pi_{ij})$ is the number of different subsets of $\{T_1, \ldots, T_\gamma\}$, which is at most 2^γ. Altogether we get that there are at most $2^\gamma \sum_{i,j=1}^{\gamma} k_i \cdot k_j$ different labels.

Finally, there are also cycles E_S^* for $S \in \mathcal{T}_e(G)$ that do not go through any vertices of W, i.e. $W \cap V(E_S^*) = \emptyset$. Thus, they are not include in Π, so we count them now separately. Recall that without loss of generality $T_{odd}(E_S^*) = S$, i.e every such cycle E_S^* is identified uniquely by a different subset $T_{odd}(\cdot)$. Since by Claim 2.31 there are at most 2^γ such subsets, we get that there are at most 2^γ such cycles. Adding them to our calculation, and Theorem 2.20 follows.

2.4.4 Flow Sparsifiers

Okamura and Seymour [OS81] proved that in every planar network with $\gamma(G) = 1$, the flow-cut gap is 1 (as usual, flow refers here to multicommodity flow between terminals). It follows immediately, see e.g. [AGK14], that for such a graph G, every (q, s)-cut-sparsifier is itself also a (q, s)-flow-sparsifier of G. Thus, Corollary 2.17 implies the following.

Corollary 2.32. *Every planar k-terminal network G with $\gamma = 1$ admits a minor $(1, O(k^4))$-flow-sparsifier.*

Chekuri, Shepherd, and Weibel [CSW13, Theorem 4.13] proved that in every planar

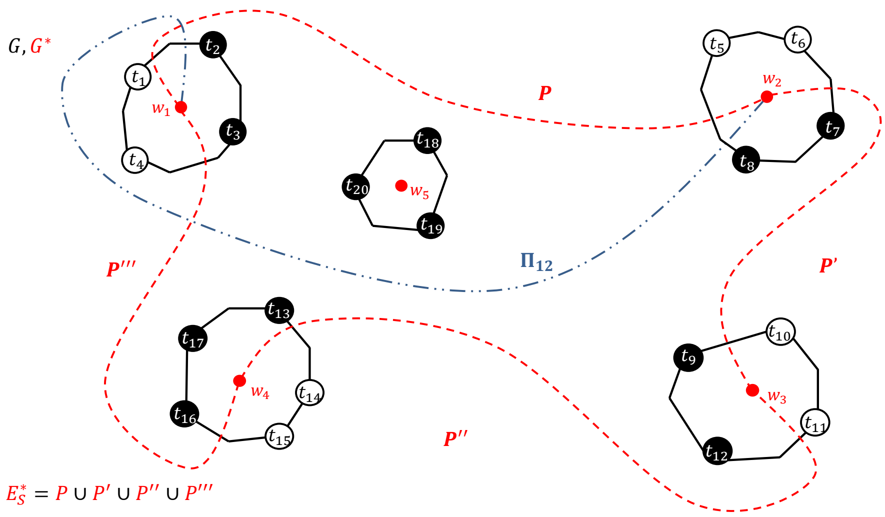


Figure 4: A planar 20-terminal network with $\gamma = 5$. Let $S \subset T$ be all the black terminals, then E_S^* (red dashed line) is split into 4 paths P, P', P'', P'''. The label of P, for example, is (1) w_1 and w_2; (2) t_1, t_2 and t_5, t_8; (3) $T_{odd}(P \cup \Pi_{12}) = \{t_1, t_2, t_3, t_4, t_{18}, t_{19}, t_{20}\}$, and is computed using Π_{12} (blue dashed line).

---

network G, the flow-cut gap is at most $3\gamma(G)$, and thus Corollary 2.21 implies the following.

Corollary 2.33. *Every planar k-terminal network G with $\gamma = \gamma(G)$ admits a minor $(3\gamma, O(\gamma 2^{2\gamma} k^4))$-flow-sparsifier.*

2.5 Terminal-Cuts Scheme

In this section we present applications of our results in Section 2.2 to data structures that store all the minimum terminal cuts in a graph G. As our focus is on the data structure's memory requirement, we do not discuss its query time. We start with a formal definition of such a data structure, and then provide our bounds of $\tilde{O}(|\mathcal{T}_e|)$ bits for general graphs (Theorem 2.35), and $\tilde{O}(2^\gamma k^2)$ bits for planar graphs (Corollaries 2.36 and 2.37). In comparison, a trivial data structure for general graphs uses $\tilde{O}(2^k)$ bits, by storing the cost of all the terminal mincuts explicitly.

Definition 2.34. *A* terminal-cuts scheme (TC-scheme) *is a data structure that uses a storage (memory) M to support the following two operations on a k-terminal network $G = (V, E, T, c)$, where $n = |V|$ and $c : E \to \{1, \ldots, n^{O(1)}\}$.*

1. *Preprocessing, denoted P, which gets as input the network G and builds M.*

2. *Query, denoted R, which gets as input a subset of terminals S, and uses M (without access to G) to output the cost of the minimum cutset E_S.*

We usually assume a machine word size of $O(\log n)$ bits, because even if G has only unit-weight edges, the cost of a cut might be $O(n^2)$, which is not bounded in terms of k.

Theorem 2.35. *Every k-terminal network $G = (V, E, T, c)$ admits a TC-scheme with storage size of $O(|\mathcal{T}_e(G)|(k + \log n))$ bits, where $\mathcal{T}_e(G)$ is the set of elementary cutsets in G.*

Proof. We construct a TC-scheme as follows. In the preprocessing stage, given G, the TC-scheme stores $\langle S, c(E_S)\rangle$ for every $S \in \mathcal{T}_e(G)$, where S is written using k bits. The cost of every cutset is at most $|E| \cdot n^{O(1)} = \mathrm{poly}(n)$, and thus the storage size of the TC-scheme is $O(|\mathcal{T}_e(G)|(k + \log n))$ bits, as required. Now given a subset $S \subset T$, the query operation $R(S; P(G))$ outputs

$$\min\Big\{\sum_{S' \in \varphi} \mathrm{mincut}_G(S') : \ \varphi \subseteq \mathcal{T}_e(G) \text{ s.t. } \bigcup_{S'\in\varphi} \mathrm{argmincut}_G(S') \text{ is } S\text{-separating in } G\Big\}. \tag{7}$$

Since for every $\varphi \subseteq 2^T$, the cutset $\cup_{S'\in\varphi} \mathrm{argmincut}_G(S')$ is S-separating in G if and only if $|\mathbb{1}_S(t_i) - \mathbb{1}_S(t_j)| \leq \sum_{S'\in\varphi} |\mathbb{1}_{S'}(t_i) - \mathbb{1}_{S'}(t_j)|$ for all $i, j \in [k]$, the calculation in (7) can be done with no access to G. Clearly, $\mathrm{mincut}_G(S) \leq R(S; P(G))$. By Theorem 2.9, there is $\varphi \subseteq \mathcal{T}_e(G)$ such that $\mathrm{argmincut}_G(S) = \uplus_{S'\in\varphi} \mathrm{argmincut}_G(S')$ and $\mathrm{mincut}_G(S) = \sum_{S'\in\varphi} \mathrm{mincut}_G(S')$. Thus, $R(S'; P(G)) = \mathrm{mincut}_G(S')$. □

Corollary 2.36. *Every planar k-terminal network G with $\gamma = 1$ admits a TC-scheme with storage size of $O(k^2 \log n)$ bits, i.e., $O(k^2)$ words.*

Proof. If G is a planar k-terminal network with $\gamma = 1$, then by Theorem 2.15 every $S \in \mathcal{T}_e(G)$ is equal to $\{t_i, t_{i+1}, \dots, t_j\}$ for some $i, j \in [k]$ and $|\mathcal{T}_e(G)| = \binom{k}{2}$ (recall that all the terminals $t_1, \dots, t_k$ are on the outerfaces of G in order). Thus, we can specify S via these two indices i and j, using only $O(\log k) \leq O(\log n)$ bits (instead of k). The storage bound follows. □

Theorem 2.37. *Every planar k-terminal network G with $\gamma = \gamma(G)$ admits a TC-scheme with storage size of $O\Big(2^\gamma\big(1 + \sum_{i,j=1}^{\gamma} k_i \cdot k_j\big)\big(\gamma + \log n\big)\Big) \leq O(2^\gamma k^2(\gamma + \log n))$ bits.*

Proof sketch. If G is a planar k-terminal network with bounded γ, then Theorem 2.20 characterize $2^\gamma k^2$ special subsets of edges together with some small addition information for each such subset that denote by *label*. It further prove that all the elementary cuts can be restored using only the special subsets and their labels. As each label can be stored by at most $O(\gamma)$ bits, the storage bound follows. □

2.6 Cut-Sparsifier vs. DAM in planar networks

In this section we prove the duality between cuts and distances in planar graphs with all terminals on the outerface. Although the duality between shortest cycles and minimum cuts in planar graphs is known, the main difficulty is to transform all the shortest cycles into shortest paths without blowing up the number of terminals in the graph. We prove this duality using the following two theorems, and applications of them can be found in Section 2.6.3.

Theorem 2.38. *Let $G = (V, E, T, c)$ be a planar k-terminal network with all its terminals T on the outerface. One can construct a planar k-terminal network $G' = (V', E', T', c')$ with all its terminals T' on the outerface, such that if G' admits a (q, s)-DAM then G admits a minor $(q, O(s))$-cut-sparsifier.*

Theorem 2.39. *Let $G = (V, E, T, c)$ be a planar k-terminal network with all its terminals T on the outerface. One can construct a planar k-terminal network $G' = (V', E', T', c')$ with all its terminals T' on the outerface, such that if G' admits minor (q, s)-cut-sparsifier then G admits a $(q, O(s))$-DAM.*

2.6.1 Proof of Theorem 2.38

Construction of the Reduction. The idea is to first use the duality of planar graphs in order to convert every minimum terminal cut into a shortest cycle, and then "open" every shortest cycle into a shortest path between two terminals, which in turn are preserved by a (q, s)-DAM. More formally, given a plane k-terminal network $G = (V, E, F, T, c)$ with all its terminals $T = \{t_1, \ldots, t_k\}$ on the outerface in a clockwise order, we firstly construct its dual graph G_1 where the boundaries of all its k terminal faces $T(G_1) = \{f_{t_1}, \ldots, f_{t_k}\}$ share the same vertex v_{f_∞}, and secondly we construct G_2 by the graph G_1 where the vertex v_{f_∞} is split into k different vertices $v_{f_\infty}^{i,i+1}$, and every edge (v_{f_∞}, v^*) that embedded between (or on) the two terminal faces f_{t_i} and $f_{t_{i+1}}$ in G_1 correspond to a new edge $(v_{f_\infty}^{i,i+1}, v^*)$ in G_2 with the same length. See Figure 5 from left to right for illustration, and see Section 2.7 for basic notions of planar duality. In the following, $f + f'$ denotes a new face that is the union of two faces f and f'.

$$
\begin{aligned}
V(G_2) :=& \big(V(G_1) \setminus \{v_{f_\infty}\}\big) \cup \{v_{f_\infty}^{1,2}, \ldots, v_{f_\infty}^{k-1,k}, v_{f_\infty}^{k,1}\} \\
E(G_2) :=& \big(E(G_1) \setminus \{(v_{f_\infty}, v^*) \;:\; v^* \in V(G_1)\}\big) \\
& \cup \{(v_{f_\infty}^{i,i+1}, v^*) \;:\; i \in [k], (v_{f_\infty}, v^*) \in E(G_1), v^* \text{ between } f_{t_i}, f_{t_{i+1}}\}^{3} \\
F(G_2) :=& \big(F(G_1) \setminus \{f_\infty, f_{t_1}, \ldots, f_{t_k}\}\big) \cup \{f_\infty + f_{t_1} + \ldots + f_{t_k}\} \\
T(G_2) :=& \{v_{f_\infty}^{1,2}, \ldots, v_{f_\infty}^{k-1,k}, v_{f_\infty}^{k,1}\}
\end{aligned}
$$

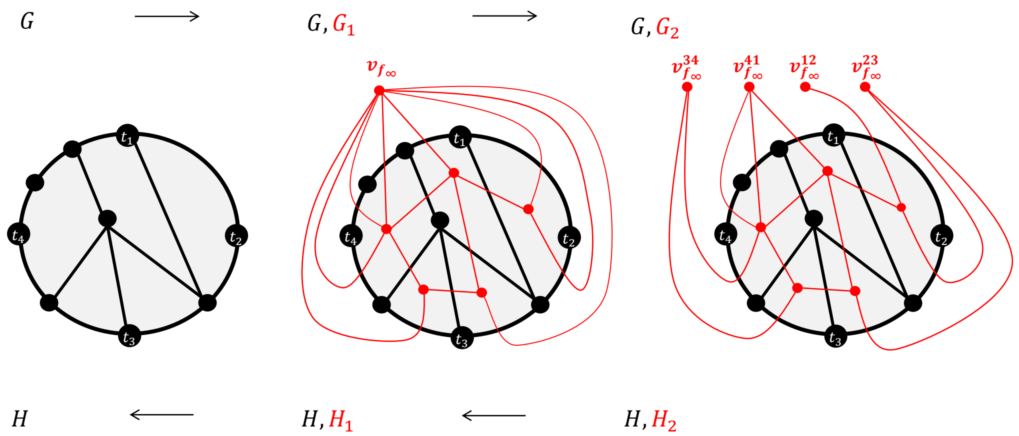


Figure 5: The first graph (in black) is the original graph G. The second is its dual graph G_1 colored in red. The third graph H and its new terminals colored in red. We get that graph by "splitting" the outerface vertex v_{f_∞} to k new vertices, which are the new terminals.

Let H_2 be an (q, s)-DAM of G_2. Since it is a minor of G_2, both are planar k-terminals network such that all their terminals are on their outerface in the same clockwise order. Hence, we can use H_2 and the same reduction above, but in reverse operations, in order to construct a $(q, O(s))$-cut-sparsifier H for G. First, we "close" all the shortest paths in H_2 into cycles by merging its k terminals $v_{f_\infty}^{i-1,i}$ into one vertex called v_{f_∞}, and denote this new graph by H_1. Note that H_1 has k new faces $f_{t_1}, \ldots, f_{t_k}$, where each face f_{t_i} was created by uniting the two terminals $v_{f_\infty}^{i-1,i}, v_{f_\infty}^{i,i+1}$ of H_2. These k new faces of H_1 will be its k terminal faces. Secondly, we argue that the dual graph of H_1 is our requested cut-sparsifier of G, which we denote by H. See Figure 5 from right to left for illustration.

Analysis of the Reduction. The key element of the reduction's proof is the duality between every shortest cycle in G_1 to a shortest path in G_2, which we formally stated in the following lemma. Given G and its dual graph $G_1 = G^*$ as stated above, for every subsets of terminals $S \subset T(G)$ we denote by $S^* \subset T(G^*)$ the corresponding set of terminal faces, i.e. $S^* = \{f_{t_i} : \forall i \in [k] \text{ s.t. } t_i \in S\}$.

Lemma 2.40. *Every shortest circuit that separates between the terminal faces $S^*_{i,j}$ and $\bar{S}^*_{i,j}$ in G_1, corresponds to a shortest path between the two terminals $v_{f_\infty}^{i-1,i}$ and $v_{f_\infty}^{j,j+1}$ in G_2, and vise versa.*

[3]We allow wraparound, i.e., $v^{k,k+1} = v^{k,1}$.

$$
\begin{aligned}
V(H_1) &:= \big(V(H_2) \setminus \{v_{f_\infty}^{1,2}, \ldots, v_{f_\infty}^{k-1,k}, v_{f_\infty}^{k,1}\}\big) \cup \{v_{f_\infty}\} \\
E(H_1) &:= \big(E(H_2) \setminus \{(v_{f_\infty}^{i,i+1}, v^*) \ : \ i \in [k]\}\big) \cup \{(v_{f_\infty}, v) \ : \ (v_{f_\infty}^{i,i+1}, v^*) \in E(H_2)\} \\
F(H_1) &:= \big(F(H_2) \setminus \{f_\infty + f_{t_1} + \ldots + f_{t_k}\}\big) \cup \{f_\infty, f_{t_1}, \ldots, f_{t_k}\} \\
T(H_1) &:= \{f_{t_1}, \ldots, f_{t_k}\}
\end{aligned}
$$

Proof. **First direction - Circuits to distances.**

Let $\mathcal{C}$ be a minimum circuit that separates between the terminal faces S^*_{ij} and $\bar{S}^*_{ij}$ in G_1 (assume without loss of generality $i \leq j$). By Theorem 2.9 that circuit is a union of a disjoint shortest l cycles for some $l \geq 1$. We prove that this circuit corresponds to a simple path in G_2 between the terminals $v_{f_\infty}^{i-1,i}$ and $v_{f_\infty}^{j,j+1}$ with the same weight using an induction on l.

Induction base: $l = 1$. The circuit $\mathcal{C}$ contains exactly one simple cycle C that separates between the terminal faces S^*_{ij} and $\bar{S}^*_{ij} = S^*_{(j+1)(i-1)}$ in G_1. So the vertex v_{f_∞} appear in C exactly once, i.e. $C = \langle v_{f_\infty}, v_1, v_2, \ldots, v_x, v_{f_\infty}\rangle$. According to our construction, the graph G_2 contains the same vertices and edges as G_1, except of the vertex v_{f_∞} and all the edges incident to it. Therefore, $\langle v_1, v_2, \ldots, v_x\rangle$ is a simple path in G_2. Moreover, since without loss of generality the vertex v_1 embedded between the terminal faces $f_{t_{i-1}}, f_{t_i}$ and the vertex v_x embedded between the terminal faces $f_{t_j}, f_{t_{j+1}}$, we get that $(v_{f_\infty}^{i-1,i}, v_1)$, $(v_x, v_{f_\infty}^{j,j+1}) \in E(G_2)$. Thus, $\langle v_{f_\infty}^{i-1,i}, v_1, v_2, \ldots, v_x, v_{f_\infty}^{j,j+1}\rangle$ is a simple path in G_2 with the same weight as C.

Induction step: assume that if $\mathcal{C}$ has $l' < l$ cycles, then it corresponds to a simple path in G_2 between the terminals $v_{f_\infty}^{i-1,i}$ and $v_{f_\infty}^{j,j+1}$ with the same weight, and prove it for l. There are two cases:

- If neither of the cycles in the circuit is nested. Then without loss of generality all the cycles $C \in \mathcal{C}$ bound terminal faces of S^*_{ij}. Let $C \in \mathcal{C}$ be the cycle that bound the terminal faces $f_{t_i}, \ldots, f_{t_x}$ were $i < x < j$. Thus $\{C\}$ is a simple circuit that separates between the terminal faces S^*_{ix} to $\bar{S}^*_{ix} = S^*_{(x+1)(i-1)}$, and $\mathcal{C} \setminus \{C\}$ is a simple circuit that separates between the terminal faces $S^*_{(x+1)j}$ to $\bar{S}^*_{(x+1)j} = S^*_{(j+1)x}$ in G_1. By the inductive assumption these two circuits correspond to two simple paths in G_2 with the same weights. The first path is between the two terminals $v_{f_\infty}^{i-1,i}$ and $v_{f_\infty}^{x,x+1}$, and the second is between the two terminals $v_{f_\infty}^{x,x+1}$ and $v_{f_\infty}^{j,j+1}$, which form a simple path from $v_{f_\infty}^{i-1,i}$ to $v_{f_\infty}^{j,j+1}$ in G_2 with the same weight as $\mathcal{C}$.

