מכון ויצמן למדע

WEIZMANN INSTITUTE OF SCIENCE

Thesis for the degree Master of Science

Mimicking Networks and Succinct Representations of Terminal Cuts

Inbal Rika

under the supervision of

Prof. Robert Krauthgamer

January 11, 2013

The only real valuable thing
is intuition.

Albert Einstein

Acknowledgements

I would like to express my deep appreciation and my sincere gratitude to my Masters thesis advisor, Prof. Robert Krauthgamer, for his kindly and patient guidance, friendly support and for his devoted attention. It was a pleasure to work and to be inspired by him.

I would like to thank my fellow students, the faculty members and the administrative staff of the Department of Computer Science and Applied Mathematics for the pleasant atmosphere and the convenient workspace.

Abstract

Given a large edge-weighted network G with k vertices designated as terminals, we wish to compress it and store, using little memory, the value of the minimum cut (or equivalently, maximum flow) between every bipartition of terminals. One appealing methodology to implement a compression of G is to construct a *mimicking network*: a small network G' with the same k terminals, in which the minimum cut value between every bipartition of terminals is the same as in G. This notion was introduced by Hagerup, Katajainen, Nishimura, and Ragde [JCSS '98], who proved that such G' of size at most 2^{2^k} always exists. Obviously, by having access to the smaller network G', certain computations involving cuts can be carried out much more efficiently.

We provide several new bounds, which narrow the previously gap between known upper and lower bounds from doubly-exponential to only singly-exponential, both for planar and for general graphs. Our first and main result is that every k-terminal planar network admits a mimicking network G' of size $O(k^2 2^{2k})$, which is moreover a minor of G. On the other hand, some planar networks G require $|E(G')| \geq \Omega(k^2)$. For general networks, we show that certain bipartite graphs only admit mimicking networks of size $|V(G')| \geq 2^{\Omega(k)}$, and moreover, every data structure that stores the minimum cut value between all bipartitions of the terminals must use $2^{\Omega(k)}$ machine words, which proves that the trivial upper bound is tight.

Contents

1 Introduction

These days, more than ever, we deal with huge graphs such as social networks, communication networks, roadmaps and so forth. But even when our main interest is only in a small portion of the input graph G, we still need to process all or most of it in order to answer our query, since the runtime and memory requirements of many common graph algorithms depend on the input (graph) size. Therefore, a natural question is whether we can represent the graph G in a succinct structure that contains only the relevant information about the original graph.

Turán [Tur84] had previously studied the problem of succinctly representing an unlabeled planar graph. The representation obtained was optimal in size but could not be used to speed-up algorithms. The problem of finding a succinct representation of unlabeled general graphs, was solved a few years later by Naor [Nao90].

Later on, Feder and Motwani [FM95] were the first to introduce the basic concept of *graph compression*. They require that the compressed graph has fewer edges than the original graph, and that each graph can be quickly computed from the other one. Their results are stronger than previous ones in two important aspects. First, their results are derived for the case of labeled graphs. Second, since their results are not only concerned with representing a graph using the fewest possible number of bits, but also with finding representations that do not obscure the structure present in the graph so as to enable an efficient implementation of a large class of algorithms. Feder and Motwani have demonstrated how their paradigm leads to significantly improved running time of the best algorithms known for graph problems such as matchings, vertex connectivity, edge connectivity and all-pairs shortest paths.

We ask whether for every labeled graph G we can find a smaller graph (in terms of vertices or edges) G' that exactly (or approximately) preserves some properties of the original graph G such as distances, cuts or connectivity. Notice that in a sense, G' is a compressed graph of G. Indeed, we do not require G' to be computed very quickly from G (and we do not require to be able to restore the original graph G from G'), but typical graph algorithms run faster on the smaller graph G' than on the bigger original graph G. If G has some special structure which associates it to a family of graphs, such as planar graphs, trees, bounded treewidth etc., then we could ask that G' preserves that property of G, and has also a similar structure as G such that they both belong to the same graph family. For example, Krauthgamer and Zondiner [KZ12] study planar graphs, and seek a smaller graph that is a minor of the original graph (and thus also a planar) and preserves the distances between a specific set of vertices.

Another significant advantage of the compressed graph G' is that it requires far less memory than storing the original graph G, which could be critical for machines with limited resources such as smartphones, assuming that the preprocessing can be executed in advance on much more powerful machines. This paradigm becomes indispensable when computations on the compressed graph are to be performed repeatedly (after a one-time preprocessing).

We focus on cuts and flows, which are of fundamental importance in computer science, engineering, and operations research, because of their frequent usage in many application areas. More specifical, we consider the cuts and flows between k "important" vertices denoted by Q called terminals. One of the central (and classical) problems in network flows is the characterization of the (single commodity) flow behavior of networks with $k > 2$ terminals, first motivated and solved by Gomory and Hu [GH61] and later improved and simplified by many others. The Gomory-Hu approach, as well as its subsequent improvements and simplifications, deals only with the case where all the vertices in the graph are the terminals, i.e. $V = Q$. They computing maximum flow between all pairs of vertices (terminals) in a network. However, there may be cases where the terminals is a small subset of the vertices in the network, i.e. $Q \subset V$ and $|Q| << |V|$. Under this perspective, there is a recent, renewed interest in the problem of characterizing the flow behavior of networks with a small number of terminals [HKNR98, Moi09].

Specifically, we study the compression of a large graph G containing k terminals, into a smaller graph G' containing the same terminals, while maintaining the following condition: the minimum cut between every bipartition of the terminals has exactly the same value in G and in G'. The above cut condition can be also stated in terms of maximum flow, because it effectively deals with the single-source single-sink case, for which we have the max-flow min-cut theorem of Ford and Fulkerson [FF]. We now turn to define this problem more formally, restricting our attention (throughout) to undirected graphs.

A *network* (G, c) is a graph G with an edge-costs function $c : E(G) \to \mathbb{R}^+$. The *size* of a network is the number of vertices of G. The network is called a *k-terminal network* if the graph G has k distinguished vertices called *terminals*, denoted $Q = \{q_1, \dots, q_k\} \subseteq V(G)$. In such a network, a cut $(W, V(G) \setminus W)$ is said to be *S-separating* if it separates the terminals subset $S \subset Q$ from the remaining terminals $\bar{S} := Q \setminus S$, i.e. if $W \cap Q$ is either S or $\bar{S}$. When clear from the context, $(W, V(G) \setminus W)$ may refer not only to a bipartition of the vertices, but also to its corresponding *cutset* (set of edges crossing the cut). The *cost* of a cut $(W, V(G) \setminus W)$ is the sum of costs of all the edges in the cutset. We let $\text{mincut}_{G,c}(S, \bar{S})$ and $\text{min-cutset}_{G,c}(S, \bar{S})$ denote the cost and the cutset (correspondingly) of an S-separating cut in the network (G, c) of minimum cost (breaking ties arbitrarily). We omit the subscript c when clear from the context.

