An Exponential Lower Bound for Cut Sparsifiers in Planar Graphs*

Nikolai Karpov†[1], Marcin Pilipczuk[2], and Anna Zych-Pawlewicz[3]

1 **St. Petersburg Department of V.A. Steklov Institute of Mathematics of the Russian Academy of Sciences, St. Petersburg, Russia and Institute of Informatics, University of Warsaw, Warsaw, Poland**
`kimaska@gmail.com.`

2 **Institute of Informatics, University of Warsaw, Warsaw, Poland**
`malcin@mimuw.edu.pl`

3 **Institute of Informatics, University of Warsaw, Warsaw, Poland**
`anka@mimuw.edu.pl`

Abstract

Given an edge-weighted graph G with a set Q of k terminals, a *mimicking network* is a graph with the same set of terminals that exactly preserves the sizes of minimum cuts between any partition of the terminals. A natural question in the area of graph compression is to provide as small mimicking networks as possible for input graph G being either an arbitrary graph or coming from a specific graph class.

In this note we show an exponential lower bound for cut mimicking networks in planar graphs: there are edge-weighted planar graphs with k terminals that require 2^{k-2} edges in any mimicking network. This nearly matches an upper bound of $\mathcal{O}(k2^{2k})$ of Krauthgamer and Rika [SODA 2013, arXiv:1702.05951] and is in sharp contrast with the $\mathcal{O}(k^2)$ upper bound under the assumption that all terminals lie on a single face [Goranci, Henzinger, Peng, arXiv:1702.01136]. As a side result we show a hard instance for the double-exponential upper bounds given by Hagerup, Katajainen, Nishimura, and Ragde [JCSS 1998], Khan and Raghavendra [IPL 2014], and Chambers and Eppstein [JGAA 2013].

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases mimicking networks, planar graphs

Digital Object Identifier 10.4230/LIPIcs.IPEC.2017.24

1 Introduction

One of the most popular paradigms when designing effective algorithms is preprocessing. These days in many applications, in particular mobile ones, even though fast running time is desired, the memory usage is the main limitation. The preprocessing needed for such applications is to reduce the size of the input data prior to some resource-demanding computations, without (significantly) changing the answer to the problem being solved. In this work we focus on this kind of preprocessing, known also as graph compression, for flows

* This research is a part of projects that have received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreements No 714704 (Marcin Pilipczuk and Anna Zych-Pawlewicz).

† Nikolai Karpov has been supported by the Warsaw Centre of Mathematics and Computer Science and the Government of the Russian Federation (grant 14.Z50.31.0030).

and cuts. The input graph needs to be compressed while preserving its essential flow and cut properties.

Central to our work is the concept of a *mimicking network*, introduced by Hagerup, Katajainen, Nishimura, and Ragde [6]. Let G be an edge-weighted graph with a set $Q \subseteq V(G)$ of k terminals. For a partition $Q = S \uplus \bar{S}$, a minimum cut between S and $\bar{S}$ is called a *minimum S-separating cut.* A *mimicking network* is an edge-weighted graph G' with $Q \subseteq V(G')$ such that the weights of minimum S-separating cuts are equal in G and G' for every partition $Q = S \uplus \bar{S}$. Hagerup et al [6] observed the following simple preprocessing step: if two vertices u and v are always on the same side of the minimum cut between S and $\bar{S}$ for every choice of the partition $Q = S \uplus \bar{S}$, then they can be merged without changing the size of any minimum S-separating cut. Such a procedure always leads to a mimicking network with at most 2^{2^k} vertices.

The above upper bound can be improved to a still double-exponential bound of roughly $2^{\binom{k-1}{\lfloor (k-1)/2 \rfloor}}$, as observed both by Khan and Raghavendra [7] and by Chambers and Eppstein [2]. In 2013, Krauthgamer and Rika [10] observed that the aforementioned preprocessing step can be adjusted to yield a mimicking network of size $\mathcal{O}(k^2 2^{2k})$ for planar graphs. Furthermore, they introduced a framework for proving lower bounds, and showed that there are (non-planar) graphs, for which any mimicking network has $2^{\Omega(k)}$ edges; a slightly stronger lower bound of $2^{(k-1)/2}$ has been shown by Khan and Raghavendra [7]. On the other hand, for planar graphs the lower bound of [10] is $\Omega(k^2)$. Furthermore, the planar graph lower bound applies even in the special case when all the terminals lie on the same face.

Very recently, two improvements upon these results for planar graphs have been announced. In a sequel paper, Krauthgamer and Rika [11] improve the polynomial factor in the upper bound for planar graphs to $\mathcal{O}(k 2^{2k})$ and show that the exponential dependency actually adheres only to the *number of faces containing terminals*: if the terminals lie on γ faces, one can obtain a mimicking network of size $\mathcal{O}(\gamma 2^{2\gamma} k^4)$. In a different work, Goranci, Henzinger, and Peng [5] showed a tight $\mathcal{O}(k^2)$ upper bound for mimicking networks for planar graph with all terminals on a single face.

Our results

We complement these results by showing an exponential lower bound for mimicking networks in planar graphs.

▶ **Theorem 1.1.** *For every integer $k \geq 3$, there exists a planar graph G with a set Q of k terminals and edge cost function under which every mimicking network for G has at least 2^{k-2} edges.*

This nearly matches the upper bound of $\mathcal{O}(k 2^{2k})$ of Krauthgamer and Rika [11] and is in sharp contrast with the polynomial bounds when the terminals lie on a constant number of faces [5, 11]. Note that it also nearly matches the improved bound of $\mathcal{O}(\gamma 2^{2\gamma} k^4)$ for terminals on γ faces [11], as k terminals lie on at most k faces.