- There are nested cycles in the circuit. Let $C \in \mathcal{C}$ be a simple cycle that separates between S^*_{xy} and $\bar{S}^*_{xy}$ in G_2, and contains at least one cycle of $\mathcal{C} \setminus \{C\}$. If $i < x \leq y < j$ or $x < i \leq j < y$ then $\mathcal{C} \setminus \{C\}$ separates between S^*_{ij} to $\bar{S}^*_{ij}$ in contradiction to the minimality of $\mathcal{C}$. Therefore either $i = x \leq j < y$ or $i < x \leq j = y$. Assume without loss of generality that the first case holds, i.e. C is a minimum circuit that separates between S^*_{iy} and $\bar{S}^*_{iy}$, and $\mathcal{C} \setminus \{C\}$ is a minimum circuit that separates between the terminal faces $S^*_{(y+1)j}$ to $\bar{S}^*_{(y+1)j}$ in G_1. By the inductive assumption these two circuits correspond to two simple paths in G_2 with the same weights. The first simple path is between the two terminals $v^{i-1,i}_{f_\infty}$ and $v^{y,y+1}_{f_\infty}$, and the second simple path is between the two terminals $v^{y,y+1}_{f_\infty}$ and $v^{j,j+1}_{f_\infty}$. Uniting these two paths forms a simple path between $v^{i-1,i}_{f_\infty}$ to $v^{j,j+1}_{f_\infty}$ in G_2 with the same weight as $\mathcal{C}$ as we required.

Second direction - Distances to cuts. Let P be a shortest path between the terminals $v^{i-1,i}_{f_\infty}$ and $v^{j,j+1}_{f_\infty}$ in G_1 (assume $i \leq j$), and let l be the number of terminals in that path (including the two terminals in its endpoints). It is easy to verify that replacing each terminal $v^{x,x+1}_{f_\infty}$ in P with the vertex v_{f_∞} transform it to a circuit in G_1 with $l-1$ disjoint simple cycles and with the same weight of P. We prove that this circuit separates between the terminal faces $S^*_{ij} = \{f_{t_i}, \ldots, f_{t_j}\}$ and $\bar{S}^*_{ij}$ in G_1 by an induction on l.

Induction base: $l = 2$, i.e. the only terminals on the path P are those on the endpoints. Thus, all the inner vertices on that path are non terminal vertices, i.e. $P = \langle v^{i-1,i}_{f_\infty}, v_1, v_2, \ldots, v_x, v^{j,j+1}_{f_\infty} \rangle$. Substitute the terminals $v^{i-1,i}_{f_\infty}$ and $v^{j,j+1}_{f_\infty}$ of G_2 with the vertex v_{f_∞} of G_1 and get $C = \langle v_{f_\infty}, v_1, v_2, \ldots, v_x, v_{f_\infty} \rangle$. According to our construction, $\langle v_1, v_2, \ldots, v_x \rangle$ is a simple path in G_1, and $(v^{i-1,i}_{f_\infty}, v_1), (v_l, v^{j,j+1}_{f_\infty}) \in E(G_2)$ if and only if $(v_{f_\infty}, v_1), (v_x, v_{f_\infty}) \in E(G_1)$. Therefore, C is a simple cycle in G_1 and the two edges that incident to the vertex v_{f_∞} are embedded between the terminal faces $f_{t_{i-1}}$ to f_{t_i} and f_{t_j} to $f_{t_{j+1}}$ in G_1. Thus, C separates between S^*_{ij} to $\bar{S}^*_{ij}$, and has the same weight as P.

Induction step: assume that if P has $l' < l$ inner terminals then it corresponds to a simple circuit with l' cycles that separates between the terminal faces S^*_{ij} and $\bar{S}^*_{ij}$ in G_1, and prove it for $l' = l$. Let $v^{x,x+1}_{f_\infty}$ be some inner terminal in the path P that brake it into two simple sub-paths P_1 and P_2, i.e. P_1 is a simple path between $v^{i-1,i}_{f_\infty}$ to $v^{x,x+1}_{f_\infty}$ and P_2 is a simple path between $v^{x,x+1}_{f_\infty}$ to $v^{j,j+1}_{f_\infty}$ in G_2. Since both of these paths have less than l terminals we can use the inductive assumption and get that P_1 corresponds to a circuit $\mathcal{C}_1$ in G_1 with the same weight that separates between the terminals $S^*_{ix}$and $\bar{S}^*_{ix}$ and $\bar{S}^*_{ix}$, and P_2 corresponds to a circuit $\mathcal{C}_2$ in G_1 with the same weight that separates between the terminals $S^*_{(x+1)j}$ and $\bar{S}^*_{(x+1)j}$. If $i \leq x \leq j$, then $S^*_{ij} = S^*_{ix} \cup S^*_{(x+1)j}$. And if $i \leq j < x$ (symmetric to the case were $x < i \leq j$), then $S^*_{(x+1)j} = S^*_{(j+1)x}$ and so $S^*_{ij} = S^*_{ix} \setminus S^*_{(j+1)x}$. In both cases we get that $\mathcal{C}_1 \cup \mathcal{C}_2$ is a simple circuit in G_1 with the same weight as P that separates between the terminal faces S^*_{ij} and $\bar{S}^*_{ij}$ in G_1, and the Lemma follows. $\square$

Lemma 2.41. *The elementary cuts $\mathcal{T}_e(G)$ and $\mathcal{T}_e(H)$ are equal, and* $\mathrm{mincut}_G(S) \leq \mathrm{mincut}_H(S) \leq q \cdot \mathrm{mincut}_G(S)$ *for every $S \in \mathcal{T}_e(G)$.*

Proof. Let us call a shortest path between two terminals *elementary* if all the internal vertices on the path are Steiner, and denote by D_e all the terminal pairs that the shortest path between them is elementary. Moreover, recall that every elementary subset $S \in \mathcal{T}_e(G)$ is of the form $\{t_i, t_{i+1}, \ldots, t_j\}$, and denote it S_{ij} and $\bar{S}_{ij} = S_{(j+1)(i-1)}$ for simplicity.

By Lemma 2.40 a shortest circuit that separates between S^*_{ij} to $\bar{S}^*_{ij}$ in G_1 contains l elementary cycles if and only if a shortest path between the terminals $v^{i-1,i}_{f_\infty}$ and $v^{j,j+1}_{f_\infty}$ in G_2 contains $l+1$ terminals (including the endpoints). Notice that Lemma 2.40 holds also in the graphs H_2 and H_1, therefore $\mathcal{T}_e(G_1) = D_e(G_2)$ and $D_e(H_2) = \mathcal{T}_e(H_1)$. In addition, the equalities $\mathcal{T}_e(G) = \mathcal{T}_e(G_1)$ and $\mathcal{T}_e(H_1) = \mathcal{T}_e(H)$ holds by the duality between cuts and circuits, and $D_e(G_2) = D_e(H_2)$ because of the triangle inequality in the distance metric. Altogether we get that $\mathcal{T}_e(G) = \mathcal{T}_e(H)$.

Again by the duality between cuts and circuits and by Lemma 2.40 on the two pairs of graphs G, G_2 and H_2, H_1 we get that $\mathrm{mincut}_G(S_{ij}) = d_{G_2}(v^{i-1,i}_{f_\infty}, v^{j,j+1}_{f_\infty})$ and $\mathrm{mincut}_H(S_{ij}) = d_{H_2}(v^{i-1,i}_{f_\infty}, v^{j,j+1}_{f_\infty})$. Since H_2 is an (q, s)-DAM of G_2 we get that $\mathrm{mincut}_G(S_{ij}) \leq \mathrm{mincut}_H(S_{ij}) \leq q \cdot \mathrm{mincut}_G(S_{ij})$ and the lemma follows. $\square$

Lemma 2.42. *The size of H is $O(s)$.*

Proof. Given that H_2 is an (q, s)-DAM, i.e. $|V(H_2)| = s$, we need to prove that $|V(H)| = O(|V(H_2)|)$. Note that by the reduction construction $|V(H)| = |F(H_1)| = |F(H_2)| + k - 1$. Moreover, we can assume that H_2 is a simple planar graph (if it has parallel edges, we can keep the shortest one). Thus, $|E(H_2)| \leq 3|V(H_2)| + 6$. Plug it in Euler's Formula to get $|F(H_2)| \leq 2|V(H_2)| + 8$. Since $s \geq k$ we derive that $|V(H)| \leq 2s + 8 + k - 1 = O(s)$ and the lemma follows. $\square$

Proof of Theorem 2.38. Given H_2 a (q, s)-DAM of G_2 and let H be the graph that was constructed from H_2. By Lemma 2.42 and Lemma 2.41 the graph H is a $(q, O(s))$-cut-sparsifier of G. Since H_2 is a minor of G_2, and minor is closed under planar duality, then H is furthermore a minor of G and the theorem follows. $\square$

2.6.2 Proof of Theorem 2.39

Construction of the Reduction. The idea is to first "close" the shortest paths between every two terminals into shortest cycles that separates between terminal faces, and then use the planar duality between cuts and cycles to get that every shortest cycle corresponds to a minimum terminal cut that in turn preserved by an (q, s)-cut-sparsifier. More formally, given a plane k-terminal network $G = (V, E, F, T, c)$ with all its terminals $T = \{t_1, \ldots, t_k\}$ on the outerface in a clockwise order. Firstly, construct a graph G_1 by adding to G a new vertex v_{f_∞} and connects it to all its k terminals t_i using edges with 0 capacity. Note that G_1 has k new faces $f_{1,2}, \ldots, f_{k-1,k}, f_{k,1}$, where each $f_{i,i+1}$ was created by adding the two new edges (v_{f_∞}, t_i) and (v_{f_∞}, t_{i+1}). These k new faces will be the terminals of G_1.

$$\begin{aligned}
V(G_1) &:= V \cup \{v_{f_\infty}\} \\
E(G_1) &:= E \cup \{(v_{f_\infty}, t_i) \ : \ t_i \in T\} \\
F(G_1) &:= F \cup \{f_{1,2}, \ldots, f_{k-1,k}, f_{k,1}\} \\
T(G_1) &:= \{f_{1,2} \ , \ \ldots \ , \ f_{k-1,k} \ , \ f_{k,1}\}
\end{aligned}$$

Secondly, we denote by G_2 the dual graph of G_1, where its k terminals are $T(G_2) = \{v_{i,i+1} \ : \ f_{i,i+1} \in T(G_1)\}$. Moreover, the new vertex v_{f_∞} in G_1 corresponds to the outerface f_∞ of G_2, the k new edges (v_{f_∞}, t_i) we added to G_1 are the edges that lie on the outerface of G_2, and the vertices on the outerface of G_2 are the k terminals $v_{i,i+1}$ in a clockwise order. See Figure 5 from left to right for illustration, and see Section 2.7 for basic notions of planar duality.

Let H_2 be a (q, s)-cut-sparsifier and a minor of G_2. Since H_2 is a minor of G_2, then both are plane graphs with all their terminals on the outerface in the same clockwise order, and there is an edge with capacity 0 on the outerface that connects between every two adjacent terminals. Hence, we can use H_2 and the same reduction above (but in opposite order of operations) in order to construct an $(q, O(s))$-DAM H of G as follows. Firstly, let H_1 be the dual graph of H_2, where every minimum terminal cut in H_2 is equivalent to a shortest cycle that separates terminal faces. Notice that again each terminal face $f_{i,i+1}$ in H_1 contains the two edges (v_{f_∞}, t_i) and (v_{f_∞}, t_{i+1}) with capacity 0 on their boundary. Secondly, we "open" each shortest cycle in H'^* into a shortest path between terminals by removing the vertex v_{f_∞} and all its incidence edges, and denote this new graph by H. The terminals of H are all the vertices $v \in V(H_1)$ such that (v_{f_∞}, v) is an edge in H_1, which are equal to the original terminals of G. See Figure 6 from right to left for illustration.

$$\begin{aligned}
V(H) &:= V(H_1) \setminus \{v_{f_\infty}\} \\
E(H) &:= E(H_1) \setminus \{(v_{f_\infty}, t_i) \ : \ t_i \in T\} \\
F(H) &:= F(H_1) \setminus \{f_{1,2}, \ldots, f_{k-1,k}, f_{k,1}\} \\
T(H) &:= T(G)
\end{aligned}$$

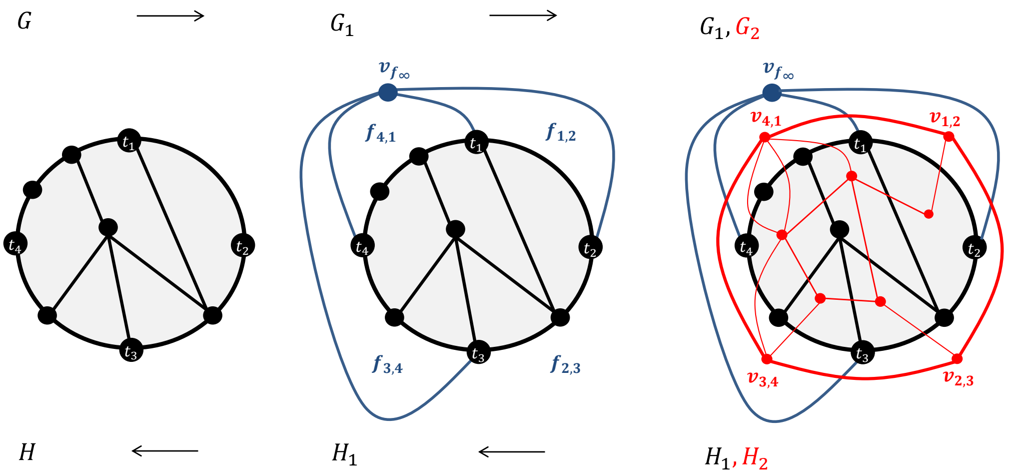


Figure 6: The first graph (in black) is the original graph G. The second is the graph G_1, where the additional vertex and edges and terminal faces colored in blue. And the third graph G_2 and its terminals are colored in red. The bold red edges are the dual of the blue edges, and both of them have capacity 0.

Analysis of the Reduction.

Lemma 2.43. *The size of H is $O(s)$.*

Proof. Given that H_2 is an (q,s)-cut-sparsifier, i.e. $|V(H_2)| = s$, we will prove that $|V(H)| = O(|V(H_2)|)$. We can assume that H_2 is a simple planar graph (if not, we can replace all the parallel edges between every two vertices by one edge where its capacity is the sum over all the capacities of these parallel edges), thus $|E(H_2)| \geq \frac{3}{2}|F(H_2)|$. Plug it in Euler's Formula to get $|F(H_2)| \leq 2|V(H_2)| - 4 = 2s - 4$. By the reduction construction $|V(H)| + 1 = |V(H_1)| = |F(H_2)| = O(s)$, and the lemma follows. □

Lemma 2.44. *The graph H is a minor of G.*

Proof. Given that H_2 is a minor of G_2, and that minor is close under deletion and contraction of edges we get that H_1 is a minor of G_1. Now by deleting the same vertex v_{f_∞} together with all its incidence edges from both G_1 and H_1, we get the graphs G and H correspondingly. Therefore H is a minor of G, and the lemma follows. □

Lemma 2.45. *The graph H preserve all the distances between every two terminals by factor q, i.e. $d_G(t_i,t_j) \leq d_H(t_i,t_j) \leq q \cdot d_G(t_i,t_j)$ for every $t_i,t_j \in T$.*

Proof. Notice that connecting all the terminals to a new vertex using edges with capacity 0 is equivalent to uniting all the terminals into one vertex, and also splitting the vertex v_{f_∞} to k new terminals is equivalent to disconnecting all the terminals by deleting that vertex. Thus our reduction is equivalent to the reduction of Theorem 2.38. In particular, Lemma 2.40 holds on the graphs G, G_1 and on the graphs H, H_1 correspondingly, i.e. every shortest path between two terminals t_i and t_j in G (or H) corresponds to a minimum circuit in G_1 (or H_1) that separates between the terminal faces $\{f_{i,i+1}, \ldots, f_{j-1,j}\}$ to $\{f_{j,j+1}, \ldots, f_{i-1,i}\}$ and vise versa.

Let $S_{(i,i+1),(j-1,j)} = \{v_{i,i+1}, \ldots, v_{j-1,j}\}$ be a set of terminals in G_2, where every terminal $v_{l,l+1}$ corresponds to the terminal face $f_{l,l+1}$ in G_1. By the duality between cuts and circuits we get that $d_G(t_i, t_j) = \text{mincut}_{G_2}(S_{(i,i+1),(j-1,j)})$ and $d_H(t_i, t_j) = \text{mincut}_{H_2}(S_{(i,i+1),(j-1,j)})$. Since H_2 is an (q, s)-cut-sparsifier of G_2 we derive the inequalities $d_G(t_i, t_j) \le d_H(t_i, t_j) \le q \cdot d_G(t_i, t_j)$ and the lemma follows. □

Proof of Theorem 2.39. By Lemma 2.43, Lemma 2.44 and Lemma 2.45 the k-terminal network H is an $(q, O(s))$-DAM of G and the theorem follows. □

2.6.3 Duality Applications

By Theorem 2.38 and Theorem 2.39, every k-terminal network G with $\gamma = 1$ admits a (q, s)-DAM if and only if it admits a minor $(q, O(s))$-cut-sparsifier. Hence, every new upper or lower bound results, especially for $q > 1$, on DAM also holds for the minor cut-sparsifier problem and vise versa. For example, the upper bound of $(1 + \epsilon, (k/\epsilon)^2)$-DAM for planar networks [CGH16] yields the following new theorem.

Theorem 2.46. *Every planar network G with $\gamma = 1$ admits a minor $(1+\epsilon, \tilde{O}((k/\epsilon)^2)$-cut-sparsifier for every $\epsilon > 0$.*

As already mentioned, by recent independent work [GHP17] these networks also admit a $(1, O(k^2))$-sparsifier that is planar but *not* a minor of G.

In addition, we can apply known upper and lower bounds for $(1, s)$-DAM to the minor mimicking network problem (i.e., a cut-sparsifier of quality 1). In particular, the known $(1, k^4)$-DAM [KNZ14] yields an alternative proof for Corollary 2.17, and the known lower bound of $(1, \Omega(k^2))$-DAM (which is shown on grid graphs) [KNZ14] yields an alternative proof for a lower bound shown in [KR13].

2.7 Planar Duality

Using planar duality we bound the size of mimicking networks for planar graphs (Theorem 2.11), and we further use it to prove the duality between cuts in distances

(Theorem 2.38 and Theorem 2.39) Recall that every planar graph G has a dual graph G^*, whose vertices correspond to the faces of G, and whose faces correspond to the vertices of G, i.e., $V(G^*) = \{v_f^* : f \in F(G)\}$ and $F(G^*) = \{f_v^* : v \in V(G)\}$. Thus the terminals $T = \{t_1, \ldots, t_k\}$ of G corresponds to the terminal faces $T(G_1) = \{f_{t_1}, \ldots, f_{t_k}\}$ in G^*, which for the sake of simplicity we may refer them as terminals as well. Every edge $e = (v, u) \in E(G)$ with capacity $c(e)$ that lies on the boundary of two faces $f_1, f_2 \in F(G)$ has a dual edge $e^* = (v_{f_1}^*, v_{f_2}^*) \in E(G^*)$ with the same capacity $c(e^*) = c(e)$ that lies on the boundary of the faces f_v^* and f_u^*. For every subset of edges $M \subset E(G)$, let $M^* := \{e^* : e \in M\}$ denote the subset of the corresponding dual edges in G^*.

The following theorem describes the duality between two different kinds of edge sets – minimum cuts and minimum circuits – in a plane multi-graph. It is a straightforward generalization of the case of st-cuts (whose dual are cycles) to three or more terminals.