Definition 1.1 (Mimicking Network [HKNR98])**.** *Let (G, c) be a k-terminal network. A* mimicking network *of (G, c) is a k-terminal network (G', c') with the same set of terminals Q, such that for all $S \subset Q$,*[1]

$$\text{mincut}_{G',c'}(S, \bar{S}) = \text{mincut}_{G,c}(S, \bar{S}).$$

The above definition (albeit for directed networks) was introduced by Hagerup, Katajainen, Nishimura, and Ragde [HKNR98], who proved that following theorem.

Theorem 1.2 ([HKNR98])**.** *Every k-terminal network (G, c) admits a mimicking network of size at most 2^{2^k}.*

Proof (sketch). Let (G, c) be a k-terminal network, and let $(S_i, \bar{S}_i)$ be the $2^{k-1} - 1$ bipartitions of the terminals Q. For every bipartition i, fix a minimum S_i-separating cut. Define an equivalence relation on the vertices of G by declaring that two vertices are equivalent (in the same equivalence class) if and only if for all i, they are in the same side of the (fixed) minimum S_i-separating cut. The mimicking network G' has a vertex for

[1]Throughout, we omit the trivial exclusion $S \neq \emptyset, Q$.

each equivalence class in the relation, and the terminals are the vertices associated to the equivalence classes which contain terminals. Think of G' as a complete graph, where the cost of every edge between two vertices (equivalence classes) is the sum of the costs of all edges that one of their endpoints is in one equivalence class and the second endpoint is in the other; if there are no such edges in G then the cost of the edge in G' is 0. It is easy to be convinced that G' preserves all the minimum terminal cuts, since all the edges that their two endpoints are in the same equivalence class never participate in any terminal min-cut.

We now turn to bound the number of equivalence classes. To simplify the counting, we associate a binary vector with $2^{k-1}-1$ entries to each vertex in G, where the i-th entry will be 1 if the vertex is in the same side of S_i in the minimum S_i-separating cut and 0 otherwise. In this setting two vertices are in the same class if the vectors associated to them are the same. Thus, the number of different classes is the number of different vectors, which is bounded by $2^{2^{(k-1)}-1}$.

□

Notice that we prove a slightly better upper bound of $2^{2^{(k-1)}-1}$. The improvement is achieved since we compute a mimicking network for undirected graphs, and so the number of different bipartitions of terminals is bounded by $2^{(k-1)}-1$, while [HKNR98] compute a mimicking network for directed graphs, and so the number of different bipartitions (ordered pairs) is bounded by 2^k.

Subsequently, Chaudhuri, Subrahmanyam, Wagner, and Zaroliagis [CSWZ00] studied specific graph families, showing an improved upper bound of $O(k)$ for graphs G that have bounded treewidth. For the special case of outerplanar graphs G, the mimicking network G' they construct is furthermore outerplanar. Some of these previous results hold also for directed networks.

The only lower bound we are aware of on the size of mimicking networks is $k+1$ for every $k>3$, even for a star graph, due to [CSWZ00]. For $k=4,5$ they further show a matching upper bound. These results are summarized in Table 1. We mention that several other variants of the problem were studied in the literature, in particular when cut values are preserved approximately, see Section 1.2 for details.

1.1 Our Results[2]

The following results are going to appear in a paper in SODA 2013.

1.1.1 Upper Bounds

We first prove (in Section 2) a new upper bound for planar graphs, which significantly improves over the bound that follows from previous work (namely, 2^{2^k} known for general graphs [HKNR98]). See also Table 1 for the known bounds.

Theorem 1.3. *Every planar k-terminal network (G,c) admits a mimicking network of size at most $O(k^2 2^{2k})$, which is furthermore a minor of G.*

Notice that our theorem constructs for an input graph G a mimicking network that is actually a minor of G, and thus preserves additional properties of G such as planarity.

[2]The following results are going to appear in a paper in SODA 2013.

Graph family	Lower bounds		Upper bounds	
General graphs	$2^{\Omega(k)}$	Theorem 1.4	2^{2^k}	[HKNR98]
Planar graphs	$\lvert E(G')\rvert \geq \Omega(k^2)$	Theorem 1.5	$O(k^2 2^{2k})$	Theorem 1.3
Bounded treewidth			$O(k)$	[CSWZ00]
Star graphs	$k+1$	[CSWZ00]		

Table 1: Known bounds for the size of mimicking networks

1.1.2 Lower Bounds

We further provide (in Section 3) two nontrivial lower bounds. See Table 1 for comparison with the known bounds. The following theorem addresses general graphs, and narrows the previous doubly-exponential gap (between $k+1$ and 2^{2^k}) to be only singly-exponential.

Theorem 1.4. *For every $k > 5$ there exists a k-terminal network such that every mimicking network of it has size $2^{\Omega(k)}$. This holds even for bipartite networks with all the terminals on one side and all the non-terminals on the other side.*

The next theorem is for mimicking networks of planar graphs, proving a lower bound on the number of *edges*. If the mimicking network is guaranteed to be sparse (say planar, as is the case in our bound in Theorem 1.3) then we get a similar bound for the number of vertices. But if the mimicking network could be arbitrary (e.g., a complete graph) we do not know how to prove it cannot have $O(k)$ vertices.

Theorem 1.5. *For every $k > 5$ there exists a planar k-terminal network such that every mimicking network of it has at least $\Omega(k^2)$ edges.*

Remark. Very recently, we were informed of new results, obtained independently of ours, by Khan, Raghavendra, Tetali and Végh [KRTV12]. Their results include improved upper bounds for general graphs (albeit still doubly-exponential in k), for trees, and for bounded treewidth graphs, as well as lower bounds that are comparable to ours.

1.1.3 Succinct Data Structures

Our final result is an alternative formulation of graph compression as the problem of storing succinctly (i.e., summarizing or sketching) all the $2^{k-1}-1$ terminal cuts in a k-terminal network.

Definition 1.6. *A* terminal-cuts (TC) scheme *is a data structure that uses storage (memory) M to support the following two operations on a k-terminal network (G, c), where $n = \lvert V(G)\rvert$ and $c : E(G) \to \{1, \ldots, n^{O(1)}\}$.*

1. *Preprocessing P, which gets as input the network and builds M.*
2. *Query Q, which gets as input a subset of terminals S, and uses M (without access to (G, c)) to output* $\mathrm{mincut}_{G,c}(S, \bar{S})$.

Observe that putting together the two conditions above gives $Q(S; P(G,c)) = \text{mincut}_{G,c}(S, \bar{S})$ for all $S \subset Q$. The *storage requirement* (or *space complexity*) of the TC scheme is the (maximum) number of machine words used by M. Since the value of every cut in (G, c) is at most $n^{O(1)}$, and since we need to be able to represent every vertex in G, we shall count the size of the TC scheme in terms of machine words of $O(\log n)$ bits. An obvious upper bound is 2^k machine words, by explicitly storing a list of all the cut values. Perhaps surprisingly, we can show a matching lower bound for any data structure using the technology developed to prove Theorem 1.4. We prove the following theorem, including its extension to randomized schemes, in Section 4.

Theorem 1.7. *For every $k > 5$, a terminal-cuts scheme for k-terminal networks requires storage of $2^{\Omega(k)}$ machine words.*

This theorem is related to, but different from, Theorem 1.4. A TC scheme can possibly use its memory M to store an entire mimicking network; a more naive approach would be to store all the terminal-cut values, using at most 2^k machine words. Indeed, our theorem shows that the worst-case memory usage of this naive approach is essentially optimal.