As a side result, we also show a hard instance for mimicking networks in general graphs.

▶ **Theorem 1.2.** *For every integer $k \geq 1$, there exists a graph G with a set Q of $3k+1$ terminals and $2^{2^{\Omega(k)}}$ vertices such that no two vertices can be identified without strictly increasing the size of some minimum S-separating cut.*

The example of Theorem 1.2, obtained by essentially reiterating the construction of Krauthgamer and Rika [10], shows that the doubly exponential bound is natural for the preprocessing step of Hagerup et al [6], and one needs different techniques to improve upon it.

Related work

Apart from the aforementioned work on mimicking networks [5, 6, 7, 10, 11], there has been substantial work on preserving cuts and flows approximately, see e.g. [1, 4, 12]. If one wants to construct mimicking networks for vertex cuts in unweighted graphs with deletable terminals (or with small integral weights), the representative sets approach of Kratsch and Wahlström [8] provides a mimicking network with $\mathcal{O}(k^3)$ vertices, improving upon a previous quasipolynomial bound of Chuzhoy [3].

We prove Theorem 1.1 in Section 2 and show the example of Theorem 1.2 in Section 3.

2 Exponential lower bound for planar graphs

In this section we present the main result of the paper. We provide a construction that proves that there are planar graphs with k terminals whose mimicking networks are of size $\Omega(2^k)$.

In order to present the desired graph, for the sake of simplicity, we describe its dual graph (G, c). We let $Q = \{f_n, f_s, f_1, f_2, \ldots, f_{k-2}\}$ be the set of faces in G corresponding to terminals in the primal graph G^*.[1] There are two special terminal faces f_n and f_s, referred to as the north face and the south face. The remaining faces of Q are referred to as equator faces.

A set $S \subset Q$ is *important* if $f_n \in S$ and $f_s \notin S$. Note that there are 2^{k-2} important sets; in what follows we care only about minimum cuts in the primal graph for separations between important sets and their complements. For an important set S, we define its *signature* as a bit vector $\chi(S) \in [2]^{|Q|-2}$ whose i'th position is defined as $\chi(S)[i] = 1$ iff $f_i \in S$. Graph G will be composed of 2^{k-2} cycles referred to as important cycles, each corresponding to an important subset $S \subset Q$. A cycle corresponding to S is referred to as $\mathcal{C}_{\chi(S)}$ and it separates S from $\overline{S}$. Topologically, we draw the equator faces on a straight horizontal line that we call the equator. We put the north face f_n above the equator and the south face f_s below the equator. For any important $S \subset Q$, in the plane drawing of G the corresponding cycle $\mathcal{C}_{\chi(S)}$ is a curve that goes to the south of f_i if $f_i \in S$ and otherwise to the north of f_i. We formally define important cycles later on, see Definition 2.1.

We now describe in detail the construction of G. We start with a graph H that is almost a tree, and then embed H in the plane with a number of edge crossings, introducing a new vertex on every edge crossing. The graph H consists of a complete binary tree of height $k-2$ with root v and an extra vertex w that is adjacent to the root v and every one of the 2^{k-2} leaves of the tree. In what follows, the vertices of H are called *branching vertices*, contrary to *crossing vertices* that will be introduced at edge crossings in the plane embedding of H.

To describe the plane embedding of H, we need to introduce some notation of the vertices of H. The starting point of our construction is the edge $e = \{w, v\}$. Vertex v is the first branching vertex and also the root of H. In vertex v, edge e branches into $e_0 = \{v, v_0\}$ and $e_1 = \{v, v_1\}$. Now v_0 and v_1 are also branching vertices. The branching vertices are partitioned into layers $L_0, \ldots, L_{k-2}$. Vertex v is in layer $L_0 = \{v\}$, while v_0 and v_1 are in layer $L_1 = \{v_0, v_1\}$. Similarly, we partition edges into layers $\mathcal{E}_0^H, \ldots \mathcal{E}_{k-1}^H$. So far we have $\mathcal{E}_0^H = \{e\}$ and $\mathcal{E}_1^H = \{e_0, e_1\}$.

[1] Since the argument mostly operates on the dual graph, for notational simplicity, we use regular symbols for objects in the dual graph, e.g., G, c, f_i, while starred symbols refer to the dual of the dual graph, that is, the primal graph.