A *circuit* is a collection of cycles (not necessarily disjoint) $\mathcal{C} = \{C_1, \ldots, C_l\}$. Let $\mathcal{E}(\mathcal{C}) = \cup_{i=1}^{l} C_i$ be the set of edges that participate in one or more cycles in the collection (note it is not a multiset, so we discard multiplicities). The capacity of a circuit $\mathcal{C}$ is defined as $\sum_{e \in \mathcal{E}(\mathcal{C})} c(e)$.

Theorem 2.47 (Duality between cutsets and circuits)**.** *Let G be a connected plane multi-graph, let G^* be its dual graph, and fix a subset of the vertices $W \subseteq V(G)$. Then, $M \subset E(G)$ is a cutset in G that has minimum capacity among those separating W from $V(G) \setminus W$ if and only if the dual set of edges $M^* \subseteq E(G^*)$ is actually $\mathcal{E}(\mathcal{C})$ for a circuit $\mathcal{C}$ in G^* that has minimum capacity among those separating the corresponding faces $\{f_v^* : v \in W\}$ from $\{f_v^* : v \in V(G) \setminus W\}$.*

Lemma 2.48 (The dual of a connected component)**.** *Let G be a connected plane multi-graph, let G^* be its dual, and fix a subset of edges $M \subset E(G)$. Then $W \subseteq V$ is a connected component in $G \setminus M$ if and only if its dual set of faces $\{f_v^* : v \in W\}$ is a face of $G^*[M^*]$.*

Fix $S \subset T$. We call E_S^* elementary circuit if E_S is an elementary cutset in G. Note that by Lemma 2.48 E_S^* is an elementary circuit if and only if the graph $G^* \setminus E_S^*$ has exactly two faces. Thus $\mathcal{T}_e(G) = \mathcal{T}_e(G^*)$, and the circuit E_S^* has exactly one minimum cycle in G^*. For the sake of simplicity we later on use the term cycle instead of circuit when we refer to elementary minimum circuit.

3 Flow-Cut Gaps and Face Covers in Planar Graphs

3.1 Opening

We present some new upper bounds on the gap between the concurrent flow and sparsest cut in planar graphs in terms of the topology of the terminal set. Our proof employs low-distortion metric embeddings into ℓ_1, which are known to have a tight connection to the flow-cut gap (see, e.g., [LLR95, GNRS04]). We now review the relevant terminology.

Consider an undirected graph G equipped with nonnegative edge lengths $\ell : E(G) \to \mathbb{R}_+$ and a subset $\mathsf{T} = \mathsf{T}(G) \subseteq V(G)$ of *terminal vertices*. We use $d_{G,\ell}$ to denote the shortest-path distance in G, where the length of paths is computed using the edge lengths ℓ. We use $c_1^+(G, \ell; \mathsf{T})$ to denote the minimal number $D \geq 1$ for which there exists 1-Lipschitz mapping $F : V(G) \to \ell_1$ such that $F|_{\mathsf{T}(G)}$ has bilipschitz distortion D. In other words,

$$\forall u, v \in V(G) : \qquad \|f(u) - f(v)\| \leq d_{G,\ell}(u, v)\,, \tag{8}$$
$$\forall s, t \in \mathsf{T}(G) : \qquad \|f(s) - f(t)\| \geq \tfrac{1}{D} \cdot d_{G,\ell}(s, t)\,. \tag{9}$$

For an undirected graph G, we define $c_1^+(G; \mathsf{T}) \stackrel{\text{def}}{=} \sup_\ell c_1^+(G, \ell; \mathsf{T})$, where ℓ ranges over all nonnegative lengths $\ell : E(G) \to \mathbb{R}_+$. When $\mathsf{T} = V(G)$, we may omit it and write $c_1^+(G, \ell) \stackrel{\text{def}}{=} c_1^+(G, \ell; V(G))$ and $c_1^+(G) \stackrel{\text{def}}{=} c_1^+(G; V(G))$. Finally, for a family $\mathcal{F}$ of finite graphs, we denote $c_1^+(\mathcal{F}) \stackrel{\text{def}}{=} \sup\{c_1^+(G) : G \in \mathcal{F}\}$, and for $k \in \mathbb{N}$, we denote

$$c_1^+(\mathcal{F}; k) \stackrel{\text{def}}{=} \sup\left\{c_1^+(G; \mathsf{T}) : G \in \mathcal{F}, \mathsf{T} \subseteq V(G), |\mathsf{T}| = k\right\}\,.$$

Let $\mathcal{F}_{\text{fin}}$ denote the family of all finite graphs, and $\mathcal{F}_{\text{plan}}$ the family of all planar graphs. It is known that $c_1^+(\mathcal{F}_{\text{fin}}; k) = \Theta(\log k)$ [AR98, LLR95] for all $k \geq 1$. For planar graphs, one has $c_1^+(\mathcal{F}_{\text{plan}}; k) \leq O(\sqrt{\log k})$ [Rao99] and $c_1^+(\mathcal{F}_{\text{plan}}) \geq 2$ [LR10].

Fix a plane graph G (this is a planar graph G together with a drawing in the plane). For $\mathsf{T} \subseteq V(G)$, we define the quantity $\gamma(G; \mathsf{T})$ to be the smallest number of faces in G that together cover all the vertices of T, and $\gamma(G) \stackrel{\text{def}}{=} \gamma(G; V(G))$.

We say that the pair (G, T) is an *Okamura-Seymour instance*, or in short an *OS-instance*, if it can be drawn in the plane with all its terminal on the same face, i.e., if there is a planar representation for which $\gamma(G; \mathsf{T}) = 1$. A seminal result of Okamura and Seymour [OS81] implies that $c_1^+(G; \mathsf{T}) = 1$ whenever (G, T) is an OS-instance.

The methods of [LS09] show that $c_1^+(G;\mathsf{T}) \le 2^{O(\gamma(G;\mathsf{T}))}$, and a more direct proof of [CSW13, Theorem 4.13] later showed that $c_1^+(G;\mathsf{T}) \le 3\gamma(G;\mathsf{T})$. Our main result is the following improvement.

Theorem 3.1. *For every plane graph G and terminal set $\mathsf{T} \subseteq V(G)$,*

$$c_1^+(G;\mathsf{T}) \le O(\log \gamma(G;\mathsf{T})).$$

A long-standing conjecture [GNRS04] asserts that $c_1^+(\mathcal{F}) < \infty$ for every family $\mathcal{F}$ of finite graphs that is closed under taking minors and does not contain all finite graphs. If true, this conjecture would of course imply that one can replace the bound of Theorem 3.1 with a universal constant.

It is known that a plane graph G has treewidth $O(\sqrt{\gamma(G)})$ [KLL02]. If we use $\mathcal{F}_{\text{tw}}(w)$ and $\mathcal{F}_{\text{pw}}(w)$ to denote the families of graphs of treewidth w and pathwidth w, respectively, then it is known that $c_1^+(\mathcal{F}_{\text{tw}}(2))$ is finite [GNRS04], but this remains open for $c_1^+(\mathcal{F}_{\text{tw}}(3))$. (On the other hand, $c_1^+(\mathcal{F}_{\text{pw}}(w))$ is finite for every $w \ge 1$ [LS13], and currently the best quantitative bound is $c_1^+(\mathcal{F}_{\text{pw}}(w)) \le O(\sqrt{w})$ [AFGN18].)

The parameter $\gamma(G;\mathsf{T})$ was previously studied in the context of other computational problems, including the Steiner tree problem [EMV87, Ber90, KNvL19], all-pairs shortest paths [Fre95], and cut sparsifiers [KR20, KPZ19]. For a planar graph G (without a drawing) and $\mathsf{T} \subseteq V(G)$, the *terminal face cover*, denoted $\gamma^*(G;\mathsf{T})$, is the minimum number of faces that cover T in all possible drawings of G in the plane. All our results, including Theorems 3.1, 3.3, and 3.5, hold also for the parameter $\gamma^*(G;\mathsf{T})$, simply because the relevant quantities do not depend on the graph's drawing. When G and T are given as input, $\gamma(G;\mathsf{T})$ can be computed in polynomial time [BM88], but computing $\gamma^*(G;\mathsf{T})$ is NP-hard [BM88]. In other words, while finding faces that cover T optimally in a given drawing is tractable, finding an optimal drawing is hard.

3.1.1 The Flow-Cut Gap

We now define the flow-cut gap, and briefly explain its connection to c_1^+. Consider an undirected graph G with terminals $\mathsf{T} = \mathsf{T}(G)$. Let $c : E(G) \to \mathbb{R}_+$ denote an assignment of *capacities* to edges, and $d : \binom{\mathsf{T}}{2} \to \mathbb{R}_+$ an assignment of *demands*. The triple (G, c, d) is called an (undirected) *network*. The *concurrent flow* value of the network is the maximum value $\lambda > 0$, such that $\lambda \cdot d(\{s,t\})$ units of flow can be routed between every demand pair $\{s,t\} \in \binom{\mathsf{T}}{2}$, simultaneously but as separate commodities, without exceeding edge capacities.

Given the network (G, c, d) and a subset $S \subset V$, let $\mathsf{cap}(S)$ denote the total capacity of edges crossing the cut $(S, V \setminus S)$, and let $\mathsf{dem}(S)$ denote the sum of demands

$d(\{s,t\})$ over all pairs $\{s,t\} \in \binom{\mathsf{T}}{2}$ that cross the same cut. The *sparsity of a cut* $(S, V \setminus S)$ is defined as $\mathsf{cap}(S)/\mathsf{dem}(S)$, and the *sparsest-cut value of* (G,c,d) is the minimum sparsity over all cuts in G. Finally, the *flow-cut gap* in the network (G,c,d) is defined as the ratio

$$\mathsf{gap}(G,c,d) \stackrel{\text{def}}{=} \frac{\text{sparsest-cut}(G,c,d)}{\text{concurrent-flow}(G,c,d)} \geq 1\,,$$

where the inequality is a basic exercise.

For a graph G (without capacities and demands), denote $\mathsf{gap}(G;\mathsf{T}) \stackrel{\text{def}}{=} \sup_{c,d} \mathsf{gap}(G,c,d)$, where c and $d : \binom{\mathsf{T}}{2} \to \mathbb{R}_+$ range over assignments of capacities and demands as above. The following theorem presents the fundamental duality between flow-cut gaps and ℓ_1 distortion.

Theorem 3.2 ([AR98, LLR95, GNRS04])**.** *For every finite graph G with terminals $\mathsf{T} \subseteq V(G)$,*

$$\mathsf{gap}(G;\mathsf{T}) = c_1^+(G;\mathsf{T})\,.$$

Thus our main result (Theorem 3.1) can be stated in terms of flow-cut gaps as follows.

Theorem 3.3. *For every plane graph G and terminal set $\mathsf{T} \subseteq V(G)$,*

$$\mathsf{gap}(G;\mathsf{T}) \leq O(\log \gamma(G;\mathsf{T}))\,.$$

Remark 3.4. *It is straightforward to check that our argument yields a polynomial-time algorithm that, given a plane graph G and capacities c and demands $d : \binom{\mathsf{T}}{2} \to \mathbb{R}_+$, produces a cut $(S, V(G) \setminus S)$ whose sparsity is within an $O(\log \gamma(G;\mathsf{T}))$ factor of the sparsest cut in the flow network (G,c,d).*

3.1.2 The Vertex-Capacitated Flow-Cut Gap

One can consider the analogous problems in more general networks; for instance, those which are *vertex-capacitated* (instead of edge-capacitated). In that setting, bounding the flow-cut gap appears to be significantly more challenging than for edge capacities. The authors of [FHL05] establish that the vertex-capacitated flow-cut gap is $O(\log k)$ for general networks with k terminals, and this bound is known to be tight [LR99].

For planar networks, Lee, Mendel, and Moharrami [LMM15] sought a vertex-capacitated version of the Okamura-Seymour Theorem [OS81], and proved that the vertex-capacitated flow-cut gap is $O(1)$ for instances (G,T) satisfying $\gamma(G;\mathsf{T}) = 1$.

However, it was not previously known whether the gap is bounded even for $\gamma(G;\mathsf{T}) = 2$. We prove that in planar vertex-capacitated networks (G,T) with $\gamma = \gamma(G;\mathsf{T})$, the flow-cut gap is $O(\gamma)$; see Theorem 3.20. In fact, we prove this result in the more general setting of submodular vertex capacities, also known as *polymatroid networks.* This model was introduced in [CKRV15] as a generalization of vertex capacities, and the papers [CKRV15, LMM15] showed that more refined methods in metric embedding theory are able to establish upper bounds on the flow-cut gap even in this general setting.

3.1.3 Stochastic Embeddings

Instead of embedding plane graphs with a given $\gamma(G; \mathsf{T})$ directly into ℓ_1, we will establish the stronger result that such instances can be randomly approximated by trees in a suitable sense.

If (X, d_X) is a finite metric space and $\mathcal{F}$ is a family of finite metric spaces, then a *stochastic embedding of* (X, d_X) *into* $\mathcal{F}$ is a probability distribution μ on pairs $(\varphi, (Y, d_Y))$ such that $\varphi : X \to Y$, $(Y, d_Y) \in \mathcal{F}$, and $d_Y(\varphi(x), \varphi(x')) \geq d_X(x, x')$ for all $x, x' \in X$. The *expected stretch of* μ is defined by

$$\mathrm{str}(\mu) \stackrel{\text{def}}{=} \max\left\{ \frac{\mathcal{E}_{(\varphi,(Y,d_Y))\sim\mu}\left[d_Y(\varphi(x), \varphi(x'))\right]}{d_X(x, x')} : x \neq x' \in X \right\}.$$

We will refer to an undirected graph G equipped with edge lengths $\ell_G : E(G) \to \mathbb{R}_+$ as a *metric graph,* and use d_G to denote the corresponding shortest-path distance. If G is equipped implicitly with a set $\mathsf{T}(G) \subseteq V(G)$ of terminals, we refer to it as a *terminated graph.* A graph equipped with both lengths and terminals will be called a *terminated metric graph.* We will consider any graph or metric graph G as terminated with $\mathsf{T}(G) = V(G)$ if terminals are not otherwise specified.

Given a terminated metric graph G, a *stochastic terminal embedding of* G *into a family* $\mathcal{F}$ *of terminated metric graphs* is a distribution μ over pairs (φ, F) such that $\varphi : V(G) \to V(F)$; the graph $F \in \mathcal{F}$; the terminals map to terminals:

$$\forall t \in \mathsf{T}(G), \qquad \Pr\big[\varphi(t) \in \mathsf{T}(F)\big] = 1\,;$$

and the embedding is non-contracting on terminals:

$$\forall s, t \in \mathsf{T}(G), \qquad \Pr_{(\varphi,F)\sim\mu}\big[d_F(\varphi(s), \varphi(t)) \geq d_G(s, t)\big] = 1\,. \tag{10}$$

The *expected stretch* of this embedding, again denoted $\mathrm{str}(\mu)$, is defined just as for general metric spaces:

$$\mathrm{str}(\mu) \stackrel{\text{def}}{=} \max\left\{ \frac{\mathcal{E}_{(\varphi,F)\sim\mu}\left[d_F(\varphi(u), \varphi(v))\right]}{d_G(u, v)} : u \neq v \in V(G) \right\}. \tag{11}$$

Theorem 3.5. *Consider a terminated metric plane graph G with $\gamma = \gamma(G; \mathsf{T}(G))$. Then G admits a stochastic terminal embedding into the family of metric trees with expected stretch $O(\log \gamma)$.*

Theorem 3.5 immediately yields Theorem 3.1 using the fact that every finite tree metric embeds isometrically into ℓ_1 (see, e.g., [GNRS04] for further details). The bound $O(\log \gamma)$ is optimal up to the hidden constant, as it is known that for an $m \times m$ planar grid equipped with uniform edge lengths, the expected stretch of any stochastic embedding into metric trees is at least $\Omega(\log m)$ [KRS01]. (A similar lower bound holds for the diamond graphs [GNRS04].)

Theorem 3.5 may also be of independent interest (including when $\mathsf{T}(G) = V(G)$) as embedding into dominating trees has many applications, including to competitive algorithms for online problems such as buy-at-bulk network design [AA97], and to approximation algorithms for combinatorial optimization, e.g., for the group Steiner tree problem [GKR00]. We remark that stochastic terminal embeddings into metric trees were employed by [GNR10] in the context of approximation algorithms, and were later used in [EGK+14] to design flow sparsifiers.

3.2 Approximation by random trees

Before introducing our primary technical tools, we will motivate their introduction with a high-level overview of the proof of Theorem 3.5. Fix a terminated metric plane graph G with $\gamma = \gamma(G; \mathsf{T}(G)) > 1$. Our plan is to approximate G by an OS-instance (where all terminals lie on a single face) by uniting the γ faces covering $\mathsf{T}(G)$, while approximately preserving the shortest-path metric on G. The use of stochastic embeddings will come from our need to perform this approximation randomly, preserving distances only in expectation. Using the known result that OS-instances admit stochastic terminal embeddings into metric trees, this will complete the proof.

A powerful tool for randomly "simplifying" a graph is the Peeling Lemma [LS09], which informally "peels off" any subset $A \subset V(G)$ from G, by providing a stochastic embedding of G into graphs obtained by "gluing" copies of $G \setminus A$ to the induced graph $G[A]$. The expected stretch of the embedding depends on how "nice" A is; for example, it is $O(1)$ when A is a shortest path in a planar G. The Peeling Lemma can be used to stochastically embed G into dominating OS-instances with expected stretch $2^{O(\gamma)}$ [CSW13, Section 4.5], by iteratively peeling off a shortest path A between two special faces (which has the effect of uniting them into a single face).

In contrast, our argument applies the Peeling Lemma only once. We pick A to form a connected subgraph in G that spans the γ distinguished faces. By cutting along A, one effectively merges all γ faces into a single face in a suitably chosen drawing of $G \setminus A$. The Peeling Lemma then provides a stochastic terminal embedding of G into a family of OS-instances that are constructed from copies of A and $G \setminus A$.

The expected stretch we obtain via the Peeling Lemma is controlled by how well the (induced) terminated metric graph on A can be stochastically embedded into a distribution over metric trees. For this purpose, we choose the set A to be a shortest-path tree in G that spans the γ distinguished faces, and then use a result of Sidiropoulos [Sid10] to stochastically embed A into metric trees with expected stretch that is logarithmic in the number of *leaves* (rather than logarithmic in the number of vertices, as in stochastic embeddings for general finite metric spaces [FRT04]). We remark that this is non-trivial because, while A is (topologically) a tree spanning γ faces, the relevant metric on A is d_G (which is not a path metric on $G[A]$).

3.2.1 Random partitions, embeddings, and peeling

For a finite set S, we use $\mathsf{Trees}(S)$ to denote the set of all metric spaces (S, d) that are isometric to $(V(T), d_T)$ for some metric tree T.

Theorem 3.6 (Theorem 4.4 in [Sid10])**.** *Let G be a metric graph, and let $P_1, \ldots, P_m$ be shortest paths in G sharing a common endpoint. Then the metric space $\big(\cup_{i=1}^m V(P_i), d_G\big)$ admits a stochastic embedding into $\mathsf{Trees}(\cup_{i=1}^m V(P_i))$ with expected stretch $O(\log m)$.*

Let (X, d) be a finite metric space. A distribution ν over partitions of X is called (β, Δ)-*Lipschitz* if every partition P in the support of ν satisfies $S \in P \implies \mathrm{diam}_X(S) \leq \Delta$, and moreover,

$$\forall x, y \in X, \qquad \Pr_{P \sim \nu}[P(x) \neq P(y)] \leq \beta \cdot \frac{d(x,y)}{\Delta},$$

where for $x \in X$, we use $P(x)$ to denote the unique set in P containing x.