1.2 Related Work

Graph compression can be interpreted quite broadly, and indeed it was studied extensively in the past, with many results known for different graphical features (the properties we wish to preserve). For instance, in the context of preserving the graph distances, concepts such as spanners [PS89] and probabilistic embedding into trees [AKPW95, Bar96], have developed into a rich area with productive area, and variations of it that involve a subset of terminal vertices were studied more recently, see e.g. [CE06, KZ12].

In the context of preserving cuts (and flows), which is also our theme, the problem of graph sparsification [BK96] has recently seen an immense progress, see [BSS09] and references therein. Even closer to our own work are analogous questions that involve a subset of terminals, and the goal is to find a small network that preserves (the cost of) all minimum terminal cuts *approximately*. In particular, Chuzhoy [Chu12] recently showed a constant factor approximation using a network whose size depends on (certain) edge-costs is in the original graph. Another variation of our problem is that of a cut (and flow) sparsifier, in which the compressed network should contain only k vertices (the terminals) and the goal is to minimize the approximation factor (sometimes called congestion), see [CLLM10, EGK$^+$10, MM10] for the latest results.

2 Upper Bound for Planar Graphs

In this section we prove Theorem 1.3, showing that every planar k-terminal network (G, c) admits a mimicking network of size $O(k^2 2^{2k})$, which is in fact a minor of G.

2.1 Technical Outline

Let G be a planar k-terminal network, and assume it is connected. Let $E_S = \text{min-cutset}_{G,c}(S, \bar{S})$ be the cutset of a minimum S-separating cut in (G, c), and let $\hat{E}$ be the union of E_S over all subsets $S \subset Q$. Removing the edges $\hat{E}$ from the graph G disconnects it to some number of connected components, and we construct our mimicking network G' by contracting every such connected component into a single vertex. It is easy to verify that these contractions maintain the minimum terminal cuts. This method of constructing G' resembles the one in [HKNR98], except that the sets of vertices that we unite are always connected, hence our G' is a minor of G. We proceed to bound the number of connected components one gets in this way, as this will clearly be the size of our mimicking network G'.

We first consider removing from G a single cutset E_S (for arbitrary $S \subset Q$), and show (in Lemma 2.1) that it can disconnect the graph into at most k connected components. We then extend this result to removing from G two cutsets, namely E_S and E_T (for arbitrary $S, T \subset Q$), and show (in Lemma 2.2) such a removal can disconnect the graph into at most $3k$ connected components. Next, we consider removing all the $m = 2^{k-1} - 1$ cutsets of the minimum terminal cuts from G (i.e., $G \setminus \hat{E}$). However, naive counting of the number of resulting connected components, which argues that every additional cutset splits each existing component into at most $O(k)$ components, would give us in total a poor bound of roughly k^m.

The crucial step here is to use the planarity of G to improve the dependence on m significantly, and we indeed obtain a bound that is quadratic in m by employing the dual graph of G denoted by G^*. Loosely speaking, the cutsets in G correspond to (multiple) cycles in 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 now use Euler's formula and the special structure of this subgraph of cycles; more specifically, we count its vertices of degree > 2, which turns out to require the aforementioned bound of $3k$ for two sets of terminals S, T. This gives us a bound on the number of faces in this subgraph (in Lemma 2.6), which in turn is exactly the number of connected components in the primal graph (Corollary 2.7).

2.2 Preliminaries

Recall that a graph is called a *multi-graph* if we allow it to have parallel edges and loops. A *cycle* in a multi-graph G is a sequence of edges $(u_0, v_0), \ldots, (u_{l-1}, v_{l-1})$ such that $v_i = u_{(i+1) \bmod l}$ for all $i = 0, \ldots, l-1$. The cycle is *simple* if it contains l distinct vertices and l distinct edges. Note that two parallel edges define a simple cycle of length 2, and that a loop is a cycle of length 1 that contributes 2 to the degree of its vertex. A *circuit* is a collection of cycles (not necessarily disjoint) $\mathcal{C} = \{C_1, \ldots, C_l\}$. Let $\mathcal{E}(\mathcal{C}) = \bigcup_{i=1}^{l} C_i$ 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 cost of a circuit $\mathcal{C}$ is defined as $\sum_{e \in \mathcal{E}(\mathcal{C})} c(e)$.

For a graph G, let $CC(G)$ denote the set of connected components in the graph. In particular, if $CC(G) = \{P_1, \ldots, P_h\}$ then $V(G) = P_1 \cup \cdots \cup P_h$ as a disjoint union. For a subset of the vertices $W \subset V(G)$, let $\delta(W)$ denote the set of edges with exactly one endpoint in W, i.e. $\delta(W) = \{(u,v) \in E(G) : \ u \in W, v \notin W\}$. A vertex in G with degree more than 2 will be called a *meeting* vertex of G. We introduce special notation for two (disjoint) sets of vertices:

$$V_2(G) = \{v \in V : \ \deg(v) = 2\};$$
$$V_m(G) = \{v \in V : \ \deg(v) > 2\};$$

and for two (disjoint) sets of edges:

$$E_2(G) := \{(u,v) \in E(G) : \ u, v \in V_2(G)\};$$
$$E_m(G) := \{(u,v) \in E(G) : \ u \in V_m(G) \text{ or } v \in V_m(G)\}.$$

2.3 Proof of Theorem 1.3

Let (G, c) be a k-terminal network with terminals $Q = \{q_1, \ldots, q_k\}$, where G is a connected plane graph with faces F (if G is not connected we can apply the proof for every connected component separately). We may assume, using small perturbation on the edges cost, that every two different subsets of edges in G have different total cost. In the proof we will use the notations E_S and $\hat{E}$ defined in Section 2.1.

Lemma 2.1 (One cutset)**.** *For every subset of terminals S, the graph $G \setminus E_S$ has at most k connected components.*

Proof. If there are more than k connected components then there is at least one connected component without any terminal vertex. Since G is connected, we can unite it to any other connected component by removing some edge from E_S. We get a new cutset that separates S from $\bar{S}$ with smaller total cost than E_S in contradiction to the minimality. □

Lemma 2.2 (Two cutsets)**.** *For every two subsets of terminals S and T, the graph $G \setminus (E_S \cup E_T)$ has at most $|CC(G \setminus E_S)| + |CC(G \setminus E_T)| + k$ connected components.*

We illustrate this lemma in Figure 1. The idea is that if $G \setminus (E_S \cup E_T)$ has too many connected components, then we can find one that contains no terminals, and that moving it to the other side of (say) $G \setminus E_S$ contradicts the minimality of E_S.