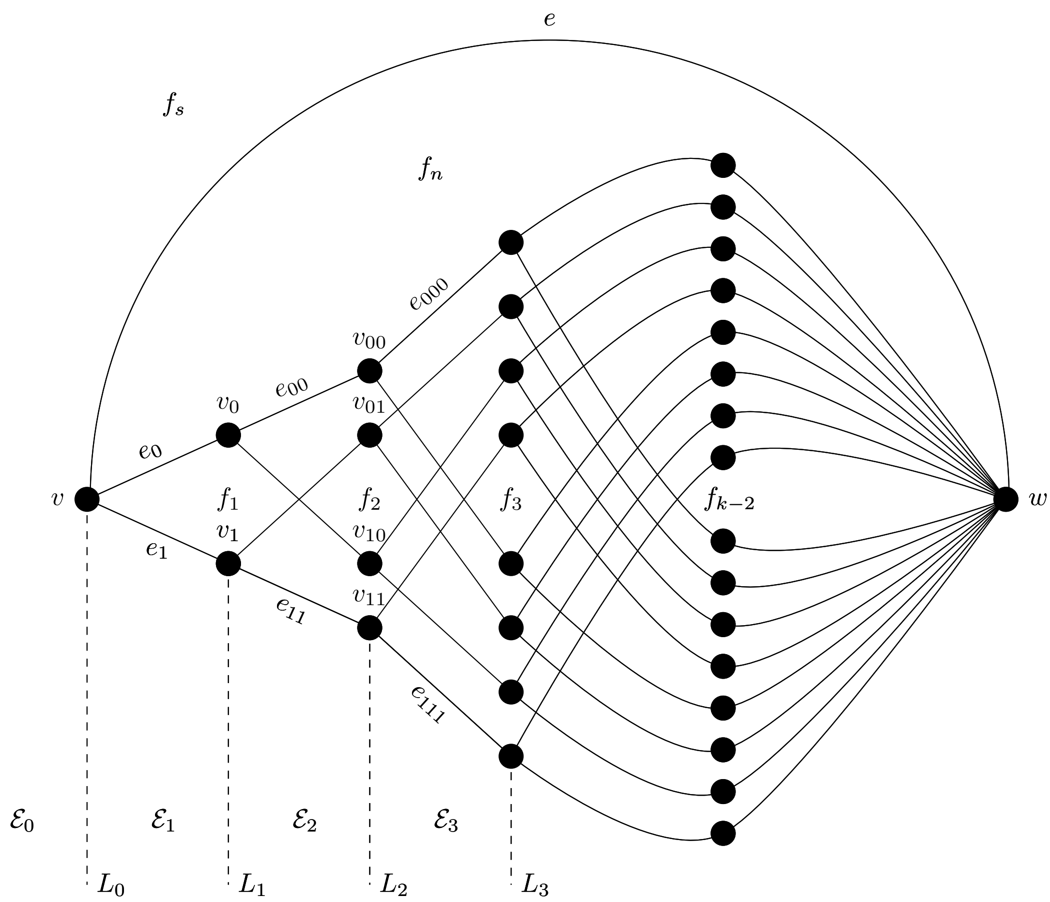

Figure 1 The graph G.

The construction continues as follows. For any layer $L_i, i \in \{1, \ldots, k-3\}$, all the branching vertices of $L_i = \{v_{00\ldots0} \ldots v_{11\ldots1}\}$ are of degree 3. In a vertex $v_a \in L_i$, $a \in [2]^i$, edge $e_a \in \mathcal{E}_i^H$ branches into edges $e_{0a} = \{v_a, v_{0a}\}, e_{1a} = \{v_a, v_{1a}\} \in \mathcal{E}_{i+1}^H$, where $v_{0a}, v_{1a} \in L_{i+1}$. We emphasize here that the new bit in the index is added *as the first symbol*. Every next layer is twice the size of the previous one, hence $|L_i| = |\mathcal{E}_i^H| = 2^i$. Finally the vertices of L_{k-2} are all of degree 2. Each of them is connected to a vertex in L_{k-3} via an edge in $\mathcal{E}_{k-2}^H$ and to the vertex w via an edge in $\mathcal{E}_{k-1}^H$.

We now describe the drawing of H, that we later make planar by adding crossing vertices, in order to obtain the graph G. As we mentioned before, we want to draw equator faces $f_1, \ldots f_{k-2}$ in that order from left to right on a horizontal line (referred to as an equator). Consider equator face f_i and vertex layer L_i for some $i > 0$. Imagine a vertical line through f_i perpendicular to the equator, and let us refer to it as an i'th meridian. We align the vertices of L_i along the i'th meridian, from the north to the south. We start with the vertex of L_i with the (lexicographically) lowest index, and continue drawing vertices of L_i more and more to the south while the indices increase. Moreover, the first half of L_i is drawn to the north of f_i, and the second half to the south of f_i. Every edge of H, except for e, is drawn as a straight line segment connecting its endpoints. The edge e is a curve encapsulating the north face f_n and separating it from f_s-the outer face of G.

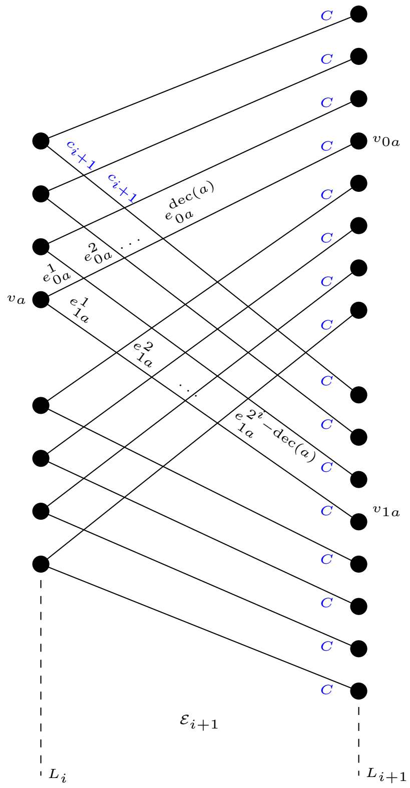

Figure 2 The layer $\mathcal{E}_{i+1}$. The vertex and edge names are black, their weights are blue.

The crossing vertices are added whenever the line segments cross. This way the edges of H are subdivided and the resulting graph is denoted by G. This completes the description of the structure and the planar drawing of G. We refer to Figure 1 for an illustration of the graph G. The set $\mathcal{E}_i$ consists of all edges of G that are parts of the (subdivided) edges of $\mathcal{E}_i^H$ from H, see Figure 2. We are also ready to define important cycles formally.