We denote by $\beta_{(X,d)}$ the infimal $\beta \geq 0$ such that for every $\Delta > 0$, the metric (X, d) admits a (β, Δ)-Lipschitz random partition. The following theorem is due to Klein, Plotkin, and Rao [KPR93] and Rao [Rao99].

Theorem 3.7. *For every planar graph G, we have $\beta_{(V(G), d_G)} \leq O(1)$.*

Let G be a metric graph, and consider $A \subseteq V(G)$. The *dilation of A inside G* is defined to be

$$\mathrm{dil}_G(A) \stackrel{\text{def}}{=} \max_{u,v \in A} \frac{d_{G[A]}(u,v)}{d_G(u,v)},$$

where $d_{G[A]}$ denotes the induced shortest-path distance on the metric graph $G[A]$.

For two metric graphs G, G', a *1-sum of G with G'* is a graph obtained by taking two disjoint copies of G and G', and identifying a vertex $v \in V(G)$ with a vertex $v' \in V(G')$. This definition naturally extends to a 1-sum of any number of graphs. Note that the 1-sum naturally inherits its length function from G and G'.

Peeling. Consider a subset $A \subseteq V(G)$. For $a \in A$, let G_A^a denote the graph $G[(V(G) \setminus A) \cup \{a\}]$. We define the graph $\widehat{G}_A$ as the 1-sum of $G[A]$ with $\{G_A^a : a \in A\}$, where $G[A]$ is glued to each G_A^a at their common copy of $a \in A$. Let us write the vertex set of $\widehat{G}_A$ as the disjoint union:

$$V(\widehat{G}_A) = \hat{A} \sqcup \bigsqcup_{a \in A} \{(a, v) : v \in V(G) \setminus A\},$$

where $\hat{A} \stackrel{\text{def}}{=} \{\hat{a} : a \in A\}$ represents the canonical image of $G[A]$ in $\widehat{G}_A$, and (a, v) corresponds to the image of $v \in V(G) \setminus A$ in G_A^a. Say that a mapping $\psi : V(G) \to V(\widehat{G}_A)$ is a *selector map* if it satisfies:

1. For each $a \in A$, $\psi(a) = \hat{a}$.
2. For each $v \in V(G) \setminus A$, $\psi(v) \in \{(a, v) : a \in A\}$.

In other words, a selector maps each $a \in A$ to its unique copy in $\widehat{G}_A$, and maps each $v \in V(G) \setminus A$ to one of its $|A|$ copies in $\widehat{G}_A$.

Lemma 3.8 (The Peeling Lemma [LS09])**.** *Let $G = (V, E)$ be a metric graph and fix a subset $A \subseteq V$. Let G' be obtained by removing all the edges inside A:*

$$G' \stackrel{\text{def}}{=} (V, E') \qquad \textit{with} \qquad E' = E \setminus E(G[A]),$$

and denote $\beta = \beta_{(V, d_{G'})}$. Then there is a stochastic embedding μ of G into the metric graph $\widehat{G}_A$ such that μ is supported on selector maps has expected stretch $\mathrm{str}(\mu) \leq O(\beta \cdot \mathrm{dil}_G(A))$.

Remark 3.9. *The statement of the Peeling Lemma in [LS09] (see also [BLS10]) does not specify explicitly all the above details about the selector maps, but they can be easily verified by inspecting the proof.*

Composition. Consider now some metric tree $T \in \mathsf{Trees}(A)$. Via the identification between A and $\hat{A} \subseteq V(\widehat{G}_A)$, we may consider the associated metric tree $\hat{T} \in \mathsf{Trees}(\hat{A})$. Define the metric graph $\widehat{G}_A[\![T]\!]$ with vertex set $V(\widehat{G}_A)$ and edge set

$$E(\widehat{G}_A[\![T]\!]) \stackrel{\text{def}}{=} \big(E(\widehat{G}_A) \setminus E(\widehat{G}_A[\hat{A}])\big) \cup E(\hat{T}),$$

where the edge lengths are inherited from $\widehat{G}_A$ and $\hat{T}$, respectively. In other words, we replace the edges of $\widehat{G}_A[\hat{A}]$ with those coming from $\hat{T}$. Finally, denote by

$$\mathcal{F}_{G,A} \stackrel{\text{def}}{=} \left\{ \widehat{G}_A[\![T]\!] : T \in \mathsf{Trees}(A) \right\}$$

the family of all metric graphs arising in this manner. The following lemma is now immediate.

Lemma 3.10. *Every metric graph in $\mathcal{F}_{G,A}$ is a 1-sum of some $T \in \mathsf{Trees}(A)$ with the graphs $\{G_A^a : a \in A\}$.*

Suppose that μ is a stochastic embedding of G into $\widehat{G}_A$ that is supported on pairs $(\psi, \widehat{G}_A)$, where ψ is a selector map. Let ν denote a stochastic embedding of (A, d_G) into $\mathsf{Trees}(A)$. By relabeling vertices, we may assume that ν is supported on pairs (id, T) where $\mathsf{id} : A \to A$ is the identity map. Altogether, we obtain a stochastic embedding of G into $\mathcal{F}_{G,A}$, which we denote $\nu \circ \mu$ and define by

$$\forall T \in \mathsf{Trees}(A), \qquad (\nu \circ \mu)(\psi, \widehat{G}_A[\![T]\!]) \stackrel{\text{def}}{=} \mu(\psi, \widehat{G}_A) \cdot \nu(\mathsf{id}, T)\,,$$

where the product between the probability measures μ and ν represents drawing from the two distributions independently. While notationally cumbersome, the following claim is now straightforward.

Lemma 3.11 (Composition Lemma)**.** *It holds that*

$$\operatorname{str}(\nu \circ \mu) \leq \operatorname{str}(\nu) \cdot \operatorname{str}(\mu)\,.$$

3.2.2 Approximation by OS-instances

Let us now show that every terminated metric plane graph G with $\gamma = \gamma(G; \mathsf{T}(G))$ admits a stochastic terminal embedding into OS-instances. In Section 3.2.3, we recall how OS-instances can be stochastically embedded into metric trees, thereby completing the proof of Theorem 3.5.

Let $F_1, \ldots, F_\gamma$ be faces of G that cover $\mathsf{T}(G)$, and denote $T_i \stackrel{\text{def}}{=} V(F_i) \cap \mathsf{T}(G)$. For each $i \geq 1$, fix an arbitrary vertex $v_i \in V(F_i)$. Denote $r \stackrel{\text{def}}{=} v_1$, and for each $i \geq 2$, let P_i be the shortest path from v_i to r. Finally, let P be the tree obtained as the union of these paths, namely, the induced graph $G[\cup_{i\geq 2} P_i]$.

We present now Klein's Tree-Cut operation [Kle06]. It takes as input a plane graph G and a tree T in G, and "cuts open" the tree to create a new face F_{new}. More specifically, consider walking "around" the tree and creating a new copy of each vertex and edge of T encountered along the way. This operation maintains planarity while replacing the tree T with a simple cycle C_T that bounds the new face. It is easy to verify that C_T has two copies of every edge of T, and $\deg_T(v)$ copies of every vertex of T, where $\deg_T(v)$ stands for the degree of v in T. This Tree-Cut operation can also be found in [Bor04, BKK07, BKM09].

We apply Klein's Tree-Cut operation to G and the tree P, and let G_1 be the resulting metric plane graph with the new face F_{new}, after we replace P with a simple cycle C_P; see Figure 7 for illustration. Since P shares at least one vertex with each face F_i in G (namely, v_i), the cycle C_P shares at least one vertex with each face F_i in G_1.

We now construct G_2 by applying two operations on G_1. First, for every face F_i that shares exactly one vertex with C_P, namely only v_i (or actually a copy of it), we split this vertex into two as follows. Let $N^1_{G_1}(v_i)$ be all the neighbors of v_i in G_1 embedded between the face F_i and F_{new} on one side, and $N^2_{G_1}(v_i)$ be all its neighbors on the other side. We split v_i into two vertices v'_i, v''_i that are connected by an edge of length 0, and connect all the vertices in $N^1_{G_1}(v_i)$ to v'_i and all the vertices in $N^2_{G_1}(v_i)$ to v''_i. See Figure 8 for illustration. Notice that this new edge $\{v'_i, v''_i\}$ is incident to both F_i and F_{new}, and that this operation maintains the planarity, along with the distance metric of G_1 (in the straightforward sense, where one takes a quotient by vertices at distance 0 from each other).

The second operation adds between all the copies of the same $v \in V(P)$ a star with edge length 0 drawn inside F_{new}. Note that adding the stars inside F_{new} does not violate the planarity since all the copies of the vertices in C_P are ordered by the walk around P; see Figure 7 for illustration. It is easy to verify that if we identify each $v \in V(P)$ with one of its copies in G_2 arbitrarily then

$$\forall x, y \in V(G), \qquad d_G(x, y) = d_{G_2}(x, y). \tag{12}$$

Lemma 3.12. *$(V(P), d_G)$ admits a stochastic embedding into $\mathsf{Trees}(V(P))$ with expected stretch at most $O(\log \gamma)$.*

Proof. Apply Theorem 3.6 on the shortest-paths $P_2, \ldots, P_\gamma$ in G, with shared vertex $v_1 = r$. □

Let $A \subseteq V(G_2)$ denote all the vertices on the boundary of F_{new} in G_2. To every $T \in \mathsf{Trees}(V(P))$, we can associate a tree $T' \in \mathsf{Trees}(A)$ by identifying $x \in V(P)$ with one of its copies in A, and attaching the rest of its copies to x with an edge of length 0. Using (12) in conjunction with Lemma 3.12 yields the following.

Corollary 3.13. *(A, d_{G_2}) admits a stochastic embedding into $\mathsf{Trees}(A)$ with expected stretch at most $O(\log \gamma)$.*

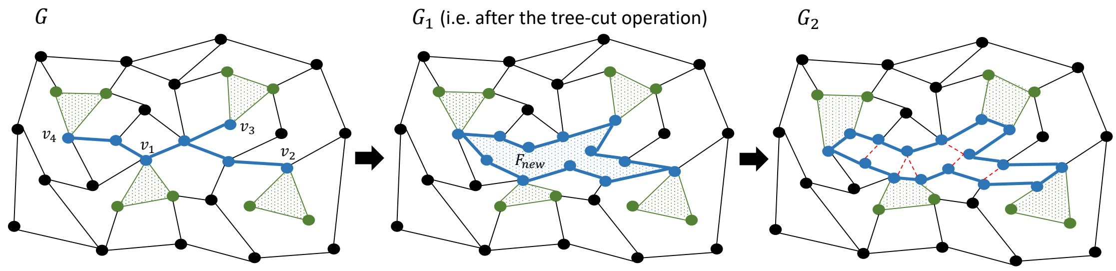


Figure 7: In G, the tree P (in blue) is incident to all $\gamma = 4$ distinguished special faces (drawn in green). G_1 is obtained by applying the Tree-Cut operation on G and P, which creates a new face F_{new}. Finally, G_2 is obtained by duplicating some vertices on F_{new} and connecting copies of the same vertex by length-zero edges (the dashed red edges).

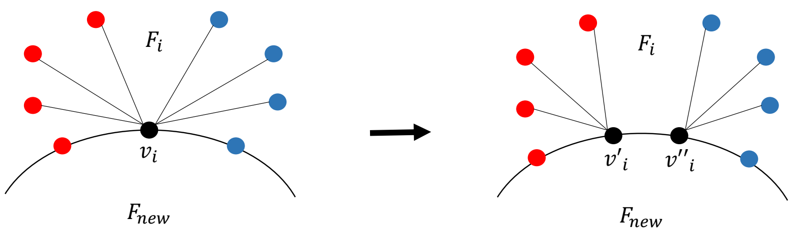


Figure 8: The neighbors of v_i are partitioned into two sets (colored red and blue) by going around v_i in the plane and watching for the location of faces F_i and F_{new}, to eventually split v_i into two.

Let H be the graph obtained from G_2 by adding an edge $\{u,v\}$ of length $d_G(u,v)$ between every pair of vertices $u,v \in A$. By construction, we have $\mathrm{dil}_H(A) = 1$. Let $E' \stackrel{\text{def}}{=} E(H) \setminus E(H[A])$, and $H' = (V(H), E')$. While H is in general non-planar, the graph H' and H_A^a for $a \in A$ are subgraphs of the planar graph G_2, and are thus planar as well, and by Theorem 3.7 we have $\beta_{(V(H),d_{H'})} \leq O(1)$.

By applying the Peeling Lemma (Lemma 3.8) to H and $A \subseteq V(H)$, we obtain a stochastic embedding μ of H into $\widehat{H}_A$ such that μ is supported on selector maps and $\mathrm{str}(\mu) \leq O(1)$. Using Corollary 3.13 and the fact that (A, d_H) is the same as (A, d_{G_2}), we obtain a stochastic embedding ν of (A, d_H) into $\mathsf{Trees}(A)$ with $\mathrm{str}(\nu) \leq O(\log \gamma)$.

Define $\mathsf{T}(H)$ to be the set of vertices in $\mathsf{T}(G)$ together with all their copies created in the construction of H, and

$$\mathsf{T}(\widehat{H}_A) \stackrel{\text{def}}{=} \{\hat{a} : a \in \mathsf{T}(H)\} \cup \{(v,a) : v \in \mathsf{T}(H), a \in A\}\,.$$

By convention, for any subgraph H' of H we have $\mathsf{T}(H') \stackrel{\text{def}}{=} V(H') \cap \mathsf{T}(H)$.

Applying the Composition Lemma (Lemma 3.11) to the pair μ, ν (in conjunction with Lemma 3.10) yields a stochastic embedding $\pi \stackrel{\text{def}}{=} \nu \circ \mu$ satisfying the next lemma.

Lemma 3.14. *$(V(G), d_G)$ admits a stochastic embedding π into the family of metric graphs that are 1-sums of a metric tree with the graphs $\{H_A^a : a \in A\}$, where H_A^a is glued to T along a vertex of $\mathsf{T}(H_A^a)$, and such that $\mathrm{str}(\pi) \leq O(\log \gamma)$. Moreover, every $(\varphi, W) \in \mathrm{supp}(\pi)$ satisfies $\varphi(\mathsf{T}(G)) \subseteq \mathsf{T}(W)$.*

It remains to prove that π in this lemma is an embedding into OS-instances, i.e., every 1-sum in the support of π is an OS-instance. We first show this for every pair $\{(H_A^a, \mathsf{T}(H_A^a)) : a \in A\}$.

Lemma 3.15. *For every $a \in A$, there is a face F_a in H_A^a such that $\mathsf{T}(H_A^a) \subseteq V(F_a)$.*

Proof. Fix $a \in A$. The graph G_2 is planar, and while H need not be planar, the subgraphs $G_2[(V(G_2) \setminus A) \cup \{a\}]$ and H_A^a are identical for each $a \in A$. Thus, it suffices to prove the lemma for the subgraphs $G_2[(V(G_2) \setminus A) \cup \{a\}]$.

Observe that if we remove from G_2 a vertex $v \in V(G_2)$, then all the faces incident to v in G_2 become one new face in the graph $G_2 \setminus \{v\}$. Moreover, if we remove from G_2 both endpoints of an edge $\{u,v\}$, then all the faces incident to either u or v become one new face in $G_2 \setminus \{u,v\}$. Recall that $G_2[A]$ is a simple cycle (bounding F_{new}), thus $G_2[A \setminus \{a\}] = G_2[A] \setminus \{a\}$ is connected, and all the faces incident to at least one vertex in $A \setminus \{a\}$ become one new face in $G_2[(V(G_2) \setminus A) \cup \{a\}]$, which we denote F_{new}^a.

By construction of G_2 (which splits a vertex of G_1 if it is the only vertex incident to both F_i and F_{new}), every face F_i is incident to at least two vertices in A, and thus to at least one in $A \setminus \{a\}$. It follows that all the terminals in $G_2[(V(G_2) \setminus A) \cup \{a\}]$ are on the same face F^a_{new}. In addition, since a has at least one neighboring vertex $b \in A$, at least one face is incident to both a and b in G_2, and it becomes part of the face F^a_{new} in $G_2[(V(G_2) \setminus A) \cup \{a\}]$. Therefore, $a \in V(F^a_{new})$ as well, and the lemma follows. □

Lemma 3.16. *Suppose W is a planar graph formed from the 1-sum of a tree T and a collection of (pairwise disjoint) plane graphs $\{H_a : a \in A\}$, where each H_a has a distinguished face F_a, and H_a is glued to T along a vertex of $V(F_a)$. Then there exists a drawing of W in which all the vertices $V(T) \cup \bigcup_{a\in A} V(F_a)$ lie on the outer face.*

Proof. It is well-known that every plane graph can be redrawn so that any desired face is the outer face (see, e.g., [Whi32]). So we may first construct a planar drawing of T, and then extend this to a planar drawing of W where each H_a is drawn so that F_a bounds the image of H_a, and the interior of F_a contains only the images of vertices in $V(H_a)$. □

Combining Lemmas 3.14, 3.15 and 3.16 yields the following corollary.

Corollary 3.17. *G admits a stochastic embedding with expected stretch $O(\log \gamma)$ into a family $\mathcal{F}$ of terminated metric plane graphs, where each $W \in \mathcal{F}$ satisfies $\gamma(W; \mathsf{T}(W)) = 1$.*

Note that in the stochastic embedding of this corollary, the stretch guarantee applies to all vertices (and not only to terminals), and the choice of terminals restricts the host graphs $W \in \mathcal{F}$, as they are OS-instances.

3.2.3 From OS-instances to random trees

We need a couple of known embedding theorems.

Theorem 3.18 ([GNRS04, Thm. 5.4])**.** *Every metric outerplanar graph admits a stochastic embedding into metric trees with expected stretch $O(1)$.*

The next result is proved in [LMM15, Thm. 4.4] (which is essentially a restatement of [EGK+14, Thm. 12]).

Theorem 3.19. *If G is a terminated metric plane graph and $\gamma(G; \mathsf{T}(G)) = 1$, then G admits a stochastic terminal embedding into metric outerplanar graphs with expected stretch $O(1)$.*

In conjunction with Theorem 3.18, this shows that every OS-instance admits a stochastic terminal embedding into metric trees with expected stretch $O(1)$. Combined with Corollary 3.17, this finishes the proof of Theorem 3.5.

3.3 Polymatroid flow-cut gaps

We now discuss a network model introduced in [CKRV15] that generalizes edge and vertex capacities. Recall that if S is a finite set, then a function $f : 2^S \to \mathbb{R}$ is called *submodular* if $f(A) + f(B) \geq f(A \cap B) + f(A \cup B)$ for all subsets $A, B \subseteq S$. For an undirected graph $G = (V, E)$, we let $E(v)$ denote the set of edges incident to v. A collection $\vec{\rho} = \{\rho_v : 2^{E(v)} \to \mathbb{R}_+\}_{v \in V}$ of monotone, submodular functions are called *polymatroid capacities on G.*

Say that a function $\varphi : E \to \mathbb{R}_+$ is *feasible with respect to $\vec{\rho}$* if it holds that for every $v \in V$ and subset $S \subseteq E(v)$, it holds that $\sum_{e \in S} \varphi(e) \leq \rho_v(S)$. Given demands $\mathsf{dem} : V \times V \to \mathbb{R}_+$, one defines the *maximum concurrent flow value* of the polymatroid network $(G, \vec{\rho}, \mathsf{dem})$, denoted $\mathrm{mcf}_G(\vec{\rho}, \mathsf{dem})$, as the maximum value $\epsilon > 0$ such that one can route an ϵ-fraction of all demands simultaneously using a flow that is feasible with respect to $\vec{\rho}$.