Proof of Lemma 2.2. Let $CC(G \setminus (E_S \cup E_T)) = \{P_0, \ldots, P_h\}$. For every P_i, we let $W_S(P_i) := \delta(P_i) \cap E_S$ be the set of edges in E_S that have exactly one of their endpoints in P_i, and similarly $W_T(P_i) := \delta(P_i) \cap E_T$. We can use the above notation to associate every connected components P_i of $G \setminus (E_S \cup E_T)$, to one of the following four sets:

1. $W_S(P_i) = \emptyset$; in particular, $P_i \in CC(G \setminus E_T)$.
2. $W_T(P_i) = \emptyset$; in particular, $P_i \in CC(G \setminus E_S)$.
3. $W_S(P_i) = W_T(P_i)$; in particular $P_i \in CC(G \setminus E_S) \cap CC(G \setminus E_T)$.

4. $W_S(P_i) \neq \emptyset$, $W_T(P_i) \neq \emptyset$ and $W_S(P_i) \neq W_T(P_i)$; in particular $P_i \notin CC(G \setminus E_S) \cup CC(G \setminus E_T)$.

Every connected component that belongs to the last set (i.e. there are at least two different edges in $\delta(P_i)$, one from E_T and one from E_S) will be called a *mixed* connected component of $G\setminus(E_S \cup E_T)$. Thus, the number of connected components in $G\setminus(E_S\cup E_T)$ is bounded by $|CC(G\setminus E_S)|+|CC(G\setminus E_S)|$ plus the number of mixed connected components of $G \setminus (E_S \cup E_T)$.

Assume towards contradiction that there are more than k mixed connected components in $G \setminus (E_S \cup E_T)$. Therefore, there exists at least one mixed connected component, say without loss of generality P_0, without any terminal in it. Since P_0 is a mixed connected component in $G \setminus (E_S \cup E_T)$ we know that $W_S(P_0) \neq \emptyset$, $W_T(P_0) \neq \emptyset$ and $W_S(P_0) \neq W_T(P_0)$. For simplicity from now on we will drop the P_0 and refer W_S and W_T to $W_S(P_0)$ and $W_T(P_0)$ correspondingly. By the perturbation on the edges cost the total cost of these two subsets must be different. Assume without loss of generality that $c(W_S) < c(W_T)$. We will replace the edges W_T by the edges W_S in the cutset of T and call this new set of edges E'_T, i.e. $E'_T = (E_T \cup W_S) \setminus W_T$. It is clear that $c(E'_T) < c(E_T)$. We will prove that E'_T is also a cutset that separate T from $\bar{T}$ in the graph G, contradicting the definition of E_T. See Figures 1 and 2.

Denote $CC(G \setminus E_T) = \{P'_0, \ldots, P'_{h'}\}$ and assume without loss of generality that the set of edges W_S connects the connected component P_0 and the t connected components $P_1, \ldots P_t$ of $G \setminus (E_S \cup E_T)$ into one connected component P'_0 in $G \setminus E_T$. Therefore, by adding the edges $E_S \setminus (W_S \cup W_T)$ to the graph $G \setminus (E_S \cup E_T)$ We will get the graph $G' = G\setminus(E_T \cup W_S)$ and its connected components will be $P_0, P_1, \ldots, P_t, P'_1, \ldots, P'_{h'}$. Since the graph G' do not contains any edge from E_T, the sets T and $\bar{T}$ are still separated.

Now it remain to add the edges W_T to the graph G' in order to get the desirable graph $G \setminus E'_T$. Assume without loss of generality that P'_0 contains terminals from T. Then, by the minimality of E_T, if edges from W_T connect between P'_0 and P'_i, then the terminals of P'_i are from $\bar{T}$. In particular, adding the edges W_T to G' will connect P_0 to some connected components P'_i that contains only terminals from $\bar{T}$. Since P_0 does not contains any terminals, the connected component that was combined by the edges W_T contains only terminals from $\bar{T}$, and so E'_T separate between T and $\bar{T}$. □

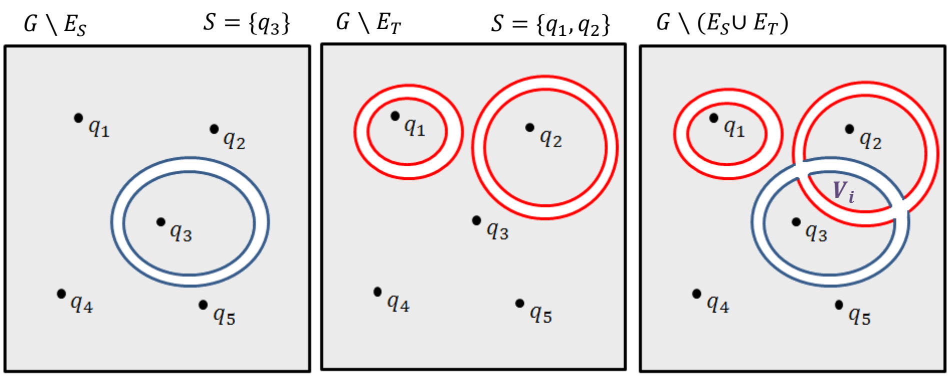

Figure 1: As depicted in gray, $G \setminus E_S$ has two connected components, $G \setminus E_T$ has three, and $G \setminus (E_S \cup E_T)$ has five. Notice the connected component V_i of $G \setminus (E_S \cup E_T)$ contains no terminals.

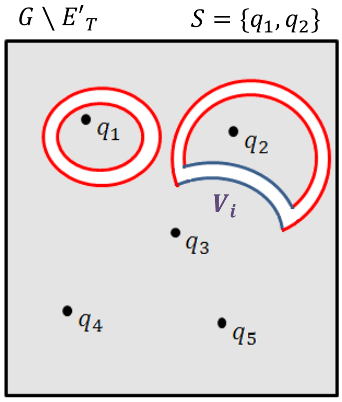

Figure 2: $E'_T = (E_T \cup W_S) \setminus W_T$, where the red edges we removed are W_T, and the blue edges we added are W_S.

Planar duality. 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 $F(G^*) = \{f_v^* : v \in V(G)\}$. Every edge $e = (v,u) \in E(G)$ with cost $c(e)$ 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 cost $c(e^*) = c(e)$ that lies on the boundary of the faces f_v^* and f_u^*. For every subset of edges $H \subset E(G)$, let $H^* := \{e^* : e \in H\}$ 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 duals are cycles) to three or more terminals. We are not aware of a reference for this precise statement, although it is similar to [HS85, Rao87]. See also Figure 3 for illustration.

Theorem 2.3 (Duality of 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, $H \subset E(G)$ is a cutset in G that has minimum cost among those separating W from $V(G) \setminus W$ if and only if the dual set of edges $H^* \subseteq E(G^*)$ is actually $\mathcal{E}(\mathcal{C})$ for a circuit $\mathcal{C}$ in G^* that has minimum cost among those separating the corresponding faces $\{f_v^* : v \in W\}$ from $\{f_v^* : v \in V(G) \setminus W\}$.*

Recall that removing edges from a graph G disconnects it into (one or more) connected components. The next lemma characterizes this behavior in terms of the dual graph G^*. See Figure 4 for illustration. Recall that $G[H]$ is a standard notation for the subgraph of G induced by the subset (of edges or vertices) H.