▶ **Definition 2.1.** Let $S \subset Q$ be important. Let π be a unique path in the binary tree $H - \{w\}$ from the root v to $v_{\overleftarrow{\chi(S)}}$, where $\overleftarrow{\cdot}$ operator reverses the bit vector. Let π' be the path in G corresponding to π. The important cycle $\mathcal{C}_{\chi(S)}$ is composed of e, π', and an edge in $\mathcal{E}_{k-1}$ adjacent to $v_{\overleftarrow{\chi(S)}}$.

We now move on to describing how weights are assigned to the edges of G. The costs of the edges in G admit $k-1$ values: $c_1, c_2, \ldots c_{k-2}$, and C. Let $c_{k-2} = 1$. For $i \in \{1 \ldots k-3\}$ let $c_i = \sum_{j=i+1}^{k-2} |\mathcal{E}_j| c_j$. Let $C = \sum_{j=1}^{k-2} |\mathcal{E}_i| c_i$. Let us consider an arbitrary edge $e_{ba} = \{v_a, v_{ba}\}$ for some $a \in [2]^i, i \in \{0 \ldots k-3\}, b \in \{0,1\}$ (see Figure 2 for an illustration). As we mentioned before, e_{ba} is subdivided by crossing vertices into a number of edges. If $b = 0$, then edge e_{ba} is subdivided by[2] $\mathrm{dec}(a)$ crossing vertices into $\mathrm{dec}(a)+1$ edges: $e_{ba}^1 = \{v_a, x_{ba}^1\}, e_{ba}^2 = \{x_{ba}^1, x_{ba}^2\} \ldots e_{ba}^{\mathrm{dec}(a)+1} = \{x_{ba}^{\mathrm{dec}(a)}, v_{ba}\}$. Among those edges $e_{ba}^{\mathrm{dec}(a)+1}$ is assigned cost C, and the remaining edges subdividing e_{ba} are assigned cost c_i. Analogically, if $b = 1$, then edge e_{ba} is subdivided by $2^i - 1 - \mathrm{dec}(a)$ crossing vertices into $2^i - \mathrm{dec}(a)$ edges: $e_{ba}^1 = \{v_a, x_{ba}^1\}, e_{ba}^2 = \{x_{ba}^1, x_{ba}^2\} \ldots e_{ba}^{2^i-\mathrm{dec}(a)} = \{x_{ba}^{2^i-1-\mathrm{dec}(a)}, v_{ba}\}$. Again, we let

[2] For a bit vector a, $\mathrm{dec}(a)$ denotes the integral value of a read as a number in binary.

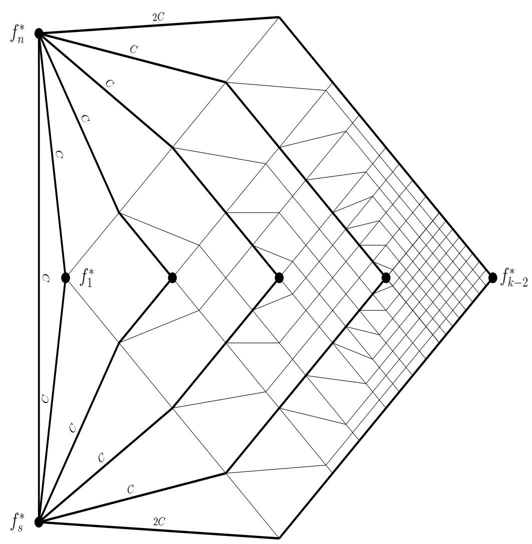

Figure 3 Primal graph G^*.

edge $e_{ba}^{2^i-\mathrm{dec}(a)}$ have cost C, and the remaining edges subdividing e_{ba} are assigned cost c_i. Finally, all the edges connecting the vertices of the last layer with w have weight $c_{k-2} = 1$. The cost assignment within an edge layer is presented in Figure 2.

This finishes the description of the dual graph G. We now consider the primal graph G^* with the set of terminals Q^* consisting of the k vertices of G^* corresponding to the faces Q of G. In the remainder of this section we show that there is a cost function on the edges of G^*, under which any mimicking network for G^* contains at least 2^{k-2} edges. This cost function is in fact a small perturbation of the edge costs implied by the dual graph G.

In order to accomplish this, we use the framework introduced in [10]. In what follows, $\mathrm{mincut}_{G,c}(S, S')$ stands for the minimum cut separating S from S' in a graph G with cost function c. Below we provide the definition of the cutset-edge incidence matrix and the Main Technical Lemma from [10].

▶ **Definition 2.2** (Incidence matrix between cutsets and edges)**.** Let (G, c) be a k-terminal network, and fix an enumeration $S_1, \ldots S_m$ of all $2^{k-1} - 1$ distinct and nontrivial bipartitions $Q = S_i \cup \overline{S}_i$. The cutset-edge incidence matrix of (G, c) is the matrix $A_{G,c} \in \{0, 1\}^{m \times E(G)}$ given by

$$(A_{G,c})_{i,e} = \begin{cases} 1 \text{ if } e \in \mathrm{mincut}_{G,c}(S_i, \overline{S}_i) \\ 0 \text{ otherwise.} \end{cases}$$

▶ **Lemma 2.3** (Main Technical Lemma of [10])**.** *Let (G, c) be a k-terminal network. Let $A_{G,c}$ be its cutset-edge incidence matrix, and assume that for all $S \subset Q$ the minimum S-separating cut of G is unique. Then there is for G an edge cost function $\tilde{c} : E(G) \mapsto \mathbb{R}^+$, under which every mimicking network (G', c') satisfies $|E(G')| \geq \mathrm{rank}(A_{G,c})$.*

Recall that G^* is the dual graph to the graph G that we constructed. By slightly abusing the notation, we will use the cost function c defined on the dual edges also on the corresponding primal edges. Let $Q^* = \{{f_n}^*, {f_s}^*, {f_1}^*, \ldots {f_{k-2}}^*\}$ be the set of terminals in G^* corresponding to $f_n, f_s, f_1, \ldots f_{k-2}$ respectively. We want to apply Lemma 2.3 to G^* and Q^*. For that we need to show that the cuts in G^* corresponding to important sets are unique and that $\mathrm{rank}(A_{G^*,c})$ is high.