For every subset $S \subseteq E$, define the cut semimetric $\sigma_S : V \times V \to \{0, 1\}$ by $\sigma_S(u, v) \stackrel{\text{def}}{=} 0$ if and only if there is a path from u to v in the graph $G(V, E \setminus S)$. Say that a map $g : S \to V$ is *valid* if it maps every edge in S to one of its two endpoints in V. One then defines the *capacity of a set $S \subseteq E$* by

$$\nu_{\vec{\rho}}(S) \stackrel{\text{def}}{=} \min_{\substack{g: S \to V \\ \text{valid}}} \sum_{v \in V} \rho_v(g^{-1}(v)) \,.$$

The *sparsity of S* is given by

$$\Phi_G(S; \vec{\rho}, \mathsf{dem}) \stackrel{\text{def}}{=} \frac{\nu_{\vec{\rho}}(S)}{\sum_{u,v \in V} \mathsf{dem}(u, v) \sigma_S(u, v)} \,.$$

We also define $\Phi_G(\vec{\rho}, \mathsf{dem}) \stackrel{\text{def}}{=} \min_{\emptyset \neq S \subseteq V} \Phi(S; \vec{\rho}, \mathsf{dem})$. Our goal in this section is to prove the following theorem.

Theorem 3.20. *There is a constant $C \geq 1$ such that the following holds. Suppose that $G = (V, E)$ is a planar graph and $D \subseteq F_1 \cup F_2 \cup \cdots \cup F_\gamma$, where each F_i is a face of G. Then for every collection $\vec{\rho}$ of polymatroid capacities on G and every set of demands $\mathsf{dem} : D \times D \to \mathbb{R}_+$ supported on D, it holds that*

$$\mathrm{mcf}_G(\vec{\rho}, \mathsf{dem}) \leq \Phi_G(\vec{\rho}, \mathsf{dem}) \leq C\gamma \cdot \mathrm{mcf}_G(\vec{\rho}, \mathsf{dem}) \,.$$

3.3.1 Embeddings into thin trees

In order to prove this, we need two results from [LMM15]. Suppose G is an undirected graph, T is a connected tree, and $f : V(G) \to V(T)$. For every distinct pair $u, v \in V(G)$, let P^T_{uv} denote the unique simple path from $f(u)$ to $f(v)$ in T. Say that the map f is Δ-*thin* if, for every $u \in V(G)$, the induced subgraph on $\bigcup_{v:\{u,v\}\in E(G)} P^T_{uv}$ can be covered by Δ simple paths in T emanating from $f(u)$.

Suppose further that G is equipped with edge lengths $\ell : E(G) \to \mathbb{R}_+$. If (X, d_X) is a metric space and $f : V(G) \to X$, we make the following definition. For $\tau > 0$ and any $u \in V(G)$:

$$|\nabla_\tau f(u)|_\infty \stackrel{\text{def}}{=} \max \left\{ \frac{d_X(f(u), f(v))}{\ell(u,v)} : \{u,v\} \in E \text{ and } \ell(u,v) \in [\tau, 2\tau] \right\}.$$

Fact 3.21. *Suppose that $f : V(G) \to \mathbb{R}$ is 1-Lipschitz, where $V(G)$ is equipped with the path metric $d_{G,\ell}$. Then f is 2-thin and*

$$\max \{|\nabla_\tau f(u)|_\infty : u \in V(G), \tau > 0\} \le 1\,.$$

Theorem 3.22 (Rounding theorem [LMM15])**.** *Consider a graph $G = (V, E)$ and a subset $D \subseteq V$. Suppose that for every length $\ell : E \to \mathbb{R}_+$, there is a random Δ-thin mapping $\Psi : V \to V(T)$ into some random tree T that satisfies:*

1. *For every $v \in V$ and $\tau > 0$: $\mathcal{E}|\nabla_\tau \Psi(v)|_\infty \le L$.*
2. *For every $u, v \in D$:*

$$\mathcal{E}\left[d_T(\Psi(u), \Psi(v))\right] \ge \frac{d_{G,\ell}(u,v)}{K}\,.$$

Then for every collection $\vec{\rho}$ of polymatroid capacities on G and every set of demands $\mathsf{dem} : D \times D \to \mathbb{R}_+$ *supported on D, it holds that*

$$\mathrm{mcf}_G(\vec{\rho}, \mathsf{dem}) \le \Phi_G(\vec{\rho}, \mathsf{dem}) \le O(\Delta K L) \cdot \mathrm{mcf}_G(\vec{\rho}, \mathsf{dem})\,.$$

Theorem 3.23 (Face embedding theorem [LMM15])**.** *Suppose that $G = (V, E)$ is a planar graph and $D \subseteq V$ is a subset of vertices contained in a single face of G. Then for every $\ell : E \to \mathbb{R}_+$, there is a random 4-thin mapping $\Psi : V \to V(T)$ into a random tree metric that satisfies the assumptions of Theorem 3.22 with $K, L \le O(1)$.*

We now use this to prove the following multi-face embedding theorem; combined with Theorem 3.22, it yields Theorem 3.20.

Theorem 3.24 (Multi-face embedding theorem)**.** *Suppose that $G = (V, E)$ is a planar graph and $D \subseteq F_1 \cup F_2 \cup \cdots \cup F_\gamma$, where each F_i is a face of G. Then for every $\ell : E \to \mathbb{R}_+$, there is a random 4-thin mapping $\Psi : V \to V(T)$ into a random tree metric that satisfies the assumptions of Theorem 3.22 with $L \leq O(1)$ and $K \leq O(\gamma)$.*

Proof. For each $i = 1, 2, \ldots, \gamma$, let $\Psi_i : V \to V(T_i)$ be the random 4-thin mapping guaranteed by Theorem 3.23 with constants $1 \leq K_0, L_0 \leq O(1)$, and let $\Psi'_i : V \to \mathbb{R}$ be the 2-thin mapping given by $\Psi'_i(v) = d_{G,\ell}(v, F_i)$ (recall Fact 3.21). Now let $\Psi : V \to V(T)$ be the random map that arises from choosing one of $\{\Psi_1, \ldots, \Psi_\gamma, \Psi'_1, \ldots, \Psi'_\gamma\}$ uniformly at random. Then Ψ is a random 4-thin mapping satisfying (1) in Theorem 3.22 for some $L \leq O(1)$.

Consider now some $u \in F_i$ and $v \in V$. Let $u' \in F_i$ be such that $d_{G,\ell}(v, u') = d_{G,\ell}(v, F_i)$. If $d_{G,\ell}(u', v) \geq \frac{d_{G,\ell}(u,v)}{4K_0L_0}$, then

$$\mathcal{E}\left[d_T(\Psi(u), \Psi(v))\right] \geq \frac{1}{2\gamma}\left|\Psi'_i(u) - \Psi'_i(v)\right| = \frac{d_{G,\ell}(u', v)}{2\gamma} \geq \frac{d_{G,\ell}(u, v)}{8\gamma K_0 L_0}\,.$$

If, on the other hand, $d_{G,\ell}(u', v) < \frac{d_{G,\ell}(u,v)}{4K_0L_0}$, then

$$\begin{aligned}
\mathcal{E}\left[d_T(\Psi(u), \Psi(v))\right] &\geq \frac{1}{2\gamma}\mathcal{E}\left[d_{T_i}(\Psi_i(u), \Psi_i(v))\right] \\
&\geq \frac{1}{2\gamma}\mathcal{E}\left[d_{T_i}(\Psi_i(u), \Psi_i(u')) - d_{T_i}(\Psi_i(u'), \Psi_i(v))\right] \\
&\geq \frac{1}{2\gamma}\left(\frac{d_{G,\ell}(u, u')}{K_0} - L_0\, d_{G,\ell}(u', v)\right) \\
&\geq \frac{1}{2\gamma}\left(\frac{d_{G,\ell}(u, v) - d_{G,\ell}(u', v)}{K_0} - \frac{d_{G,\ell}(u, v)}{4K_0}\right) \\
&\geq \frac{1}{2\gamma}\left(\frac{3}{4}\frac{d_{G,\ell}(u, v)}{K_0} - \frac{d_{G,\ell}(u', v)}{K_0}\right) \\
&\geq \frac{d_{G,\ell}(u, v)}{4\gamma K_0}\,.
\end{aligned}$$

Thus Ψ also satisfies (2) in Theorem 3.22 with $K \leq O(\gamma)$, completing the proof. $\square$

4 Faster Algorithms for Orienteering and k-TSP

4.1 Opening

The Traveling Salesman Problem (TSP) is of fundamental importance to combinatorial optimization, computer science and operations research. It is a prototypical problem for planning routes in almost any context, from logistics to manufacturing, and is therefore studied extensively. In this problem, the input is a list of cities (aka sites) and their pairwise distances, and the goal is to find a (closed) tour of minimum length that visits all the sites. This problem is known to be NP-hard even in the Euclidean case [GGJ76, Pap77, Tre00], which is the focus of our work.

One important variant of TSP are *orienteering* problems, which ask to maximize the number of sites visited when the tour length is constrained by a given budget. These problems model scenarios where the "salesman" has limited resources, such as gasoline, time or battery-life. This genre is related to prize-collecting traveling salesman problems, introduced by Balas [Bal95], where the sites are also associated with non-negative "prize" values, and the goal is to visit a subset of the sites while minimizing the total distance traveled and maximizing the total amount of prize collected. Note that there is a trade-off between the cost of a tour and how much prize it spans. Another related family is the vehicle routing problem (VRP) [TV02], where the goal is to find optimal routes for multiple vehicles visiting a set of sites. These problems arise from real-world applications such as delivering goods to locations or assigning technicians to maintenance jobs.

We consider the *rooted* orienteering problem in Euclidean space, in which the input is a set of n points P in $\mathbb{R}^d$, a starting point s and a budget $\mathcal{B} > 0$, and the goal is to find a path that starts at s and visits as many points of P as possible, such that the path length is at most $\mathcal{B}$. A $(1-\delta)$-approximate solution is a path satisfying these constraints (start at s and have length at most $\mathcal{B}$) that visits at least $(1-\delta)k_{\text{opt}}$ points, where k_{opt} denotes the maximum possible, i.e., the number of points visited by an optimal path.

Arkin, Mitchell, and Narasimhan [AMN98] designed the first approximation algorithms for the rooted orienteering problem. They considered this problem for points in the Euclidean plane when the desired "tour" (network in their context) is a path, a cycle, or a tree, and achieved a $O(1)$–approximation for these problems. Blum et al. [BCK+07] and Bansal et al. [BBCM04] designed an $O(1)$-approximation algorithm for rooted path orienteering when the points lie in a general metric space.

Chen and Har-Peled [CH08] were the first to design a Polynomial-Time Approximation Scheme (PTAS), i.e., a $(1-\delta)$-approximation algorithm for every fixed $\delta > 0$, when the points lie in a Euclidean space of fixed dimension. Their algorithm reduces the orienteering problem into (a multi-path version of) rooted k-TSP, and thus the heart of their algorithm is a PTAS for the latter, where the approximation is actually with respect to a parameter called *excess*, which can be much smaller than the optimal tour length. This follows an earlier approach of Blum et al. [BCK+07], who introduced the concept of excess-based approximation, and designed a reduction to a simpler (single-path version of) k-TSP. However, that earlier reduction increases the approximation ratio by a constant factor and cannot yield a PTAS. Chen and Har-Peled [CH08] presented a different reduction, to a more complicated (multi-path) version of rooted k-TSP, and for the latter problem they designed an algorithm that cleverly combines two very different divide-and-conquer methods, of Arora [Aro98] and of Mitchell [Mit99]. As they point out, a key difficulty in this problem is the relative lack of algorithmic tools to handle rigid budget constraints.

We design a PTAS for the rooted orienteering problem that has a better running time than the known running time $n^{O(d\sqrt{d}/\delta)}(\log n)^{(d/\delta)^{O(d)}}$ of Chen and Har-Peled [CH08]. For fixed δ and small dimension d, the leading term in their running time is about $n^{O(d\sqrt{d}/\delta)}$, which we improve to $n^{O(1/\delta)}$. Thanks to this improvement, our algorithm is polynomial even for a moderately large dimension, roughly up to $d = O(\log \log n)$ instead of $d = O(1)$.

4.1.1 Our Results

Our main result is a PTAS for the rooted orienteering problem, with improved running time compared to that of Chen and Har-Peled [CH08].

Theorem 4.1. *Given as input a set P of n points in $\mathbb{R}^d$, a starting point s, a budget $\mathcal{B} > 0$, and an accuracy parameter $\delta \in (0, 1)$, one can compute in time $n^{O(1/\delta)}(\log n)^{(d/\delta)^{O(d)}}$, a path that starts at s, has length at most $\mathcal{B}$, and visits at least $(1-\delta)k_{\text{opt}}$ points of P, where k_{opt} is the maximum possible number of points that can be visited under these constraints.*

Similarly to Chen and Har-Peled [CH08], our algorithm reduces the rooted orienteering problem to (a multi-path version of) rooted k-TSP, and the main challenge is to solve the latter problem with good approximation with respect to the excess parameter. Their algorithm for k-TSP uses Mitchell's divide-and-conquer method [Mit99] based on splitting the space into windows. These windows contain subpaths of the k-TSP path, and the algorithm finds such subpaths in every window and then combines them into the requested path. The leading term $n^{\text{poly}(d)}$ in the running time of Chen and Har-Peled [CH08] arises from defining each window via $2d$ independent hyperplanes, which gives rise to $n^{O(d)}$ possible windows. We define and order the windows in a way that is similar to, but different from, Blum et al. [BCK+07], which yields at most n^2 different windows (see Section 4.2 for more details). This improvement is readily seen in our first technical result (Theorem 4.2), which provides a PTAS for (a simple version of) rooted k-TSP. For fixed δ and small dimension d, the leading term in our running time is $n^{O(1)}$. For our main result, we need to solve a multi-path version that we call rooted (m,k)-TSP. This problem asks to find m paths that visit k points in total, when the input prescribes the endpoints of all these m paths (see Theorem 4.3). We note that although our result for k-TSP can be obtained also by applying the techniques of Blum et al. [BCK+07], our version of the windows is essential for solving the rooted (m,k)-TSP, and is thus needed to obtain a PTAS for orienteering.

4.1.2 Related Work

The orienteering problem was first introduced by Golden et al. [GLV87], and intensely studied since then. The problem has numerous variants. For example, Chekuri et al. [CKP12] designed $(2+\epsilon)$-approximation algorithm for orienteering in undirected graphs, and an $O(\log^2 \mathrm{OPT})$-approximation algorithm in directed graphs. Gupta et al. [GKNR15] designed an $O(1)$-approximation algorithm for the best non-adaptive policy for stochastic orienteering. Friggstad et al. [FS17] introduced the first polynomial-size LP-relaxations for the orienteering problem and its rooted version, and obtained $O(1)$-approximation algorithms via LP-rounding. For algorithms for orienteering with deadlines and time-windows see [BBCM04, CK04, CP05]. A survey on orienteering can be found in [GLV16], and a survey on the vehicle routing problem with profits can be found in [ASV14].

4.1.3 Preliminaries

Notation. Let $\pi = \langle p_1, \dots, p_k \rangle$ be a path that visits k points of P in $\mathbb{R}^d$, starting at p_1 and ending at p_k. The *length* of π is denoted by $\|\pi\| := \sum_{j=1}^{k-1} \|p_{j+1} - p_j\|$, and let $P(\pi)$ be all the points in P that are visited by π. Define the *excess* of π to be

$$\mathcal{E}(\pi) := \|\pi\| - \|p_k - p_1\|.$$

Note that the excess of π may be considerably smaller than the length of π. Similarly, given a set Π of m paths, such that each path π_i, $i \in [m]$, connects endpoints s_i, t_i, we denote the total length of its m paths by $\|\Pi\| := \sum_{i=1}^{m} \|\pi_i\|$. Let $P(\Pi)$ be all points visited by Π, i.e. $P(\Pi) = \cup_{i=1}^{m} P(\pi_i)$, and let the excess of Π be $\mathcal{E}(\Pi) := \sum_{i=1}^{m} (\|\pi_i\| - \|t_i - s_i\|)$.

Given a set P of n points and m pairs $s_i, t_i \in P$, the *rooted (m,k)-TSP problem* is to find a set of m paths $\Pi = \{\pi_i |\ i \in [m]\}$ with minimum total length, such that each path π_i connects endpoints s_i, t_i, and $|P(\Pi)| = k$. A *δ-excess-approximation* to the rooted (m,k)-TSP problem is a set of m paths $\Pi = \{\pi_i |\ i \in [m]\}$, such that each path π_i connects endpoints s_i, t_i, $|P(\Pi)| = k$, and $\|\Pi\| \le \|\Pi^*\| + \delta \cdot \mathcal{E}(\Pi^*)$, where Π^* a solution of minimum length. We define the *rooted k-TSP problem* to be the rooted $(1, k)$-TSP problem.

Given a set P of n points, a budget $\mathcal{B}$, and a starting point s, the *rooted orienteering problem* is the problem of finding a path π^* rooted at s which visits the maximum number of points of P under the constraint that $\|\pi^*\| \le \mathcal{B}$. Let k_{opt} denote the number of points visited by π^*. A $(1-\delta)$-approximation to the rooted orienteering problem is a path π rooted at s which visits at least $(1-\delta)k_{\text{opt}}$ vertices under the constraint that $\|\pi\| \le \mathcal{B}$.

Algorithms. Arora [Aro98] gave a PTAS for Euclidean TSP which runs in time $n(\log n)^{(d/\delta)^{O(d)}}$. He also showed how to modify the algorithm to solve k-TSP in time $k^2 n(\log k)^{(d/\delta)^{O(d)}}$. This is done by modifying the dynamic program, so that for every candidate cell the program computes an optimal tour visiting at least k' points, for each $k' \in [k]$. (We also note the dependence on $\log k$ instead of $\log n$.) Another simple modification to Arora's algorithm is to compute m tours which together visit all points, and this increases the runtime to $n(2^m \log n)^{(d/\delta)^{O(d)}}$.[4] Combining these two separate extensions gives a solution to (m, k)-TSP in time $k^2 n(2^m \log n)^{(d/\delta)^{O(d)}}$.

4.2 A δ-excess-approximation algorithm for rooted (m, k)-TSP

In this section, we present the δ-excess-approximation algorithm for rooted (m, k)-TSP. Later in Section 4.3, we use this algorithm as a subroutine to approximate the orienteering problem. For purposes of exposition, we will first show how to solve the case $m = 1$, i.e., rooted k-TSP, using a plane sweep algorithm (PSA), and then extend this plane sweep algorithm to general m, i.e., solve (m, k)-TSP.

Our algorithm combines ideas from Blum et al. [BCK+07] and Chen and Har-Peled [CH08]. In Section 4.2.1 we solve k-TSP using the techniques of Blum et al. [BCK+07], but with a critical modification that uses the Euclidean (rather than metric) setting and allows extension of our algorithm to the more general (m, k)-TSP in Section 4.2.2.

[4] For those familiar with Arora's construction, we must add to each active portal a list of tours incident upon it. The factor 2^m represents the ensuing increase in the number of possible configurations.

4.2.1 Algorithm for rooted k-TSP

We present δ-excess-approximation algorithm for rooted k-TSP, as follows.