Lemma 2.4 (Dual of a connected component)**.** *Let G be a connected plane multi-graph, let G^* be its dual, and fix a subset of edges $H \subset E(G)$. Then P is a connected component in $G \setminus H$ if and only if its dual set of faces $\{f_v^* : v \in P\}$ is a face of $G^*[H^*]$.*

Leveraging the planarity. We proceed with the proof of Theorem 1.3, and now use the duality of planar graphs. In the following corollary we will deal with the dual graph $G^*[E_S^* \cup E_T^*]$ for two arbitrary subsets of terminals S and T.

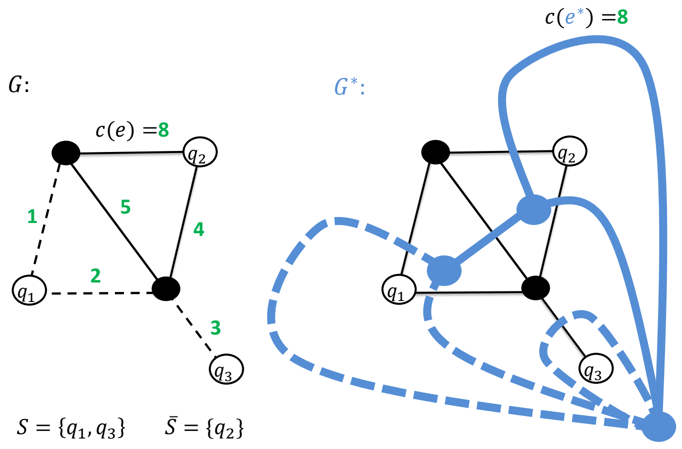

Figure 3: A planar 3-terminal network G (in black), with E_S depicted as dashed edges. The dual graph G^* is shown in blue, with E_S^* depicted as dashed edges.

Corollary 2.5. *For all $S, T \subset Q$, the graph $G^*[E_S^* \cup E_T^*]$ has at most $6k$ meeting vertices.*

Proof. According Lemmas 2.1 and 2.2, the graph $G \setminus (E_S \cup E_T)$ has at most $|CC(G \setminus E_S)| + |CC(G \setminus E_T)| + k \leq 3k$ connected components. By Lemma 2.4 every connected component in $G \setminus (E_S \cup E_T)$ corresponds to a face in $G^*[E_S^* \cup E_T^*]$. Therefore, $G^*[E_S^* \cup E_T^*]$ has at most $3k$ faces.

By the duality of cuts and circuits, every set of edges E_S^* is a circuit. Therefore, every vertex v appearing in these edges E_S^*, has degree at least 2. $E_S^* \cup E_T^*$ is circuit as well, and all its vertices have degree at least 2, i.e. $V(G^*[E_S^* \cup E_T^*]) = V_2(G^*[E_S^* \cup E_T^*]) \cup V_m(G^*[E_S^* \cup E_T^*])$. To simplify the notation we denote $G_{ST}^* = G^*[E_S^* \cup E_T^*]$. By Handshaking lemma,

$$\begin{aligned} 2|E(G_{ST}^*)| &= \sum_{v \in V(G_{ST}^*)} \deg(v) \\ &\geq 3|V_m(G_{ST}^*)| + 2|V_2(G_{ST}^*)| \\ &= 2|V(G_{ST}^*)| + |V_m(G_{ST}^*)|. \end{aligned}$$

By Euler's formula

$$\begin{aligned} 3k &\geq |F(G_{ST}^*)| \\ &= |E(G_{ST}^*)| - |V(G_{ST}^*)| + |CC(G_{ST}^*)| + 1 \\ &\geq \frac{1}{2}|V_m(G_{ST}^*)|, \end{aligned}$$

and the corollary follows. □

Recall that in Section 2.1 we defined $\hat{E} := \bigcup_{S \subset Q} E_S$, and denote its set of dual edges by $\hat{E}^* := \{e^* : \ e \in \hat{E}\} = \bigcup_{S \subset Q} E_S^*$.

Lemma 2.6. *The graph $G^*[\hat{E}^*]$ has at most $O(k^2 2^{2k})$ faces.*

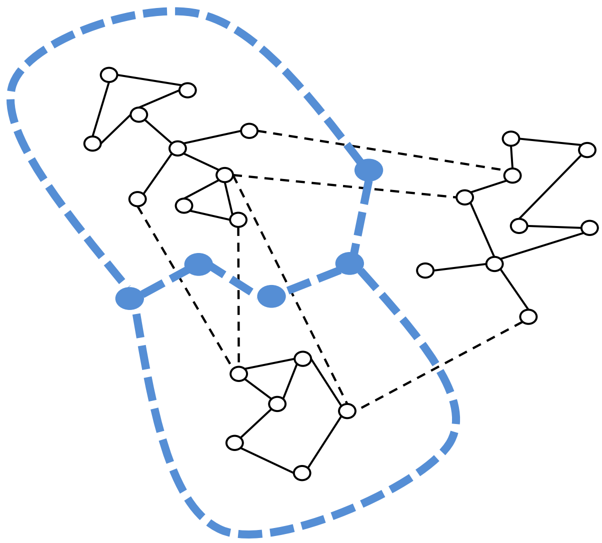

Figure 4: The graph G is in black. Removing the black dashed edges H disconnects the graph G into three connected components. The blue bold dashed edges are the dual edges H^*, that form the dual subgraph $G^*[H^*]$.

Proof. Using Theorem 2.3 we get that for every $S \subset Q$, E_S is a minimum cutset in G if and only if E_S^* (the dual set of edges of E_S) is a minimum circuit in G^*. Moreover, as defined in Section 2.3 $\hat{E}^* = \bigcup_{S \subset Q} E_S^*$. Thus, $\hat{E}^*$ is also a circuit, and so

$$|V(G^*[\hat{E}^*])| = |V_2(G^*[\hat{E}^*])| + |V_m(G^*[\hat{E}^*])|, \tag{1}$$

$$|E(G^*[\hat{E}^*])| = |E_2(G^*[\hat{E}^*])| + |E_m(G^*[\hat{E}^*])|. \tag{2}$$

According to the definitions and the Handshaking lemma we get that

$$|E_2(G^*[\hat{E}^*])| \leq |V_2(G^*[\hat{E}^*)|. \tag{3}$$

By a union bound, the two following inequalities holds

$$|V_m(G^*[\hat{E}^*])| \leq \sum_{S \subset Q} |V(G^*[E_S^*]) \cap V_m(G^*[\hat{E}^*])| \tag{4}$$

$$|E_m(G^*[\hat{E}^*])| \leq \sum_{S \subset Q} |E(G^*[E_S^*]) \cap E_m(G^*[\hat{E}^*])| \tag{5}$$

Fix a subset S. We will start by bounding the set of vertices $V(G^*[E_S^*]) \cap V_m(G^*[\hat{E}^*])$. For every vertex v in $V(G^*[E_S^*]) \cap V_m(G^*[\hat{E}^*])$ there exists a subset T such that v is also in $V(G^*[E_S^*]) \cap V_m(G^*[E_S^* \cup E_T^*])$. According to Corollary 2.5, $|V_m(G^*[E_S^* \cup E_T^*])| \leq 6k$. Therefore $|V(G^*[E_S^*]) \cap V_m(G^*[E_S^* \cup E_T^*])| \leq 6k$, and by union bound on all the subsets T we get $|V(G^*[E_S^*]) \cap V_m(G^*[\hat{E}^*])| \leq 6k2^k$.