As an intermediate step let us argue that the following holds.

▶ **Claim 1.** *There are k edge disjoint simple paths in G^* from ${f_n}^*$ to ${f_s}^*$: $\pi_0, \pi_1, \ldots, \pi_{k-2}, \pi_{k-1}$. Each π_i is composed entirely of edges dual to the edges of $\mathcal{E}_i$ whose cost equals C. For $i \in \{1 \ldots k-2\}$, π_i contains vertex ${f_i}^*$. Let π_i^n be the prefix of π_i from ${f_n}^*$ to ${f_i}^*$ and π_i^s be the suffix from ${f_i}^*$ to ${f_s}^*$. The number of edges on π_i is 2^i, and the number of edges on π_i^n and π_i^s is 2^{i-1}.*

Proof. The primal graph G^* together with paths $\pi_0, \pi_1 \ldots \pi_{k-2}, \pi_{k-1}$ is pictured in Figure 3. The paths π_{k-2}, π_{k-1} visit the same vertices in the same manner, so for the sake of clarity only one of these paths is shown in the picture. This proof contains a detailed description of these paths and how they emerge from in the dual graph G.

Consider a layer L_i. Recall that for any $ba \in [2]^i$ edge e_{ba} of the almost tree is subdivided in G, and all the resulting edges are in $\mathcal{E}_i$. If $b = 0$, then edge e_{ba} is subdivided by $\mathrm{dec}(a)$ crossing vertices into $\mathrm{dec}(a)+1$ edges: $e_{ba}^1 = \{v_a, x_{ba}^1\}, e_{ba}^2 = \{x_{ba}^1, x_{ba}^2\} \ldots e_{ba}^{\mathrm{dec}(a)+1} = \{x_{ba}^{\mathrm{dec}(a)}, v_{ba}\}$, where $c(e_{ba}^{\mathrm{dec}(a)+1}) = C$. Analogically, if $b = 1$, then edge e_{ba} is subdivided by $2^i - 1 - \mathrm{dec}(a)$ crossing vertices into $2^i - \mathrm{dec}(a)$ edges: $e_{ba}^1 = \{v_a, x_{ba}^1\}, e_{ba}^2 = \{x_{ba}^1, x_{ba}^2\} \ldots e_{ba}^{2^i-\mathrm{dec}(a)} = \{x_{ba}^{2^i-1-\mathrm{dec}(a)}, v_{ba}\}$. Again, $c(e_{ba}^{2^i-\mathrm{dec}(a)}) = C$. Consider the edges of $\mathcal{E}_i$ incident to vertices in L_i. If we order these edges lexicographically by their lower index, then each consecutive pair of edges shares a common face. Moreover, the first edge $e_{00\ldots0}^1$ is incident to f_n and the last edge $e_{11\ldots1}^1$ is incident to f_s. This gives a path π_i from f_n to f_s through f_i in the primal graph where all the edges on π_i have cost C. Path π_{k-1} is given by the edges of $\mathcal{E}_{k-1}$ in a similar fashion and path π_0 is composed of a single edge dual to e. ◀

We move on to proving that the condition in Lemma 2.3 holds. We extend the notion of important sets $S \subseteq Q$ to sets $S^* \subseteq Q^*$ in the natural manner.

▶ **Lemma 2.4.** *For every important $S^* \subset Q^*$, the minimum cut separating S^* from $\overline{S^*}$ is unique and corresponds to cycle $\mathcal{C}_{\chi(S)}$ in G.*

Proof. Let $\mathcal{C}$ be the set of edges of G corresponding to some minimum cut between S^* and $\overline{S}^*$ in G^*. Let $S \subseteq Q$ be the set of faces of G corresponding to the set S^*. We start by observing that the edges of G^* corresponding to $\mathcal{C}_{\chi(S)}$ form a cut between S^* and $\overline{S}^*$. Consequently, the total weight of edges of $\mathcal{C}$ is at most the total weight of the edges of $\mathcal{C}_{\chi(S)}$.

By Claim 1, $\mathcal{C}$ contains at least k edges of cost C, at least one edge of cost C per edge layer (it needs to hit an edge in every path $\pi_0, \ldots \pi_{k-1}$). Note that $\mathcal{C}_{\chi(S)}$ contains exactly k edges of cost C. We assign the weights in a way that C is larger than all other edges in the graph taken together. This implies that $\mathcal{C}$ contains exactly one edge of cost C in every edge layer $\mathcal{E}_i$. In particular, $\mathcal{C}$ contains the edge $e = \{v, w\}$.

Furthermore, the fact that ${f_i}^*$ lies on π_i implies that the edge of weight C in $\mathcal{E}_i \cap \mathcal{C}$ lies on π_i^n if ${f_i}^* \notin S$ and lies on π_i^s otherwise. Consequently, in $G^* - \mathcal{C}$ there is one connected component containing all vertices of S^* and one connected component containing all vertices of $\overline{S^*}$. By the minimality of $\mathcal{C}$, we infer that $G^* - \mathcal{C}$ contains no other connected components apart from the aforementioned two components. By planarity, since any minimum cut in a planar graph corresponds to a collection of cycles in its dual, this implies that $\mathcal{C}$ is a single cycle in G.