Theorem 4.2. *Given as input the endpoints $s, t \in \mathbb{R}^d$, a set of n points $P \subset \mathbb{R}^d$, an integer $2 \leq k \leq n$, and an accuracy parameter $\delta \in (0,1)$, there is an algorithm that runs in time $n^{O(1)}(\log k)^{(d/\delta)^{O(d)}}$ and finds a k-TSP path from s to t of length at most* $\mathrm{OPT} + \delta \cdot \mathcal{E}$, *where* OPT *is the minimum length of a k-TSP path from s to t, and its excess is denoted by $\mathcal{E} = \mathrm{OPT} - \|t - s\|$.*

The rest of this section is devoted to the proof of Theorem 4.2. Before introducing the construction and proof, let us present the intuition behind it. First let us rotate the space so that s, t both lie on the x-axis, with the x-coordinate of s smaller than the x-coordinate of t. Now suppose that the optimal path π^* is monotonically increasing in x. In this case, the optimal tour could be computed in quadratic time by a simple PSA, a dynamic programming algorithm defined by a plane orthogonal to the x-axis and sweeping across it from $x = -\infty$ to $x = \infty$. For every encountered point $p \in P$, the algorithm must determine the optimal tour from s to p visiting k' points for all $k' \in [k]$, and since all edges are x-monotone, all these k' points must have been encountered previously by the sweep. It follows that in this case the optimal k'-TSP tour ending at p can be computed by taking an optimal $(k'-1)$-TSP tour ending at each previously encountered point p', extending it to p by one edge of length $\|p - p'\|$, and then choosing among these tours (all choices of p') the one of minimum cost. This PSA ultimately computes the best k-TSP ending at each possible point, and the minimum among these is the optimal tour.

The difficulty with the above approach is that the optimal tour might be non-monotone in the x-coordinate. It may contain *backward* edges, while the PSA algorithm described above can only handle *forward* edges. However, for an edge facing backwards, its entire length accounts for excess in the tour. Hence, in the space (or more precisely, window) containing the backward edge, we can afford to run Arora's k-TSP algorithm, and pay $(1 + \delta)$ times the entire tour cost in the window. This motivates an algorithm which combines a sweep with Arora's k-TSP algorithm. We proceed with the actual proof, denoting the x-coordinate of a point p by $p[0]$.

This algorithm is similar to that of Blum et al. [BCK+07], which defines every window using only 2 points, and distinguishes between windows with only forward edges and windows that contain backward edges (called type-1 and type-2 in [BCK+07]). They use the MCP routine of [CGRT03] to approximate the minimum length of a path inside windows with backward edges, and stitch all the subpaths together by dynamic programming. There are two main differences in our algorithm. First, we run Arora's algorithm on the windows with backward edges. Second, our dynamic programming goes over the points in a different order. They order the points in increasing order of distance from s (the starting point), while we order the points by their x-coordinate.

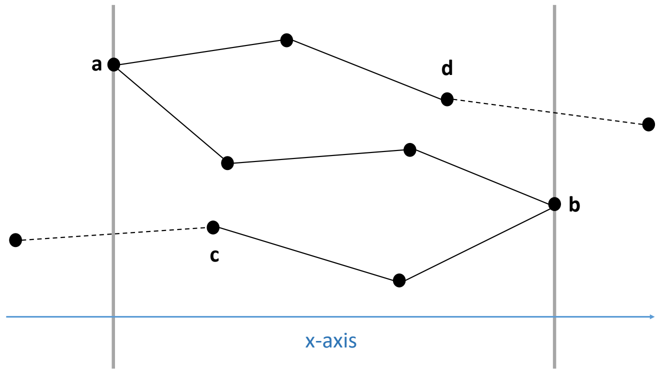


Figure 9: The window $\bar{w}_{a,b}$ is the space between the two thick gray lines, The solid black lines represent the subpath Arora$(a, b, c, d, 8)$, and the two dashed lines represent extending it outside that window.

Proof of Theorem 4.2. We rotate the space so that s, t lie on the x-axis, and then order all points based on increasing x-coordinate. For two distinct points $p, q \in P$, we say that p *is before* q, denoted $p < q$, if p's x-coordinate $p[0]$ is smaller than q's x-coordinate $q[0]$; otherwise, we say that q *is after* p, denoted $q > p$. We can make an infinitesimally small perturbation on the points to ensure that $p[0] \neq q[0]$ for all distinct points $p, q \in P$.

We define a *window* in $\mathbb{R}^d$ to be the space between (and including) two $(d-1)$-dimensional hyperplanes orthogonal to the x-axis. For every point pair $a, b \in P$ with $a \le b$ (which means we allow $a = b$), let $\bar{w}_{a,b}$ be a window of width $b[0] - a[0]$ containing a, b on its respective ends. Note that a window is bounded in the x direction and unbounded in all other directions, and also that a window may have width 0 (if $a = b$) and then it can contain at most one point of P. We denote the points of P contained in a window $\bar{w}_{a,b}$ by $P(\bar{w}_{a,b}) := \{p \in P \mid a \le p \le b\}$. Let $\mathcal{W} := \{\bar{w}_{a,b} \mid a, b \in P, a \le b\}$, and so $|\mathcal{W}| \le O(n^2)$.

Algorithm. For some $\delta' = \Theta(\delta)$ to be specified below, let Arora(a, b, c, d, k) be the output of Arora's $(1-\delta')$-approximate k-TSP algorithm on the set $P(\bar{w}_{a,b})$ and tour endpoints c, d with $a \le c, d \le b$; recall this algorithm returns the length of a near-optimal tour. For every $a, b, c, d \in P$ and $k' \in [k]$, we precompute Arora(a, b, c, d, k'), see Figure 9. We then order the points in P by their x-coordinate as $p_1, \dots, p_n$, and let $P_i := \{p \in P \mid p \le p_i\}$. The algorithm sweeps over $p_1, \dots, p_n$ (i.e., by their x-coordinate), and upon encountering point p_i, it calculates for every $k' \in [k]$ a path from s to p_i visiting k' points of P_i. However, the algorithm first precomputes approximate subpaths on many windows using Arora's algorithm, and thus the sweep is actually stitching these subpaths together into a global solution.

Let V be a 3-dimensional dynamic programming table with each entry $V(p_i, d, k')$ for $p_i \in P$, $d \in P_i$ and $k' \in [k]$, containing the length of an already computed path from s to d that visits k' points in P_i. To initialize the table, for all d, p_i satisfying $d \leq p_i = s$ and $k' \in [k]$, we fix entries $V(p_i, d, k') = \text{Arora}(p_1, p_i, s, d, k')$. All other entries are set to ∞. (Note that this forces all paths to begin at s, even if portions of those paths travel to the left of s.) The algorithm then considers each $p_i > s$ in increasing order, and calculates the entries for all d satisfying $s < d \leq p_i$ and all $k' \in [k]$, by choosing the shortest path among several possibilities, as follows.

$$V(p_i, d, k') = \min \Big\{ V(p_j, d', k'') + \|d' - c\| + \text{Arora}(p_{j+1}, p_i, c, d, k' - k'') \mid \\ p_j \in P_i,\ d' \in P_j,\ c \in P_i \setminus P_j,\ k'' < k' \Big\}$$

The path associated with this entry combines a previously computed path from s to d' visiting $k'' < k'$ points in P_j with an Arora subpath connecting endpoints c, d inside window $\bar{w}_{p_{j+1}, p_i}$ and visiting $k' - k''$ points in that window. Connecting these two paths using an edge (d', c) produces a path from s to d that visits k' points in P_i. After populating the table, the algorithm reports the entry $V(p_n, t, k)$.

The above algorithm computes the *length* of a path, but as usual it is extends easily to return also the path itself. It remains to prove that the returned path has length at most OPT $+ \delta \cdot \mathcal{E}$.

Correctness. We will show that there exists a solution of length at most OPT $+\delta \cdot \mathcal{E}$ that is considered by the dynamic program. Let π^* be an optimal path from s to t visiting k points of P, i.e., $|\pi^*| = \text{OPT}$. Given a path π and two points $p, q \in P(\pi)$, denote by $\pi(p, q)$ the subpath of π from p to q.

The solution produced by our PSA represents a set of windows connected by edges between them. However, some of these windows may be trivial and contain only a single point (and no edges), so in fact the PSA produces a solution which is a set of non-trivial windows connected by x-monotone subpaths (i.e., subpaths with only forward edges). As such, our analysis will similarly split π^* into windows and subpaths, where the windows contain all the backward edges of π^*, and the remaining edges constitute x-monotone subpaths. Assume that there are ℓ maximal backward subpaths in π^* (meaning that all edges of these subpaths face backwards), denoted $\pi^*(b_i, a_i)$ for $i \in [\ell]$, where $a_i < b_i$. Clearly, $\pi^*(b_i, a_i)$ is fully contained in window $\bar{w}_{a_i, b_i}$. Since these windows may overlap (have non-empty intersection), we repeatedly merge overlapping windows, i.e., replace any two overlapping windows $\bar{w}_{a_i, b_i}, \bar{w}_{a_j, b_j}$ by the united window $\bar{w}_{\min\{a_i, a_j\}, \max\{b_i, b_j\}}$, until no overlapping windows remain. We

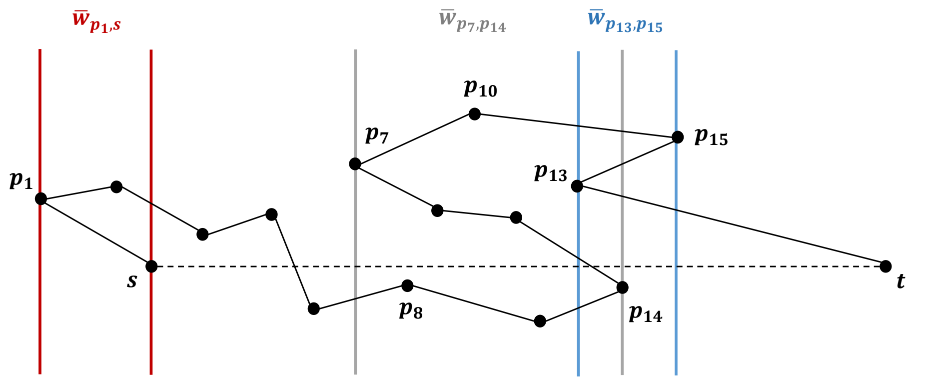


Figure 10: The solid black lines represent an optimal k-TSP path π^* between s and t for $k = 16$. It has 3 backward-facing subpaths $\pi^*(s, p_1)$, $\pi^*(p_{14}, p_7)$ and $\pi^*(p_{13}, p_{15})$. After merging overlapping windows $\bar{w}_{p_7,p_{14}}$ and $\bar{w}_{p_{13},p_{15}}$, we have $W^* = \{\bar{w}_{p_1,s}, \bar{w}_{p_7,p_{15}}\}$. The window $\bar{w}_{p_7,p_{15}}$ contains the subpath $\pi^*(p_8, p_{13})$, and thus $\mathcal{E}^{\text{win}}(\bar{w}_{p_7,p_{15}}, p_8, p_{13}) = \|\pi^*(p_8, p_{13})\| - |p_{13}[0] - p_8[0]|$.

---

thus assume henceforth that the l windows $\bar{w}_{a_i,b_i}$ are pairwise disjoint and denote $\mathcal{W}^* := \{\bar{w}_{a_i,b_i} \mid i \in [\ell]\}$. See Figure 10 for illustration.

Having merged overlapping windows, we have an ordered set of windows where every two successive windows are connected by an x-monotone subpath of π^*. Now for a window $\bar{w}_{a,b} \in \mathcal{W}^*$, let $c^*(\bar{w}_{a,b})$ and $d^*(\bar{w}_{a,b})$ be the entry and exit points of the optimal path π^* inside $\bar{w}_{a,b}$; notice these points are necessarily unique.

Recall that the excess of π^* is defined as $\mathcal{E}(\pi^*) = \|\pi^*\| - \|t - s\|$. Let E_f be all the edges in π^* that face forwards, and denote by $\|E_f\|$ their total length. Similarly, let E_b be all edges in π^* that face backwards, and denote by $\|E_b\|$ their total length. Clearly $\|E_f\| \geq \|t - s\|$, hence every edge that faces backwards contributes its entire length to the excess $\mathcal{E}(\pi^*)$, i.e.,

$$\mathcal{E}(\pi^*) = \|\pi^*\| - \|t - s\| = \|E_f\| + \|E_b\| - \|t - s\| \geq \|E_b\| \,. \tag{13}$$

We now define the excess of a window $\bar{w}_{a,b} \in \mathcal{W}^*$ with endpoints $c^* = c^*(\bar{w}_{a,b})$ and $d^* = d^*(\bar{w}_{a,b})$ to be

$$\mathcal{E}^{\text{win}}(\bar{w}_{a,b}) := \|\pi^*(c^*, d^*)\| - |d^*[0] - c^*[0]|,$$

which is non-negative because $\|\pi^*(c^*, d^*)\| \geq |b[0] - a[0]| \geq |d^*[0] - c^*[0]|$. Because the windows in $\mathcal{W}^*$ are pairwise disjoint, it is immediate that

$$\sum_{\bar{w}_{a,b}\in\mathcal{W}^*} \mathcal{E}^{\text{win}}(\bar{w}_{a,b}) \le \mathcal{E}(\pi^*). \tag{14}$$

see Figure 10 for illustration.

Applying Arora's k-TSP algorithm on window $\bar{w}_{a,b} \in \mathcal{W}^*$ with endpoints $c^* = c^*(\bar{w}_{a,b})$ and $d^* = d^*(\bar{w}_{a,b})$ returns a path of length at most $(1+\delta')\,\|\pi^*(c^*,d^*)\|$, and we would like to bound $\delta'\,\|\pi^*(c^*,d^*)\|$ relative to the excess $\mathcal{E}(\pi^*)$. To this end, we first bound it relative to backward edges and excess in that subpath/window, by claiming that

$$\|\pi^*(c^*,d^*)\| \le 2\max\left\{\,\|E_b\cap\pi^*(c^*,d^*)\|\,,\mathcal{E}^{\text{win}}(\bar{w}_{a,b})\right\}, \tag{15}$$

where $E_b\cap\pi^*(c,d)$ denotes the backward-facing edges in $\pi^*(c,d)$. Indeed, the claim holds trivially if $\|\pi^*(c^*,d^*)\| \le 2\,\|E_b\cap\pi^*(c^*,d^*)\|$, and otherwise we have $\|\pi^*(c^*,d^*)\| > 2\,\|E_b\cap\pi^*(c^*,d^*)\| \ge 2|d^*[0]-c^*[0]|$, and thus $\mathcal{E}^{\text{win}}(\bar{w}_{a,b}) = \|\pi^*(c^*,d^*)\| - |d^*[0]-c^*[0]| \ge \frac{1}{2}\,\|\pi^*(c^*,d^*)\|$, as claimed.

It follows that applying Arora's algorithm with $\delta' = \frac{1}{4}\delta$ on each of the non-overlapping windows in $\mathcal{W}^*$ will approximate the optimum $\|\pi^*\|$ within total additive error

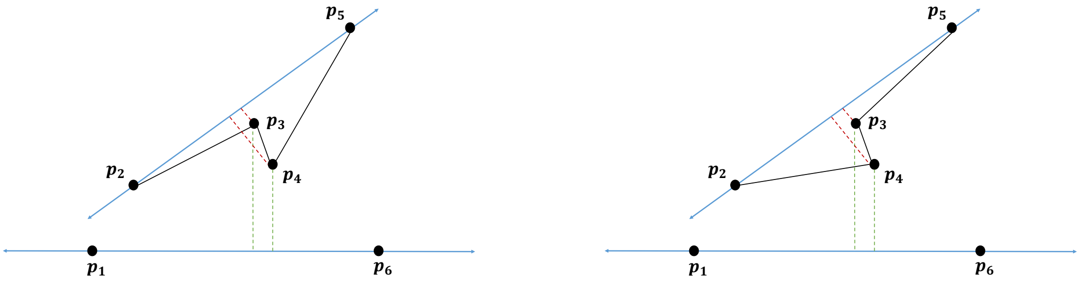
$$\begin{aligned}
&\sum_{\bar{w}_{a,b}\in\mathcal{W}^*} \delta'\cdot\|\pi^*(c^*(\bar{w}_{a,b}),d^*(\bar{w}_{a,b}))\| \\
&\qquad\le 2\delta' \sum_{\bar{w}_{a,b}\in\mathcal{W}^*} \left[\mathcal{E}^{\text{win}}(\bar{w}_{a,b}) + \|E_b\cap\pi^*(c^*(\bar{w}_{a,b}),d^*(\bar{w}_{a,b}))\|\,\right] && \text{by (15)}\\
&\qquad\le 2\delta'\left[\mathcal{E}(\pi^*) + \|E_b\|\,\right] \le 4\delta'\cdot\mathcal{E}(\pi^*) = \delta\cdot\mathcal{E} && \text{by (14) and (13).}
\end{aligned}$$

Our PSA is a dynamic program that optimizes over many combinations of windows, including the above collection $\mathcal{W}^*$, and thus the path that it returns, which can be only shorter, must be a δ-excess-approximation to the optimal k-TSP solution π^*.

Running Time. The table V has $O(n^3)$ entries, and computing each entry requires consulting $O(n^3)$ other entries. In addition, one has to invoke Arora's k-TSP algorithm $O(n^3)$ times, each executed in time $k^2 n(\log k)^{(d/\delta)^{O(d)}}$. Thus, the total running time is indeed $n^{O(1)}(\log k)^{(d/\delta)^{O(d)}}$. This completes the proof of Theorem 4.2. □

4.2.2 Algorithm for rooted (m,k)-TSP

Having shown how to construct a PSA for k-TSP, we extend this result to the more general (m,k)-TSP. We can prove the following theorem, which is the extension of Theorem 4.2 to multiple tours:



(a) (b)

Figure 11: Both figures consist of the same points $p_1, \ldots, p_6 \in R^d$, and the two hyperplanes h_1 and h_2 are determined by the points p_1, p_6 and p_2, p_5 correspondingly. Figures 11a and 11b differ by the path that connects the points p_2, p_3, p_4, p_5. In Figure 11a the edge (p_3, p_4) is a backward-facing edge with respect to h_2, while in Figure 11b the edge (p_4, p_3) is a backward-facing edge with respect to h_1.

Theorem 4.3. *There is an algorithm that, given as input m source-sink pairs $s_i, t_i \in \mathbb{R}^d$ for $i \in [m]$, a set of n points $P \subset \mathbb{R}^d$, an integer $1 \le k \le n$, and an accuracy parameter $\delta \in (0,1)$, runs in time $n^{O(m)}(\log n)^{(md/\delta)^{O(d)}}$ and reports m paths, one from each s_i to its corresponding t_i, that together visit k points of P and have total length at most $\|\Pi^*\| + \delta \cdot \mathcal{E}(\Pi^*)$, where Π^* is the minimum total length of m such paths.*

As before, we first provide the construction, and then demonstrate correctness. The construction closely parallels that of the PSA for k-TSP, being a collection of x-monotone multi-paths connecting windows.

For points $s, t \in \mathbb{R}^d$, let st be the directed line segment connecting them. Let the *angle* of st be the angle of its direction vector to the x-axis. Given a path π with endpoints s, t, we define the angle of π to be the angle of the vector st.