We will now move to bound $E(G^*[E_S^*]) \cap E_m(G^*[\hat{E}^*])$. By Lemma 2.1 there are at most k cycles that cover the graph $G^*[E_S^*]$, so every vertex in $V(G^*[E_S^*]) \cap V_m(G^*[\hat{E}^*])$ can be shared by at most k cycles of $G^*[E_S^*]$, which bound the degree of every vertex in $G^*[E_S^*]$ by $2k$. Thus

$$|E(G^*[E^*_S]) \cap E_m(G^*[\hat{E}^*])| \le 2k|V(G^*[E^*_S]) \cap V_m(G^*[\hat{E}^*])| = O(k^2 2^k) \quad (6)$$

We can bound $|CC(G^*[\hat{E}^*])|$ by extending the argument in Lemma 2.1. Assume toward contradiction that $|CC(G^*[\hat{E}^*])| \ge k+1$. Thus, there exists at least one connected component P in $G^*[\hat{E}^*]$ that does not contains any terminal face of G^*. By the construction of $\hat{E}^*$, there exists a subset S such that P contains at least one cycle C of the circuit E^*_S. Since P does not contain any terminal face, we can remove some edge e^* of the cycle C from the circuit E^*_S and get a circuit with smaller cost that separates between f^*_S and $f^*_{\bar{S}}$ in contradiction.

Now by Euler's formula,

$$\begin{aligned}
&|F(G^*[\hat{E}^*])| \\
&\quad = |E(G^*[\hat{E}^*])| - |V(G^*[\hat{E}^*])| + 1 + |CC(G^*[\hat{E}^*])| \\
&\quad \le |E_m(G^*[\hat{E}^*])| - |V_m(G^*[\hat{E}^*])| + 1 + k \\
&\quad \le \textstyle\sum_{S \subset Q} O(k^2 2^k) + 1 + k = O(k^2 2^{2k})
\end{aligned}$$

the first inequality is by Equations (1), (2) and (3), the second inequality is by Equations (5) and (6), and the lemma follows. □

Corollary 2.7. *There are at most $O(k^2 2^{2k})$ connected components in the graph $G \setminus \hat{E}$.*

This corollary follows from Lemma 2.6 by applying Lemma 2.4 with $H = \hat{E}$. We now complete the proof of Theorem 1.3. Merge the vertices in each connected component of $G \setminus \hat{E}$ into a single vertex (formally, contract all the internal edges in each connected component) and call this new multi-graph M. Notice there is at most one terminal vertex in each connected component. So a vertex in M, which corresponds to a connected component (of $G \setminus \hat{E}$) that contains some terminal vertex q, will be identified with that terminal q. To be concrete, the vertices and the terminals of M are the sets

$$V(M) := \{v_i : \ P_i \in CC(G \setminus \hat{E})\}$$
$$Q(M) := \{q = v_i : \ P_i \in CC(G \setminus \hat{E}) \text{ and } q \in P_i\}$$

In addition, (v_i, v_j) is an edge in M if there exist two vertices $u_i, u_j \in E(G)$ such that $u_i \in P_i$, $u_j \in P_j$ and (u_i, u_j) is an edge in G. The cost of every edge $(v_i, v_j) \in E(M)$ is

$$c'(v_i, v_j) := \sum_{u_i \in P_i, u_j \in P_j: \ (u_i, u_j) \in E(G)} c(u_i, u_j).$$

It is easy to verify that M is a minor of G with $O(k^2 2^{2k})$ vertices that includes the same k terminals Q. We now prove that (M, c') is a mimicking network of G using the same argument as in [HKNR98], but applied to the connected components. Fix a subset of terminals S. Since we only contract edges, every cut that separates S and $\bar{S}$ in M has a cut in G that separates S and $\bar{S}$ with the same cost, thus $\text{mincut}_{M,c'}(S, \bar{S}) \ge \text{mincut}_{G,c}(S, \bar{S})$. In the other direction, notice that by the construction of M, all the vertices in each connected component of $G \setminus \hat{E}$ are on the same side of the minimum S-separating cut in G. Thus, there is a cut in M that separates between S and $\bar{S}$ and has cost $\text{mincut}_G(S, \bar{S})$. Combining these together, we get the equality $\text{mincut}_{M,c'}(S, \bar{S}) = \text{mincut}_{G,c}(S, \bar{S})$ for every S, and Theorem 1.3 follows.

3 Lower Bounds

In this section we prove Theorems 1.4 as well as Theorem 1.5.

3.1 Techniques and Proof Outline

All our lower bounds are proved using the same technique, which basically counts the number of "degrees of freedom" needed to express all the relevant cut values. Formally, we develop a certain machinery based on linear algebra, which relates the size of any mimicking network to the rank of some matrix.

The lower bound proofs start by describing a k-terminal network (G, c) that seems minimal in the sense that it does not admit a smaller mimicking network. The networks used in Theorems 1.4 and 1.5 are different. We then identify the minimum cost S-separating cuts for all (or some) $S \subset Q$, and capture this information in a matrix.

Definition 3.1 (Cutsets-edges incidence matrix)**.** *Let (G, c) be a k-terminal network, and fix an enumeration $S_1, \ldots, S_m$ of all $m = 2^{k-1} - 1$ distinct and nontrivial bipartitions $Q = S_i \cup \bar{S}_i$. The* cutset-edge incidence matrix *of (G, c) is the matrix $A_{G,c} \in \{0, 1\}^{m \times E(G)}$ given by*

$$(A_{G,c})_{i,e} = \begin{cases} 1 & \text{if } e \in \text{min-cutset}_{(G,c)}(S_i, \bar{S}_i); \\ 0 & \text{otherwise.} \end{cases}$$

We also define the vector of minimum-cut values between every bipartition of terminals

$$\Phi_{G,c} = \begin{pmatrix} \text{mincut}_{G,c}(S_1, \bar{S_1}) \\ \vdots \\ \text{mincut}_{G,c}(S_m, \bar{S_m}) \end{pmatrix} \in \mathbb{R}^m.$$

Here and throughout, we shall omit the subscript c when it is clear from the context. Observe that if we think of the edge costs c as a column vector $\vec{c} \in (\mathbb{R}^+)^{E(G)}$, then $A_G \cdot \vec{c} = \Phi_G$. For a given $S \subset Q$, a minimum S-separating cut $(W, V(G) \setminus W)$ is called *unique* if all other S-separating cuts have a strictly larger cost.

The core of our analysis is the next lemma, as it immediately provides a lower bound on the size of any mimicking network; the theorems would follow by calculating the rank of A_G.