Let e_i be the unique edge of $\mathcal{E}_i \cap \mathcal{C}$ of weight C and let e_i' be the unique edge of $\mathcal{E}_i \cap \mathcal{C}_{\chi(S)}$ of weight C. We inductively prove that $e_i = e_i'$ and that the subpath of $\mathcal{C}$ between e_i and e_{i+1} is the same as on $\mathcal{C}_{\chi(S)}$. For the base of the induction, note that $e_0 = e_0' = e$.

Consider an index $i > 0$ and the face f_i. If $f_i \in S$, i.e., f_i belongs to the north side, then e_i lies south of f_i, that is, lies on π_i^s. Otherwise, if $f_i \notin S$, then e_i lies north of f_i, that is, lies on π_i^n.

Let v_a and v_{ba} be the vertices of $\mathcal{C}_{\chi(S)}$ that lie on L_{i-1} and L_i, respectively. By the inductive assumption, v_a is an endpoint of $e'_{i-1} = e_{i-1}$ that lies on $\mathcal{C}$. Let $e_i = xv_{bc}$, where $v_{bc} \in L_i$ and let $e'_i = x'v_{ba}$. Since $\mathcal{C}$ is a cycle in G that contains exactly one edge on each path π_i, we infer that $\mathcal{C}$ contains a path between v_a and v_{bc} that consists of e_i and a number of edges of $\mathcal{E}_i$ of weight c_i. A direct check shows that the subpath from v_a to v_{ba} on $\mathcal{C}_{\chi(S)}$ is the unique such path with minimum number of edges of weight c_i. Since the weight c_i is larger than the total weight of all edges of smaller weight, from the minimality of $\mathcal{C}$ we infer that $v_{ba} = v_{bc}$ and $\mathcal{C}$ and $\mathcal{C}_{\chi(S)}$ coincide on the path from v_a to b_{ba}.

Consequently, $\mathcal{C}$ and $\mathcal{C}_{\chi(S)}$ coincide on the path from the edge $e = vw$ to the vertex $v_{\overleftarrow{\chi(S)}} \in L_{k-2}$. From the minimality of $\mathcal{C}$ we infer that also the edge $\{w, v_{\overleftarrow{\chi(S)}}\}$ lies on the cycle $\mathcal{C}$ and, hence, $\mathcal{C} = \mathcal{C}_{\chi(S)}$. This completes the proof. ◀

▶ **Claim 2.** $\mathrm{rank}(A_{G,c}) \geq 2^{k-2}$.

Proof. Recall Definition 2.1 and the fact that $\mathcal{C}_{\chi(S)}$ is defined for every important $S \subseteq Q$. This means that the only edge in $\mathcal{E}_{k-1}$ that belongs to $\mathcal{C}_{\chi(S)}$ is the edge adjacent to $v_{\overleftarrow{\chi(S)}}$. Let us consider the part of adjacency matrix where rows correspond to the cuts corresponding to $\mathcal{C}_{\chi(S)}$ for important $S \subset Q$ and where columns correspond to the edges in $\mathcal{E}_{k-1}$ of weight C. Let us order the cuts according to $\overleftarrow{\chi(S)}$ and the edges by the index of the adjacent vertex in L_{k-2} (lexicographically). Then this part of $A_{G,c}$ is an identity matrix. Hence, $\mathrm{rank}(A_{G,c}) \geq 2^{k-2}$. ◀

Lemma 2.4 and Claim 2 provide the conditions necessary for Lemma 2.3 to apply. This proves our main result stated in Theorem 1.1.

3 Doubly exponential example

In this section we show an example graph for which the compression technique introduced by Hagerup et al [6] does indeed produce a mimicking network on $2^{2^{\Omega(k)}}$ vertices. Our example relies on doubly exponential edge costs. Note that an example with single exponential costs can be compressed into a mimicking network of size single exponential in k using the techniques of [8].

Before we go on, let us recall the technique of Hagerup et al [6]. Let G be a weighted graph and Q be the set of terminals. Observe that a minimum cut separating $S \subset Q$ from $\overline{S} = Q \setminus S$, when removed from G, divides the vertices of G into two sides: the side of S and the side of $\overline{S}$. The side is defined for each vertex, as all connected components obtained by removing the minimum cut contain a terminal. Now if two vertices u and v are on the same side of the minimum cut between S and $\overline{S}$ for every $S \subset Q$, then they can be merged without changing the size of any minimum S-separating cut. As a result there is at most 2^{2^k} vertices in the graph. After this brief introduction we move on to describing our example.

Our construction builds up on the example provided in [10] in the proof of Theorem 1.2. Without loss of generality, assume that k is divisible by 3 and that $l := \binom{k}{\frac{2}{3}k}$ is even. Their graph is a complete bipartite graph $G = (Q, U, E)$, where one side of the graph consists of the k terminals $Q = \{q_1, \ldots, q_k\}$, and the other side of the graph consists of $l = \binom{k}{\frac{2}{3}k}$ non-terminals $U = \{u_{S_1}, \ldots, u_{S_l}\}$, with $S_1, \ldots, S_l$ denoting the different subsets of terminals of size $2/3k$. The costs of the edges of G are as follows: every non-terminal u_{S_i} is connected by edges of cost 1 to every terminal in S_i , and by edges of cost $2 + \epsilon$ to every terminal in $\overline{S_i} = Q \setminus S_i$, for $\epsilon = 1/k$. We modify the cost function defined in this example by multiplying each edge cost by a constant α that we define later. We also need to be more careful with