Given a set Π of m paths with respective endpoints s_i, t_i for $i \in [m]$ the space may be rotated and (if necessary) some values s_i, t_i swapped to ensure that in the resulting space each directed path has angle in the range $[0, \frac{\boldsymbol{\pi}}{2} - \frac{1}{m'}]$,[5] where $m' = 8m^{3/2}$ (see Lemma 4.6 in Section 4.4). We execute this rotation step before the run of the multi-path PSA. The crux is that this direction is "good enough" for each of the m paths, meaning that it is effective as a sweep direction for all the m paths simultaneously. In contrast, the ordering of Blum et al. [BCK+07] (according to distances from a starting point s) must use a single starting point and does not extend to multiple paths.

[5] We use $\boldsymbol{\pi}$ to denote the mathematical constant, and π to denote a path.

Construction. The construction handles m paths simultaneously. Let S, T be arrays of length m, with entries $S[j], T[j]$ corresponding to the source-sink pair of the j-th path. Let a window be defined as in Section 4.2.1. For some $\delta' = \Omega_m(\delta)$ to be specified below, let $\text{Arora}(a, b, S, T, k)$ be the output of Arora's $(1-\delta')$-approximate (m, k)-TSP algorithm on the set $P(\bar{w}_{a,b})$ and tour endpoint arrays S, T. (We may assume for simplicity that $a \leq S[j], T[j] \leq b$ for all j. If both $S[j], T[j]$ are null, the algorithm will ignore the j-th path. If exactly one is null, the algorithm with return ∞.) For every $a, b \in P$, $S, T \subset P^m$ and $k' \in [k]$, we precompute $\text{Arora}(a, b, S, T, k')$. The algorithm then sweeps the x-axis from left to right as before to calculate the solution to subproblems up to a point $p_i \in P$.

Similarly to what was done above, let V be a 4-dimensional lookup table with an entry $V(p_i, S, T, k')$ for every $p_i \in P$, $S, T \in P_i^m$ and $k' \in [k]$, that contains the length of computed paths from sources S to sinks T that together visit k' points in P_i. For the initialization, we add a dummy point p_0 to P and initialize the single entry $V(p_0, S, T, 0)$, where arrays S, T contain all null points, to be 0. We initialize all other table entries to ∞. Define the distance from a point to a null point (or between two null points) to be 0. The algorithm considers each $p_i \in P$ $(i \geq 0)$ in increasing order, and computes the entries for all $S, T \in P_i^m$ by choosing the shortest path among several possibilities, as follows:

$$V(p_i, S, T, k') = \min \Big\{ V(p_j, S_1, T_1, k'') + \sum_{l=1}^{m} \|T_1[l] - S_2[l]\| + \text{Arora}(p_{j+1}, p_i, S_2, T_2, k' - k'') \mid \\ p_j \in P_i,\ S_1, T_1 \in P_j^m,\ S_2, T_2 \in (P_i \setminus P_j)^m,\ k'' < k' \Big\}$$

As before, this computation combines the length of a previously computed approximated shortest path to a new Arora multi-path. However, we add to the above description feasiblity requirements, which are sufficient to ensure the validity of the final tour. First note that the invocation to Arora's algorithm on arrays S, T ensures that a non-infinite solution is possible only if $S[j], T[j]$ are both null or both non-null for all j. We further require for S, S_1, S_2 and all j that $S_1[j] = S[j]$, unless $S_1[j]$ is null, in which case we require that $S_2[j] = S[j]$. This ensures that the computed subtour has source $S[j]$. Likewise, we require for T, T_1, T_2 and all j that $T_2[j] = T[j]$, unless $T_2[j]$ is null, in which case we require that $T_1[j] = T[j]$. This ensures that the computed subtour has sink $T[j]$. Also, if $S[j]$ $(T[j])$ is null, then $S_1[j], S_2[j]$ $(T_1[j], T_2[j])$ must be null as well. This ensures that the subproblems do not feature additional tours. If these requirements are not met for some set $\{S, S_1, S_2, T, T_1, T_2\}$, then the table value is not changed. After populating the table, the algorithm reports the entry $V(p_n, S, T, k)$ for S, T containing the sources and sinks of the master problem.

Correctness. We must show that there is a set of paths Π with $\|\Pi\| \leq \|\Pi^*\| + \mathcal{E}(\Pi^*)$ which can be found by the above algorithm. As before, it suffices to show that the optimal solution can be divided into windows connected by x-monotone paths.

In the analysis of the k-TSP algorithm, we used the fact that any backward-facing edge contributes its entire length to the excess. This does not hold in the (m, k)-TSP case, as the definition of "backwards" remains with respect to the x-axis, but excess is measured with respect to the angle of the relevant path, see Figure 11. To address this, we will require the following lemma:

Lemma 4.4. *Given parameter $0 \leq \gamma \leq 1$ (where γ is a measure in radians) and directed path π with angle ϕ to the x-axis and edge-set E, let $E' \subset E$ consist of directed edges with angles in the range $[\phi - \gamma, \phi + \gamma]$. Then we have*

$$\sum_{e \in E \setminus E'} \|e\| \leq \frac{24}{11\gamma^2} \mathcal{E}(\pi).$$

Proof. For edge $e \in E$, let $p(e)$ be the length of the projection of e onto segment st, where s, t are the endpoints of π. We can charge each edge e a share of the excess as follows. Define $\mathcal{E}(e) = \|e\| - p(e)$; then by the triangle inequality $\|s - t\| \leq \sum_{e \in E} p(e)$, and thus $\sum_{e \in E} \mathcal{E}(e) = \|\pi\| - \sum_{e \in E} p(e) \leq \mathcal{E}(\pi)$. Now consider an edge $e \in E \setminus E'$. Recalling the Taylor expansion $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \ldots$, we have that

$$p(e) \leq \|e\| \cos(\gamma) \leq \|e\| \left(1 - \frac{\gamma^2}{2} + \frac{\gamma^4}{24}\right) \leq \|e\| \left(1 - \frac{11}{24}\gamma^2\right),$$

and so $\mathcal{E}(e) = \|e\| - p(e) \geq \frac{11}{24}\gamma^2 \|e\|$. It follows that

$$\frac{11\gamma^2}{24} \sum_{e \in E \setminus E'} \|e\| \leq \sum_{e \in E \setminus E'} \mathcal{E}(e) \leq \mathcal{E}(\pi).$$

□

Now take optimal tour Π^*, and let $\mathcal{W}^*$ be defined as in Section 4.2.1, that is consisting of mergers of maximal windows which together cover all backward-facing edges (where the direction is defined with respect to the x-axis). The edges not in windows of $\mathcal{W}^*$ constitute forward-facing paths. Now consider some window $\bar{w}_{a,b} \in \mathcal{W}^*$, and we will show that we can afford to run Arora's (m, k)-TSP on this window with sufficiently small parameter δ'.

We begin with the set E_b of backward-facing edges of Π^*. As the angle of all paths in Π^* was shown above to be in the range $[0, \frac{\pi}{2} - \frac{1}{m'}]$ radians (and backward-facing edges necessarily have angle greater than $\frac{\pi}{2}$) we can apply Lemma 4.4 with parameter $\gamma = \frac{1}{m'}$, and conclude that

$$\|E_b\| \leq \frac{24(m')^2}{11}\mathcal{E}(\Pi^*) = O(m^3) \cdot \mathcal{E}(\Pi^*).$$

We now turn to the set E_f of forward-facing edges of Π^*. Set $E'_f \subset E_f$ will contain edges whose angle is very close to the angle of their path. More precisely, let path π_i^* have angle ϕ_i. E'_f includes every edge of every path π_i^* with angle in the range $\left[\phi_i - \frac{1}{2m'}, \phi_i + \frac{1}{2m'}\right]$. Now consider edges of $E_f \setminus E'_f$: Applying Lemma 4.4 with parameter $\gamma = \frac{1}{2m'}$, we have that

$$\left\|E_f \setminus E'_f\right\| \leq \frac{24(2m')^2}{11}\mathcal{E}(\Pi^*) = O(m^3) \cdot \mathcal{E}(\Pi^*).$$

Finally, we now turn to the set E'_f. First recall that by the Taylor expansion, $\sin(x) = x - \frac{x^3}{3!} + \ldots$ Each edge in E'_f accounts for a progression in the x-direction of at least

$$\|e\| \cos\left(\frac{\pi}{2} - \frac{1}{2m'}\right) = \|e\| \sin\left(\frac{1}{2m'}\right) \geq \|e\| \left(\frac{1}{2m'} - \frac{1}{6(2m')^3}\right) > \|e\| \frac{1}{3m'}.$$

Now consider window $\bar{w}_{a,b} \in \mathcal{W}^*$, and its associated paths $\pi_i^*(c_i^*(\bar{w}_{a,b}), d_i^*(\bar{w}_{a,b}))$. Clearly $\pi_i^*(c_i^*(\bar{w}_{a,b}), d_i^*(\bar{w}_{a,b}))$ cannot progress in the x-direction inside the window for more than its length without heading backwards, and so

$$\frac{1}{3m'}\left\|\pi_i^*(c_i^*(\bar{w}_{a,b}), d_i^*(\bar{w}_{a,b})) \cap E'_f\right\| \leq |b[0] - a[0]| + \left\|\pi_i^*(c_i^*(\bar{w}_{a,b}), d_i^*(\bar{w}_{a,b})) \cap E_b\right\|.$$

Now recall that by construction, each window $\bar{w}_{a,b} \in \mathcal{W}^*$ contains backward-facing edges whose lengths sum to at least the window length, and so $\sum_{\bar{w}_{a,b}\in\mathcal{W}^*} |b[0]-a[0]| \leq \|E_b\|$. Applying Arora's algorithm with parameter δ' on each of the non-overlapping windows in $\mathcal{W}^*$ will approximate the optimum $\|\Pi^*\|$ within total additive error

$$\begin{aligned}
&\sum_{\bar{w}_{a,b}\in\mathcal{W}^*} \sum_{i=1}^{m} \delta' \left\|\pi_i^*(c_i^*(\bar{w}_{a,b}), d_i^*(\bar{w}_{a,b}))\right\| \\
&\quad\leq \delta'\Big(\|E_b\| + \left\|E_f \setminus E'_f\right\| + \sum_{\bar{w}_{a,b}\in\mathcal{W}^*} \sum_{i=1}^{m} \left\|E'_f \cap \pi_i^*(c_i^*(\bar{w}_{a,b}), d_i^*(\bar{w}_{a,b}))\right\| \Big) \\
&\quad\leq \delta'\Big(\|E_b\| + \left\|E_f \setminus E'_f\right\| + \sum_{\bar{w}_{a,b}\in\mathcal{W}^*} \sum_{i=1}^{m} 3m'\Big[|b[0] - a[0]| + \left\|E_b \cap \pi_i^*(c_i^*(\bar{w}_{a,b}), d_i^*(\bar{w}_{a,b}))\right\|\Big]\Big) \\
&\quad\leq \delta'\Big(\|E_b\| + \left\|E_f \setminus E'_f\right\| 3m^{2.5} \cdot 2\left\|E_b\right\| \Big) \\
&\quad= O(\delta' m^{5.5}\mathcal{E}(\Pi^*)).
\end{aligned}$$

So we can afford to execute Arora's (m, k)-TSP algorithm with parameter $\delta' = \frac{c\delta}{m^{5.5}}$ for suitable constant $c > 1$.

Running Time. The table V has $n^{O(m)}$ entries, and computing each entry requires consulting $n^{O(m)}$ other entries. In addition, one has to invoke $n^{O(m)}$ times Arora's k-TSP algorithm (modified as explained in Section 4.1.3 to find m tours), and each of these is executed in time $k^2 n(2^m \log n)^{(d/\delta')^{O(d)}}$. Plugging our $\delta' = \frac{c\delta}{m^{5.5}}$ yields total running time $n^{O(m)}(\log n)^{(md/\delta)^{O(d)}}$, which completes the proof of Theorem 4.3.

4.3 A PTAS for Orienteering

Having shown in Theorem 4.3 how to compute a δ-excess-approximation to an optimal (m, k)-TSP tour, we can use this algorithm as a subroutine to solve the orienteering problem. As in [CH08], we show how to reduce the orienteering problem to $n^{O(1/\delta)}$ instances of the $(O(1/\delta), k)$-TSP problem.

Lemma 4.5. *A $(1-\delta)$-approximation to orienteering problem on n-point set P, a budget $\mathcal{B}$ and a starting point s, can be computed by making $n^{O(1/\delta)}$ queries to an $O(\delta)$-excess-approximation oracle for (m, k)-TSP, with parameters $m = O(1/\delta)$ and $k = O(k_{\text{opt}})$, where k_{opt} denotes the number of points visited by an optimal path.*

Then Theorem 4.1 follows from Lemma 4.5, with oracle queries executed by the algorithm of Theorem 4.3.

Proof. Let π^* be an optimal rooted orienteering path starting at s of length at most $\mathcal{B}$ that visits k_{opt} points of P, let $\pi^*(i, j) = \langle p_i, \dots, p_j \rangle$ be the portion of the path π^* from p_i to p_j, and let $\mathcal{E}(i, j) = \|\pi^*(i, j)\| - \|p_i - p_j\|$ be its excess. Set $m = \lfloor 1/\delta \rfloor$, and let $\alpha_i = \lceil (i-1)(k_{\text{opt}} - 1)/m \rceil + 1$ for every $1 \le i \le m+1$. By definition, we have $\alpha_1 = 1$ and $\alpha_{m+1} = k_{\text{opt}}$. Furthermore, each subpath $\pi^*(\alpha_i, \alpha_{i+1})$ visits

$$\alpha_{i+1} - \alpha_i - 1 = (\lceil i(k_{\text{opt}} - 1)/m \rceil + 1) - (\lceil (i-1)(k_{\text{opt}} - 1)/m \rceil + 1) - 1 \le \lfloor (k_{\text{opt}} - 1)/m \rfloor$$

points, excluding the endpoints p_{α_i} and $p_{\alpha_{i+1}}$.

Consider the subpaths $\pi^*(\alpha_1, \alpha_2), \dots, \pi^*(\alpha_m, \alpha_{m+1})$ of π^* and their respective excesses

$$\mathcal{E}_1 = \mathcal{E}(\alpha_1, \alpha_2), \dots, \mathcal{E}_m = \mathcal{E}(\alpha_m, \alpha_{m+1}).$$

Clearly, there exists an index ν, $1 \le \nu \le m$, such that $\mathcal{E}_\nu \ge \frac{1}{m}(\sum_{i=1}^m \mathcal{E}_i)$. By connecting the vertex p_{α_ν} directly to the vertex $p_{\alpha_{\nu+1}}$ in π^*, we obtain a new path

$\pi' = \langle p_1, p_2 \ldots, p_{\alpha_\nu}, p_{\alpha_{\nu+1}}, p_{\alpha_{\nu+1}+1}, \ldots, p_{k_{\text{opt}}} \rangle$. Observe that $\|\pi'\| = \|\pi^*\| - \mathcal{E}_\nu$, and as noted above, π' visits at least

$$k_{\text{opt}} - (\alpha_{\nu+1} - \alpha_\nu - 1) \geq k_{\text{opt}} - \lfloor (k_{\text{opt}} - 1)/m \rfloor \geq (1 - 1/m) k_{\text{opt}}$$

points of P.

By applying an (m, k_{opt})-TSP oracle on the m pairs $(s_i = p_{\alpha_i}, t_i = p_{\alpha_{i+1}})$ for every $i \in [m]$ with accuracy parameter $1/m$, one can compute a path $\hat{\pi}$ that visits at least $(1 - 1/m)k_{\text{opt}} \geq (1 - \delta)k_{\text{opt}}$ points of P, of length

$$\begin{aligned} \|\hat{\pi}\| &\leq \sum_{i=1}^{m} (\|\pi'(\alpha_i, \alpha_{i+1})\| + \frac{1}{m}\mathcal{E}_i) \\ &\leq \|\pi'\| + \mathcal{E}_\nu \\ &= \|\pi^*\| \\ &\leq \mathcal{B}. \end{aligned}$$

As the value of k_{opt} is not known in advance, the algorithm tries all possible values of k from 1 to n, returning the maximum value k' for which it finds a tour within budget $\mathcal{B}$ (that is, the algorithm terminates at the failed attempt to find a tour visiting $k'+1$ points). As we proved above, $(1 - \delta)k_{\text{opt}} \leq k' \leq k_{\text{opt}}$. In addition, since we do not know the optimal orienteering path π^* in advance, we guess the $m = \lfloor 1/\delta \rfloor$ points p_{α_i}, which gives $n^{O(1/\delta)}$ queries of (m, k)-TSP. □

4.4 Appendix

Lemma 4.6. *For every unit-length vectors $v_1, \ldots, v_m \in \mathbb{R}^d$, there are signs $\sigma_1, \ldots, \sigma_m \in \{\pm 1\}$ and a unit-length $x \in \mathbb{R}^d$ (direction in space) such that*

$$\forall i \in [m], \quad \langle x, \sigma_i v_i \rangle \geq \frac{1}{8m^{3/2}} \quad \textit{and thus} \quad 0 \leq \text{angle}(x, \sigma_i v_i) \leq \frac{\pi}{2} - \frac{1}{8m^{3/2}}.$$

We actually prove that the inner product is at least $\frac{1}{8m\sqrt{d}}$, and arguing that without loss of generality $d \leq m$, the stated bound follows.

Proof. Let $x \in \mathbb{R}^d$ be a random vector where each entry is an iid Gaussian $N(0, 1)$, i.e., chosen from the distribution $x \sim N(0, I_d)$. Then by Markov's inequality $\Pr[\|x\|^2 \geq 4d] \leq 1/4$. Now fix $i \in [m]$. The inner product $\langle x, v_i \rangle$ has the same distribution as a standard Gaussian $g \sim N(0, 1)$, for which elementary observations about its pdf (like the monotonicity) show that

$$\forall \gamma \in (0, 1), \quad \Pr_g \big(|g| \in [0, \gamma]\big) \leq e^{1/2} \cdot \Pr_g \big(|g| \in [1 - \gamma, 1]\big) \leq e^{1/2} \cdot \tfrac{1}{\gamma}.$$

Plugging $\gamma = \frac{1}{4m}$, we have that

$$\Pr_x \left(|\langle x, v_i \rangle| \leq \frac{1}{4m} \right) < \frac{1}{2m}.$$

Now applying a union bound over these $m+1$ events (one about $\|x\|^2$ and one for each i), we see there positive probability that all these events fail, the vector $y = x/\|x\|$ is a unit-length vector (in same direction as x) and satisfies

$$\forall i \in [m], \quad |\langle y, v_i \rangle| > \frac{1}{4m \cdot \|x\|} \geq \frac{1}{8m\sqrt{d}}.$$

Finally, for each $i \in [m]$ we can pick a sign $\sigma_i \in \{\pm 1\}$ such that $\langle y, \sigma_i v_i \rangle$ is non-negative, and then $\langle y, \sigma_i v_i \rangle = |\langle y, v_i \rangle|$. We may assume that $d \leq m$, as otherwise we can restrict attention to the span of the vectors, and conclude the required $\langle y, \sigma_i v_i \rangle \geq \frac{1}{8m^{3/2}}$.

To bound the angle, let $\theta_i := \frac{\pi}{2} - \text{angle}(y, \sigma_i v_i)$ and then $\frac{1}{8m^{3/2}} \leq \langle y, \sigma_i v_i \rangle \leq \cos(\frac{\pi}{2} - \theta_i) = \sin(\theta_i) \leq \theta_i$, where the last inequality relies on observing that $\theta_i \in [0, \frac{\pi}{2}]$. $\square$

References

[AA97] B. Awerbuch and Y. Azar. Buy-at-bulk network design. In *38th Annual Symposium on Foundations of Computer Science*, FOCS '97, pages 542–547. IEEE Computer Society, 1997. doi:10.1109/SFCS.1997.646143.