Lemma 3.2 (Main Technical Lemma)**.** *Let (G, c) be a k-terminal network. Let A_G 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-costs function $\hat{c} : E(G) \to \mathbb{R}^+$, under which every mimicking network (G', c') satisfies $|E(G')| \geq \text{rank}(A_{G,c})$.*

Notice that the bound is proved not for (G, c) but rather for $(G, \hat{c})$; indeed, the edge-costs $\hat{c}$ are a small random perturbation of c. Thus, the proof of this lemma first shows that a small perturbation does not change the cutset-edge incidence matrix, i.e. $A_{G,c} = A_{G,\hat{c}}$. This is where the uniqueness property is used. Next, fix a small graph G' that can potentially be a mimicking network, but without specifying its edge-costs c'; now let $\mathcal{E}_{G'}$ be the event that $(G, \hat{c})$ admits a mimicking network of the form (G', c'). Since G' has too few edges (whose costs are undetermined/free variables), we can use linear algebra to show that $\Pr[\mathcal{E}_{G'}] = 0$. The lemma then follows by a union bound over the finitely many (unweighted) graphs G' of the appropriate size.

3.2 Proof of Lemma 3.2

We turn to proving Lemma 3.2. Recall that this lemma considers a k-terminal network (G, c), and assuming a certain (uniqueness) condition, asserts that there is for G a modified edge-costs function $\hat{c}$, under which every mimicking network must have at least $\text{rank}(A_{G,c})$ edges, where $A_{G,c}$ is a cutset-edge incidence matrix of (G, c).

The proof employs two lemmas and the following notation. For $S \subset Q$, let $\Delta_{G,c}(S) \geq 0$ be the difference between the two smallest costs among all S-separating cuts in G. Observe that if these two are not equal 0 (i.e., $\Delta_{G,c}(S) > 0$) then the minimum S-separating cut is said to be unique in G. We also denote $\Delta_{G,c} := \min_{S\subset Q} \Delta_{G,c}(S)$. Without loos of generality, if $\Delta_{G,c} > 0$ we assume that $\Delta_{G,c} > 1$ (if it is not the case, i.e. $0 < \Delta_{G,c} \leq 1$, we can multiple all the edges cost by $2\Delta_{G,c}$ without changing $A_{G,c}$).

Lemma 3.3. *For every edge-costs function $w : E(G) \to [0, \frac{1}{\Delta_{G,c}|E(G)|}]$ the cutset-edge incidence matrix of (G, c) is equal to the cutset-edge incidence matrix of $(G, c + w)$, i.e. $A_{G,c} = A_{G,c+w}$, where $c + w : e \to c(e) + w(e)$.*

Proof. Let w be an edge-costs function $w : E(G) \to [0, \frac{1}{\Delta_{G,c}|E(G)|}]$. Since (G, c) and $(G, c + w)$ have the same vertices and edges, every S_i-separating cut in (G, c) is also a S_i-separating cut in $(G, c + w)$ and vice versa. The value of every such cutset in $(G, c + w)$ is ranged from the value of this cutset in G to the value of this cutset in G plus $\frac{1}{\Delta_{G,c}}$. In particular, $\text{mincut}_{G,c}(S_i, \bar{S}_i) \leq \text{mincut}_{G,c+w}(S_i, \bar{S}_i) \leq \text{mincut}_{G,c}(S_i, \bar{S}_i) + \frac{1}{\Delta_{G,c}}$. Thus, $\text{mincut}_{G,c+w}(S_i, \bar{S}_i)$ is smaller (by at least $\Delta_{G,c} - \frac{1}{\Delta_{G,c}}$) than every cut that separates S_i and $\bar{S}_i$ in G. Therefore it must be the case that the cutsets of the minimum S_i-separating cuts in (G, c) and in $(G, c + w)$ are the same. □

We proceed with the proof of Lemma 3.2. Sample an edge-costs function $w : E(G) \to [0, \frac{1}{\Delta_{G,c}|E(G)|}]$ by independently choosing each $w(e)$ from that range uniformly at random. By the above lemma, $A_{G,c} = A_{G,c+w}$ so in the rest of the proof we will omit the edge-costs function and denote this matrix by A_G. Now we argue that every mimicking network of $(G, c + w)$ must has at least $r := \text{rank}(A_G)$ edges. Consider some network G' with $|E(G')| < r$, and let's see if it can potentially be a mimicking network of $(G, c + w)$. Notice that every edge-costs function $c' : E(G') \to \mathbb{R}^+$ for this G' yields a cutset-edge incidence matrix $A_{G',c'}$ of size $m \times (r - 1)$ (if some graph has less than $r - 1$ edges we can pad the irrelevant columns with zeros). Since this matrix has only ones and zeros in its entries, there are only $2^{m(r-1)}$ such matrices. The next lemma proves that for every fixed matrix $A \in \{0, 1\}^{m\times(r-1)}$, the probability that there exists an edge-costs function $c' : E(G') \to \mathbb{R}^+$ such that $A \cdot \vec{c'} = A_G \cdot (\vec{c} + \vec{w}) = \Phi_G$ is zero.

Lemma 3.4. *Fix a matrix $A \in \{0, 1\}^{m\times(r-1)}$, and let W_{A_G} and W_A be the span of the columns of A_G and A, respectively. If each $w(e)$ is independently sampled uniformly at random from $[0, \frac{1}{\Delta_{G,c}|E(G)|}]$, then*

$$\Pr_w[A_G \cdot (\vec{c} + \vec{w}) \in W_A] = 0.$$

Proof. Without loss of generality let the first r columns of the matrix A_G, $\{\vec{a_1}, \ldots, \vec{a_r}\}$, be the basis for the space W_{A_G}. Since $\text{rank}(A) < r = \text{rank}(A_G)$ we get that $\dim(W_A) <$

$\dim(W_{A_G})$. Thus there must be some basis vector of W_{A_G}, say without loss of generality $\vec{a_1}$, that is not in the subspace W_A and denote by $c(e_1) + w(e_1)$ its corresponding cost.

We will calculate the number of vectors in W_A that can be expressed as linear combination with the vector $\vec{a_1}$. Let $f(\alpha) = \alpha\vec{a_1} + \sum_{i=2}^{r}(c(e_i) + w(e_i))\vec{a_i}$. If there are at least two such vectors, $f(\alpha)$ and $f(\alpha')$ (where $\alpha, \alpha' \neq 0$) in W_A, then $\vec{a_1}$ will be in W_A because W_A is a subspace. So there is at most one α such that $f(\alpha) \in W_A$.

Since each $w(e_i)$ is sampled independently from a uniform distribution over $[0, \frac{1}{\Delta_{G,c}|E(G)|}]$, the probability that $c(e_1) + w(e_1) = \alpha$ is 0. By independence of $w(e_i)$ for all $i \in [r]$ we can sample $w(e_1)$ last which completes Lemma 3.4. $\square$

To complete the proof of Lemma 3.2, we will calculate the probability that there exists a mimicking network (G', c') for the network $(G, c + w)$, such that $|E(G')| < r$.

$$\begin{aligned}
&\Pr_w[\exists \text{ mimicking network } (G', c') \text{ with } |E(G')| < r] \\
&= \Pr_w[\exists \text{ 0-1 matrix } A_{G',c'} \text{ s.t. } A_{G',c'} \cdot \vec{c'} = A_G(\vec{c} + \vec{w})] \\
&\leq \Pr_w[\exists \text{ 0-1 matrix } A_{G',c'} \text{ s.t. } A_G(\vec{c} + \vec{w}) \in W_{A_{G',c'}}] \\
&\leq \sum_{A \in \{0,1\}^{m \times (r-1)}} \Pr_w[A_G \cdot (\vec{c} + \vec{w}) \in W_A] = 0,
\end{aligned}$$

where the first equality is by the definition of a mimicking network, the following inequality is because the condition is necessary (but not sufficient), the second inequality is by a union bound over all possible matrices, and the final equality is by Lemma 3.4. Denoting $\hat{c} = c + w$, we see that every mimicking network (G', c') for the network $(G, \hat{c})$ has at least $\text{rank}(A_G)$ edges. Lemma 3.2 follows.