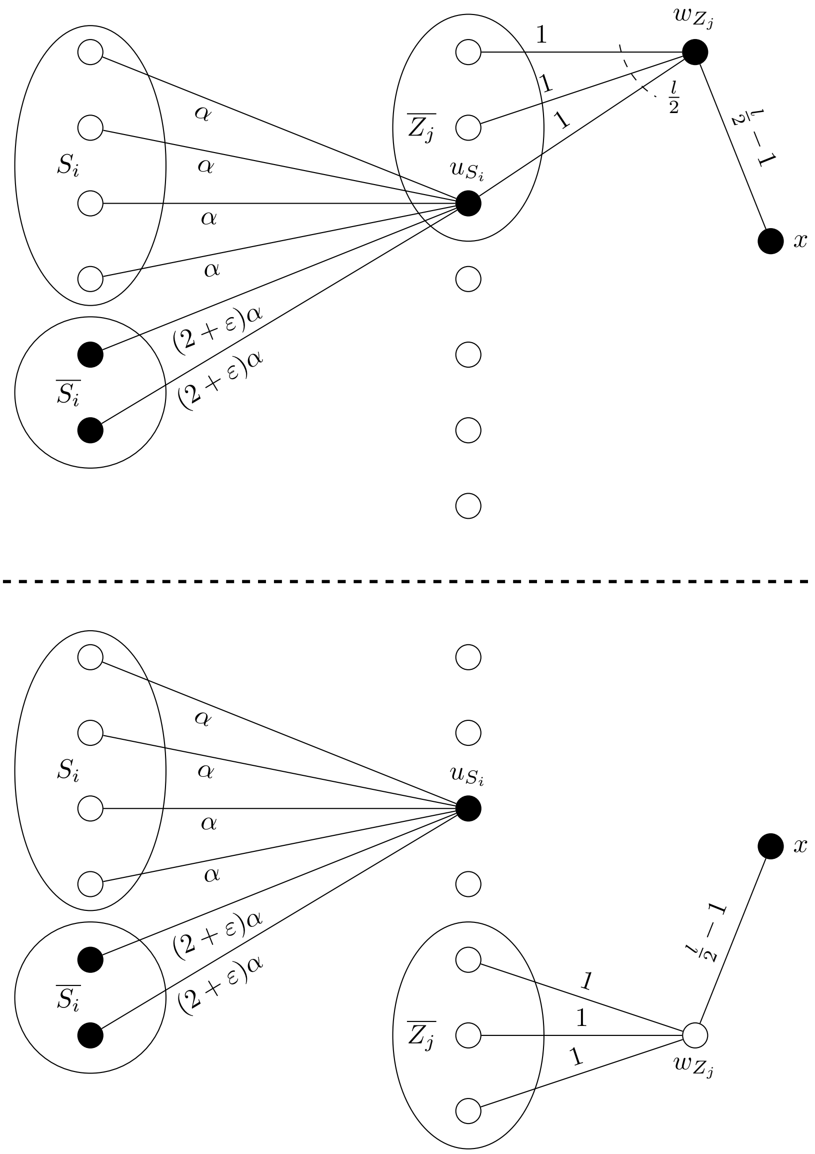

Figure 4 Illustration of the construction. The two panels correspond to two cases in the proof, either $u_{S_i} \in Z_j$ (top panel) or $u_{S_i} \notin Z_j$ (bottom panel).

ϵ. We set $\epsilon = \frac{3}{k} + \frac{6}{k^2}$. In addition to that we build a third layer of $m = \binom{l}{l/2}$ vertices $W = \{w_{Z_1}, \dots, w_{Z_m}\}$, where $Z_1, \dots, Z_m$ denote different subsets of U of size $l/2$. There is a complete bipartite graph between U and W. An edge $\{u_{S_i}, w_{Z_j}\}$ has cost 0 if $u_{S_i} \in Z_j$ and has cost 1 otherwise. We add one more vertex to the graph which we refer to as x and connect it with edges of cost $l/2 - 1$ to each vertex in W. Let us refer to the resulting graph as G'. We let $Q' = Q \cup \{x\}$ be the corresponding terminal set; a set $S \subseteq Q'$ is *important* if $x \notin S$ and $|S| = \frac{2}{3}k$.

▶ **Lemma 3.1.** *Let $S_i' \subset Q$ be important. For $\alpha = 2^{2^k} \cdot 2^k$, the vertex w_{Z_j} is on the S_i-side of the minimum cut between S_i and $\overline{S_i}$ if and only if $u_{S_i} \in Z_i$.*

Proof. In [9] it is proven that the unique minimum cut separating S_i from $\overline{S_i}$ in G, $|S_i| = 2/3k$, partitions vertices into $\overline{S_i}$ side: $Y = \{u_{S_i}\} \cup \overline{S_i}$ and S_i side: $V(G) \setminus Y = \{u_{S_j} : j \neq i\} \cup S_i$. We refer to [9] for the proof details, but the reason why this holds is the following. The terminals are connected only via vertices of U. Every vertex u_{S_j} can either cut the edges $E(u_{S_j}, S_i)$ connecting u_{S_j} with S_i (choosing $\overline{S_i}$ side) or cut the edges $E(u_{S_j}, \overline{S_i})$ connecting u_{S_j} with $\overline{S_i}$ (choosing S_i side). For $j = i$ it holds that $c(E(U_{S_j}, S_i)) = 2/3k$ while $c(E(u_{S_j}, \overline{S_i})) = (2+\epsilon)k/3$, so it is better for u_{S_i} to join $\overline{S_i}$ side. Moreover, the difference between these two values is greater than 1 for $\epsilon = \frac{3}{k} + \frac{6}{k^2}$. For $j \neq i$ it holds that $c(E(u_{S_j}, S_i)) \geq 2/3k + (1+\epsilon)$ while $c(E(u_{S_j}, \overline{S_i})) \leq k/3(2+\epsilon) - (1+\epsilon)$. It is easy to verify that for $\epsilon = \frac{3}{k} + \frac{6}{k^2}$ it is better for u_{S_j} to join S_i side and that the difference between the two alternative cut values is greater than 1. The bottom line is that u_{S_i} picks $\overline{S_i}$ side, whereas all other u_{S_j} vertices pick S_i side. If a vertex switches sides, the value of the minimum cut increases by more than 1.