[AFGN18] I. Abraham, A. Filtser, A. Gupta, and O. Neiman. Metric embedding via shortest path decompositions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 952–963. ACM, 2018. doi:10.1145/3188745.3188808.

[AGK14] A. Andoni, A. Gupta, and R. Krauthgamer. Towards $(1+\epsilon)$-approximate flow sparsifiers. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 279–293, 2014. doi:10.1137/1.9781611973402.20.

[AMN98] E. M. Arkin, J. S. B. Mitchell, and G. Narasimhan. Resource-constrained geometric network optimization. In *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*, pages 307–316, 1998. doi:10.1145/276884.276919.

[AR98] Y. Aumann and Y. Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM J. Comput.*, 27(1):291–301, 1998. doi:10.1137/S0097539794285983.

[Aro98] S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998. doi:10.1145/290179.290180.

[ASV14] C. Archetti, M. G. Speranza, and D. Vigo. Chapter 10: Vehicle routing problems with profits. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 273–297. SIAM, 2014. doi:10.1137/1.9781611973594.ch10.

[Bal95] E. Balas. The prize collecting traveling salesman problem: II. polyhedral results. *Networks*, 25(4):199–216, 1995. doi:10.1002/net.3230250406.

[BBCM04] N. Bansal, A. Blum, S. Chawla, and A. Meyerson. Approximation algorithms for deadline-TSP and vehicle routing with time-windows. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 166–174, 2004. doi:10.1145/1007352.1007385.

[BCK+07] A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, and M. Minkoff. Approximation algorithms for orienteering and discounted-reward TSP. *SIAM J. Comput.*, 37(2):653–670, 2007. doi:10.1137/050645464.

[Ben09] C. Bentz. A simple algorithm for multicuts in planar graphs with outer terminals. *Discrete Appl. Math.*, 157(8):1959–1964, April 2009. doi:10.1016/j.dam.2008.11.010.

[Ber90] M. W. Bern. Faster exact algorithms for Steiner trees in planar networks. *Networks*, 20(1):109–120, 1990. doi:10.1002/net.3230200110.

[BG08] A. Basu and A. Gupta. Steiner point removal in graph metrics. Unpublished Manuscript, available from http://www.math.ucdavis.edu/~abasu/papers/SPR.pdf, 2008.

[BK96] A. A. Benczúr and D. R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, STOC 1996*, pages 47–55, 1996. doi:10.1145/237814.237827.

[BKK07] G. Borradaile, C. Kenyon-Mathieu, and P. N. Klein. A polynomial-time approximation scheme for Steiner tree in planar graphs. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 1285–1294, 2007. Available from: http://dl.acm.org/citation.cfm?id=1283383.1283521.

[BKM09] G. Borradaile, P. N. Klein, and C. Mathieu. An $O(n \log n)$ approximation scheme for Steiner tree in planar graphs. *ACM Trans. Algorithms*, 5(3):31:1–31:31, 2009. doi:10.1145/1541885.1541892.

[BLS10] G. Borradaile, J. R. Lee, and A. Sidiropoulos. Randomly removing g handles at once. *Comput. Geom.*, 43(8):655–662, 2010. doi:10.1016/j.comgeo.2010.04.007.

[BM88] D. Bienstock and C. L. Monma. On the complexity of covering vertices by faces in a planar graph. *SIAM J. Comput.*, 17(1):53–76, February 1988. doi:10.1137/0217004.

[Bor04] G. Borradaile. *Exploiting Planarity for Network Flow and Connectivity Problems*. PhD thesis, Brown University, 2004.

[CGH16] Y. K. Cheung, G. Goranci, and M. Henzinger. Graph minors for preserving terminal distances approximately - lower and upper bounds. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP*, pages 131:1–131:14, 2016. doi:10.4230/LIPIcs.ICALP.2016.131.

[CGRT03] K. Chaudhuri, B. Godfrey, S. Rao, and K. Talwar. Paths, trees, and minimum latency tours. In *44th Symposium on Foundations of Computer Science FOCS 2003*, pages 36–45, 2003. doi:10.1109/SFCS.2003.1238179.

[CH08] K. Chen and S. Har-Peled. The Euclidean orienteering problem revisited. *SIAM J. Comput.*, 38(1):385–397, 2008. doi:10.1137/060667839.

[Che18] Y. K. Cheung. Steiner point removal - distant terminals don't (really) bother. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018*, pages 1353–1360, 2018. doi:10.1137/1.9781611975031.89.

[Chu12] J. Chuzhoy. On vertex sparsifiers with Steiner nodes. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012*, pages 673–688, 2012. doi:10.1145/2213977.2214039.

[CK04] C. Chekuri and A. Kumar. Maximum coverage problem with group budget constraints and applications. In *Approximation, Randomization, and Combinatorial Optimization, Algorithms and Techniques, 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2004, and 8th International Workshop on Randomization and Computation, RANDOM 2004, Cambridge,*, pages 72–83, 2004. doi:10.1007/978-3-540-27821-4_7.

[CKP12] C. Chekuri, N. Korula, and M. Pál. Improved algorithms for orienteering and related problems. *ACM Trans. Algorithms*, 8(3):23:1–23:27, 2012. doi:10.1145/2229163.2229167.

[CKRV15] C. Chekuri, S. Kannan, A. Raja, and P. Viswanath. Multicommodity flows and cuts in polymatroidal networks. *SIAM J. Comput.*, 44(4):912–943, 2015. doi:10.1137/130906830.

[CLLM10] M. Charikar, T. Leighton, S. Li, and A. Moitra. Vertex sparsifiers and abstract rounding algorithms. In *51st Annual Symposium on Foundations of Computer Science*, pages 265–274. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.32.

[CP05] C. Chekuri and M. Pál. A recursive greedy algorithm for walks in directed graphs. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 245–253, 2005. doi:10.1109/SFCS.2005.9.

[CSW13] C. Chekuri, F. B. Shepherd, and C. Weibel. Flow-cut gaps for integer and fractional multiflows. *Journal of Combinatorial Theory, Series B*, 103(2):248 – 273, 2013. doi:10.1016/j.jctb.2012.11.002.

[CSWZ00] S. Chaudhuri, K. V. Subrahmanyam, F. Wagner, and C. D. Zaroliagis. Computing mimicking networks. *Algorithmica*, 26:31–49, 2000. doi:10.1007/s004539910003.

[CW04] D. Z. Chen and X. Wu. Efficient algorithms for k-terminal cuts on planar graphs. *Algorithmica*, 38(2):299–316, Feb 2004. doi:10.1007/s00453-003-1061-2.

[CX00] D. Z. Chen and J. Xu. Shortest path queries in planar graphs. In *32nd Annual ACM Symposium on Theory of Computing*, STOC '00, pages 469–478. ACM, 2000. doi:10.1145/335305.335359.

[CXKR06] H. T. Chan, D. Xia, G. Konjevod, and A. W. Richa. A tight lower bound for the Steiner point removal problem on trees. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2006 and 10th International Workshop on Randomization and Computation, RANDOM 2006*, pages 70–81, 2006. doi:10.1007/11830924_9.

[EGK$^+$14] M. Englert, A. Gupta, R. Krauthgamer, H. Räcke, I. Talgam-Cohen, and K. Talwar. Vertex sparsifiers: New results from old techniques. *SIAM Journal on Computing*, 43(4):1239–1262, 2014. doi:10.1137/130908440.

[EMV87] R. E. Erickson, C. L. Monma, and A. F. Veinott. Send-and-split method for minimum-concave-cost network flows. *Math. Oper. Res.*, 12(4):634–664, 1987. doi:10.1287/moor.12.4.634.

[FF56] L. Ford and D. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[FHL05] U. Feige, M. T. Hajiaghayi, and J. R. Lee. Improved approximation algorithms for minimum-weight vertex separators. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 563–572, 2005. doi:10.1145/1060590.1060674.

[Fil18] A. Filtser. Steiner point removal with distortion $O(\log k)$. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018*, pages 1361–1373, 2018. doi:10.1137/1.9781611975031.90.

[Fil20] A. Filtser. A face cover perspective to ℓ_1 embeddings of planar graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2020*, pages 1945–1954. SIAM, 2020. doi:10.1137/1.9781611975994.120.

[FKT19] A. Filtser, R. Krauthgamer, and O. Trabelsi. Relaxed Voronoi: A simple framework for terminal-clustering problems. In *2nd Symposium on Simplicity in Algorithms, SOSA@SODA 2019*, volume 69 of *OASICS*, pages 10:1–10:14, 2019. doi:10.4230/OASIcs.SOSA.2019.10.

[FM95] T. Feder and R. Motwani. Clique partitions, graph compression and speeding-up algorithms. *J. Comput. Syst. Sci.*, 51(2):261–272, 1995.

[Fre91] G. N. Frederickson. Planar graph decomposition and all pairs shortest paths. *J. ACM*, 38(1):162–204, January 1991. doi:10.1145/102782.102788.

[Fre95] G. N. Frederickson. Using cellular graph embeddings in solving all pairs shortest paths problems. *Journal of Algorithms*, 19(1):45 – 85, 1995. doi:10.1006/jagm.1995.1027.

[FRT04] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.*, 69(3):485–497, 2004. doi:10.1016/j.jcss.2004.04.011.

[FS17] Z. Friggstad and C. Swamy. Compact, provably-good lps for orienteering and regret-bounded vehicle routing. In *Integer Programming and Combinatorial Optimization - 19th International Conference, IPCO 2017, Proceedings*, volume 10328 of *Lecture Notes in Computer Science*, pages 199–211. Springer, 2017. doi:10.1007/978-3-319-59250-3_17.

[GGJ76] M. R. Garey, R. L. Graham, and D. S. Johnson. Some NP-complete geometric problems. In *Proceedings of the 8th Annual ACM Symposium on Theory of Computing*, pages 10–22, 1976. doi:10.1145/800113.803626.

[GHP17] G. Goranci, M. Henzinger, and P. Peng. Improved guarantees for vertex sparsification in planar graphs. In *25th Annual European Symposium on Algorithms, ESA 2017*, volume 87 of *LIPIcs*, pages 44:1–44:14, 2017. doi:10.4230/LIPIcs.ESA.2017.44.

[GKNR15] A. Gupta, R. Krishnaswamy, V. Nagarajan, and R. Ravi. Running errands in time: Approximation algorithms for stochastic orienteering. *Math. Oper. Res.*, 40(1):56–79, 2015. doi:10.1287/moor.2014.0656.

[GKR00] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *J. Algorithms*, 37(1):66–84, 2000. doi:10.1006/jagm.2000.1096.

[GKR20] L.-A. Gottlieb, R. Krauthgamer, and H. Rika. Faster algorithms for orienteering and k-TSP. 2020. arXiv:2002.07727.

[GLV87] B. L. Golden, L. Levy, and R. Vohra. The orienteering problem. *Naval Research Logistics (NRL)*, 34(3):307–318, 1987. doi:10.1002/1520-6750(198706)34:3<307::AID-NAV3220340302>3.0.CO;2-D.

[GLV16] A. Gunawan, H. C. Lau, and P. Vansteenwegen. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332, 2016. doi:10.1016/j.ejor.2016.04.059.

[GNR10] A. Gupta, V. Nagarajan, and R. Ravi. An improved approximation algorithm for requirement cut. *Oper. Res. Lett.*, 38(4):322–325, 2010. doi:10.1016/j.orl.2010.02.009.

[GNRS04] A. Gupta, I. Newman, Y. Rabinovich, and A. Sinclair. Cuts, trees and l_1-embeddings of graphs. *Combinatorica*, 24(2):233–269, 2004. doi:10.1007/s00493-004-0015-x.

[GR16] G. Goranci and H. Räcke. Vertex sparsification in trees. In *Approximation and Online Algorithms - 14th International Workshop, WAOA*, pages 103–115, 2016. doi:10.1007/978-3-319-51741-4_9.

[Gup01] A. Gupta. Steiner points in tree metrics don't (really) help. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA.*, pages 220–227, 2001. Available from: http://dl.acm.org/citation.cfm?id=365411.365448.

[HKNR98] T. Hagerup, J. Katajainen, N. Nishimura, and P. Ragde. Characterizing multiterminal flow networks and computing flows in networks of small treewidth. *J. Comput. Syst. Sci.*, 57(3):366–375, 1998. doi:10.1006/jcss.1998.1592.

[IR82] A. Itai and M. Rodeh. Representation of graphs. *Acta Inf.*, 17:215–219, 1982. doi:10.1007/BF00288971.

[Kar93] D. R. Karger. Global min-cuts in rnc, and other ramifications of a simple min-cut algorithm. In *Proceedings of the Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, SODA*, pages 21–30, 1993. Available from: http://dl.acm.org/citation.cfm?id=313559.313605.

[KKN15] L. Kamma, R. Krauthgamer, and H. L. Nguyen. Cutting corners cheaply, or how to remove Steiner points. *SIAM Journal on Computing*, 44(4):975–995, 2015. doi:10.1137/140951382.

[Kle06] P. N. Klein. A subset spanner for planar graphs, : with application to subset TSP. pages 749–756, 2006. doi:10.1145/1132516.1132620.

[KLL02] T. Kloks, C. Lee, and J. Liu. New algorithms for k-face cover, k-feedback vertex set, and k-disjoint cycles on plane and planar graphs. In *Graph-Theoretic Concepts in Computer Science, 28th International Workshop, WG 2002, Revised Papers*, pages 282–295, 2002. doi:10.1007/3-540-36379-3_25.

[KLR19] R. Krauthgamer, J. R. Lee, and H. Rika. Flow-cut gaps and face covers in planar graphs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 525–534, 2019. doi:10.1137/1.9781611975482.33.

[KNvL19] S. Kisfaludi-Bak, J. Nederlof, and E. J. van Leeuwen. Nearly eth-tight algorithms for planar Steiner tree with terminals on few faces. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1015–1034, 2019. doi:10.1137/1.9781611975482.63.

[KNZ14] R. Krauthgamer, H. L. Nguyen, and T. Zondiner. Preserving terminal distances using minors. *SIAM J. Discrete Math.*, 28(1):127–141, 2014. doi:10.1137/120888843.

[KPR93] P. Klein, S. A. Plotkin, and S. Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pages 682–690, 1993. doi:10.1145/167088.167261.

[KPZ19] N. Karpov, M. Pilipczuk, and A. Zych-Pawlewicz. An exponential lower bound for cut sparsifiers in planar graphs. *Algorithmica*, 81(10):4029–4042, 2019. doi:10.1007/s00453-018-0504-8.

[KR13] R. Krauthgamer and I. Rika. Mimicking networks and succinct representations of terminal cuts. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013*, pages 1789–1799, 2013. doi:10.1137/1.9781611973105.128.

[KR14] A. Khan and P. Raghavendra. On mimicking networks representing minimum terminal cuts. *Inf. Process. Lett.*, 114(7):365–371, 2014. doi:10.1016/j.ipl.2014.02.011.

[KR20] R. Krauthgamer and H. I. Rika. Refined vertex sparsifiers of planar graphs. *SIAM Journal on Discrete Mathematics*, 34(1):101–129, 2020. doi:10.1137/17M1151225.

[KRS01] G. Konjevod, R. Ravi, and F. S. Salman. On approximating planar metrics by tree metrics. *Inform. Process. Lett.*, 80(4):213–219, 2001. doi:10.1016/S0020-0190(01)00161-2.

[LLR95] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995. doi:10.1007/BF01200757.

[LM10] F. T. Leighton and A. Moitra. Extensions and limits to vertex sparsification. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 47–56, 2010. doi:10.1145/1806689.1806698.

[LMM15] J. R. Lee, M. Mendel, and M. Moharrami. A node-capacitated Okamura-Seymour theorem. *Math. Program.*, 153(2, Ser. A):381–415, 2015. doi:10.1007/s10107-014-0810-0.

[LR99] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999. doi:10.1145/331524.331526.

[LR10] J. R. Lee and P. Raghavendra. Coarse differentiation and multi-flows in planar graphs. *Discrete & Computational Geometry*, 43(2):346–362, 2010. doi:10.1007/s00454-009-9172-4.

[LS09] J. R. Lee and A. Sidiropoulos. On the geometry of graphs with a forbidden minor. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pages 245–254, 2009. doi:10.1145/1536414.1536450.

[LS13] J. R. Lee and A. Sidiropoulos. Pathwidth, trees, and random embeddings. *Combinatorica*, 33(3):349–374, 2013. doi:10.1007/s00493-013-2685-8.

[Mit99] J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM J. Comput.*, 28(4):1298–1309, 1999. doi:10.1137/S0097539796309764.

[MM16] K. Makarychev and Y. Makarychev. Metric extension operators, vertex sparsifiers and Lipschitz extendability. *Israel Journal of Mathematics*, 212(2):913–959, 2016. doi:10.1007/s11856-016-1315-8.

[MNS85] K. Matsumoto, T. Nishizeki, and N. Saito. An efficient algorithm for finding multicommodity flows in planar networks. *SIAM Journal on Computing*, 14(2):289–302, 1985. doi:10.1137/0214023.

[Moi09] A. Moitra. Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 3–12, 2009. doi:10.1109/FOCS.2009.28.

[Nao90] M. Naor. Succinct representation of general unlabeled graphs. *Discret. Appl. Math.*, 28(3):303–307, 1990. doi:10.1016/0166-218X(90)90011-Z.

[OS81] H. Okamura and P. Seymour. Multicommodity flows in planar graphs. *Journal of Combinatorial Theory, Series B*, 31(1):75 – 81, 1981. doi:10.1016/S0095-8956(81)80012-3.

[Pap77] C. H. Papadimitriou. The Euclidean traveling salesman problem is NP-complete. *Theor. Comput. Sci.*, 4(3):237–244, 1977. doi:10.1016/0304-3975(77)90012-3.

[PU89] D. Peleg and J. D. Ullman. An optimal synchronizer for the hypercube. *SIAM J. Comput.*, 18:740–747, August 1989. doi:10.1137/0218050.

[Räc08] H. Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *40th Annual ACM Symposium on Theory of Computing*, pages 255–264. ACM, 2008. doi:10.1145/1374376.1374415.

[Rao99] S. Rao. Small distortion and volume preserving embeddings for planar and euclidean metrics. In *Proceedings of the Fifteenth Annual Symposium on Computational Geometry*, SCG '99, pages 300–306. ACM, 1999. doi:10.1145/304893.304983.

[Sid10] A. Sidiropoulos. Optimal stochastic planarization. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 163–170, 2010. doi:10.1109/FOCS.2010.23.

[ST11] D. A. Spielman and S.-H. Teng. Spectral sparsification of graphs. *SIAM J. Comput.*, 40(4):981–1025, July 2011. doi:10.1137/08074489X.

[Tre00] L. Trevisan. When Hamming meets Euclid: The approximability of geometric TSP and Steiner tree. *SIAM J. Comput.*, 30(2):475–485, 2000. doi:10.1137/S0097539799352735.

[Tur84] G. Turán. On the succinct representation of graphs. *Discret. Appl. Math.*, 8(3):289–294, 1984. doi:10.1016/0166-218X(84)90126-4.

[TV02] P. Toth and D. Vigo, editors. *The Vehicle Routing Problem*, volume 9 of *SIAM monographs on discrete mathematics and applications*. SIAM, 2002. doi:10.1137/1.9780898718515.

[Whi32] H. Whitney. Non-separable and planar graphs. *Trans. Amer. Math. Soc.*, 34(2):339–362, 1932. doi:10.1090/S0002-9947-1932-1501641-2.