In our example we multiply all the edge weights in this example by α, so the increase in the cut value is more than α. Let us now consider graph G' with terminal set Q'. Consider the cut between S_i and $\overline{S_i} = Q' \setminus S_i$ (so $\overline{S_i}$ contains x). Graph G' contains G as a subgraph, so to disconnect S_i from $Q \setminus S_i$, each vertex u_{S_j} again has to cut either $E(u_{S_j}, S_i)$ or $E(u_{S_j}, Q \setminus S_i)$. Set $\alpha = 2^{2^k} \cdot 2^k$. Consider the minimum cut in G. The minimum cut in G' restricted to G uses the same edges. It does not pay of to flip sides for any vertex in U, as we can never make up for the difference α with no more than $|U| \cdot |W|$ edges of cost 1. Now fix a vertex $w_{Z_j} \in W$. We consider two cases: $u_{S_i} \in Z_j$ and $u_{S_i} \notin Z_j$; see also Figure 4.

Case 1: $u_{S_i} \in Z_j$**.**

As argued above, all vertices of U choose their side according to what is best in G, so u_{S_i} is the only vertex in U on the $\overline{S_i}$ side. To join the S_i side, w_{Z_j} has to cut edges $\{x, w_{Z_j}\}$ and $\{u_{S_i}, w_{Z_j}\}$ of total cost $l/2 - 1$. To join the $\overline{S_i}$ side, w_{Z_j} needs to cut $l/2$ edges of cost 1 to vertices $u_{S_{i'}}$ for $u_{S_{i'}} \notin Z_j, i' \neq i$. Thus, it is less costly if w_{Z_j} joints the S_i side.

Case 2: $u_{S_i} \notin Z_j$**.**

Again all vertices of U choose their side according to what is best in G, so u_{S_i} is the only vertex in U on the $\overline{S_i}$ side. To join the S_i side, w_{Z_j} has to cut edges $\{x, w_{Z_j}\}$ and $\{u_{S_i}, w_{Z_j}\}$ of total cost $l/2$. To join the $\overline{S_i}$ side, w_{Z_j} needs to cut $l/2 - 1$ edges of cost 1 to vertices $u_{S_{i'}}$ for $u_{S_{i'}} \notin Z_j, i' \neq i$. Thus, it is less costly for w_{Z_j} to join the $\overline{S_i}$ side. ◀

Lemma 3.1 shows that G' cannot be compressed using the technique presented in [6]. To see that let us fix two vertices w_{Z_j} and $w_{Z_{j'}}$, and let $S_i \in Z_j \setminus Z_{j'}$. Then, Lemma 3.1 shows that w_{Z_j} and $w_{Z_{j'}}$ lie on different sides of the minimum cut between S_i and $\overline{S_i}$. Thus, w_{Z_j} and $w_{Z_{j'}}$ cannot be merged. Similar but simpler arguments show that no other pair of vertices in G' can be merged, finishing the proof of Theorem 1.2.

References

1. Alexandr Andoni, Anupam Gupta, and Robert Krauthgamer. Towards $(1+\epsilon)$-approximate flow sparsifiers. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 279–293. SIAM, 2014. doi:10.1137/1.9781611973402.20.
2. Erin W. Chambers and David Eppstein. Flows in one-crossing-minor-free graphs. *J. Graph Algorithms Appl.*, 17(3):201–220, 2013. doi:10.7155/jgaa.00291.
3. Julia Chuzhoy. On vertex sparsifiers with steiner nodes. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference,*

STOC 2012, New York, NY, USA, May 19 - 22, 2012, pages 673–688. ACM, 2012. doi:10.1145/2213977.2214039.

4 Matthias Englert, Anupam Gupta, Robert Krauthgamer, Harald Räcke, Inbal Talgam-Cohen, and Kunal Talwar. Vertex sparsifiers: New results from old techniques. *SIAM J. Comput.*, 43(4):1239–1262, 2014. doi:10.1137/130908440.

5 Gramoz Goranci, Monika Henzinger, and Pan Peng. Improved guarantees for vertex sparsification in planar graphs. *CoRR*, abs/1702.01136, 2017. arXiv:1702.01136.

6 Torben Hagerup, Jyrki Katajainen, Naomi Nishimura, and Prabhakar 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.

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

8 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 450–459. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.46.

9 Robert Krauthgamer and Inbal Rika. Mimicking networks and succinct representations of terminal cuts. *CoRR*, abs/1207.6246, 2012. arXiv:1207.6246.

10 Robert Krauthgamer and Inbal Rika. Mimicking networks and succinct representations of terminal cuts. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1789–1799. SIAM, 2013. doi:10.1137/1.9781611973105.128.

11 Robert Krauthgamer and Inbal Rika. Refined vertex sparsifiers of planar graphs. *CoRR*, abs/1702.05951, 2017. arXiv:1702.05951.

12 Konstantin Makarychev and Yury Makarychev. Metric extension operators, vertex sparsifiers and lipschitz extendability. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 255–264. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.31.