3.3 Lower bound for general graphs

We now prove Theorem 1.4 which asserts that for every k there exists a k-terminal network such that its mimicking network must have $2^{\Omega(k)}$ non-terminals. The proof constructs a bipartite k-terminal network, with all its terminals on one side and all its non-terminals on the other side. As we will show, the rank of its cutset-edge incidence matrix is at least $2^{\Omega(k)}$, and the corresponding cuts are unique, hence applying Lemma 3.2 to this matrix will complete the proof of Theorem 1.4.

Proof of Theorem 1.4. Consider a complete bipartite graph $G = (Q, U, E)$, where one side of the graph consists of the k terminals $Q = \{q_1, \dots, q_k\}$, the other side of the graph consists of $l = \binom{k}{\frac{2}{3}k}$ non-terminals $U = \{u_{S_1}, \dots, u_{S_l}\}$, with $S_1, \dots, S_l$ denoting the different subsets of terminals of size $\frac{2}{3}k$. 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 $2 + \varepsilon$ to every terminal in $\bar{S}_i = Q \setminus S_i$, for sufficient small $\varepsilon > 0$, in fact $\varepsilon = \frac{1}{k}$ suffices. Let $c(u_{S_i}, q_j)$ denote the cost of edge (u_{S_i}, q_j), and define $c(u_{S_i}, S_j) := \sum_{q \in S_j} c(u_{S_i}, q)$.

Lemma 3.5. *The minimum S_i-separating cut is obtained uniquely by the cut $(W, V(G) \setminus W)$ where $W = \{u_{S_i}\} \cup \bar{S}_i$ and $V(G) \setminus W = \{u_{S_j} : j \neq i\} \cup S_i$.*

Proof. First, notice that for every $i \in [l]$ the total cost of all edges incident to u_{S_i} is

$$c(u_{S_i}, Q) = c(u_{S_i}, S_i) + c(u_{S_i}, \bar{S}_i) = \frac{2k}{3} \cdot 1 + \frac{k}{3}(2+\varepsilon) = \frac{4k}{3} + \frac{k\varepsilon}{3} \tag{7}$$

Consider such a set S_i, and let us calculate the minimum S_i-separating cut. Since non-terminals are not connected to each other, the decision is done separately for every non-terminal u_{S_j} by simply comparing the costs of the edges (u_{S_j}, S_i) versus $(u_{S_j}, \bar{S}_i)$. The crucial observation is that for non-terminal u_{S_i}:

$$c(u_{S_i}, S_i) = |S_i| \cdot 1 = \frac{2k}{3} < (2+\varepsilon)|\bar{S}_i| = c(u_{S_i}, \bar{S}_i)$$

For a non-terminal u_{S_j} where $i \neq j$,

$$c(u_{S_j}, S_i) = |S_j \cap S_i| \cdot 1 + |S_i \setminus S_j| \cdot (2+\varepsilon) = |S_i| \cdot 1 + |S_i \setminus S_j| \cdot (1+\varepsilon) > \frac{2k}{3} + 1 > c(u_{S_j}, \bar{S}_i)$$

where the last inequality is by (7) and because we choose ε such that $\frac{k\varepsilon}{3} < 1$. It follows that for every S_i the minimum S_i-separating cut will be $\{u_{S_i}\} \cup \bar{S}_i$ on one side, and $\{u_{S_j} : j \neq i\} \cup S_i$ on the other side, and moreover it is the unique minimizer. □

Lemma 3.6. *Let A_G be a cutset-edge incidence matrix of G.Then* $\text{rank}(A_G) \geq l$.

Proof. By definition, A_G is a matrix of size $m \times kl$. Since $\binom{k}{\frac{2}{3}k} = l \leq m = 2^{k-1} - 1$, we need to show that l rows of A_G are linearly independent. Assume without loss of generality that the first l rows of A_G corresponds to the l subsets of terminals of size $\frac{2}{3}k$, such that row t corresponds to subset S_t. We will prove that these l first rows of A_G are linearly independent, i.e. $\sum_{t=1}^{l} \alpha_t A_{Gt} = \bar{0} \iff \alpha_1 = \ldots = \alpha_l = 0$. We will focus on a column j in A_G that corresponds to some edge (u_{S_i}, q) where $q \in S_i$. In order to know how the j-th column in A_G looks like, we need to know in which minimum cuts the edge (u_{S_i}, q) participates, i.e. we go over all the rows of A_G and in each row t we will ask if the edge is in the cutset of the minimum S_t-separating cut or not (if there is 1 or 0 in $(A_G)_{t,j}$).

According to the construction of G, if $q \in S_i$, then the terminal q and the non-terminal u_{S_i} are in different sides of the minimum S_i-separating cut, and the edge (u_{S_i}, q) in that cutset. For some subset S_t, where $t \neq i$ and $q \in \bar{S}_t$, the side of the minimum cut that contains the terminal q will be $\{u_{S_t}\} \cup \bar{S}_t$, and the other side that contains u_{S_i} will be $\{u_{S_f} : f \in [l], f \neq t\} \cup S_t$. Then again, the edge (u_{S_i}, q) will be in that cutset. It remain to look on some subset S_t, where $t \neq i$ and $q \in S_t$. The cut will be the same as above, but now both of the vertices will be in one side of the cut, i.e. $q, u_{S_i} \in \{u_{S_f} : f \in [l], f \neq t\} \cup S_t$, so the edge (u_{S_i}, q) will not participate in this cutset. In conclusion, The edge (u_{S_i}, q) participate in the cutset of the minimum S_i-separating cut, and in all the cutsets of the minimum S_t-separating cut such that S_t do not contains the terminal q. Hence we will get that the entry j (the column of A_G that corresponds to the edge (u_{S_i}, q)) in the vector $\sum_{t=1}^{l} \alpha_t A_{Gt}$ is:

$$\left(\sum_{t=1}^{l} \alpha_t A_{Gt}\right)_j = \sum_{t=1}^{l} \alpha_t (A_G)_{t,j} = \alpha_i + \sum_{t \in [l]:\ q \notin S_t} \alpha_t = 0 \tag{8}$$

Every two different subsets S_i and $S_{i'}$, have at least $\frac{1}{3}k$ terminals in common. In particular there exist some terminal q contained in both of them. Looking at the entries corresponding to (u_{S_i}, q) and $(u_{S_{i'}}, q)$ in the vector $\sum_{h=1}^{l} \alpha_h A_{Gh}$ we have

$$\alpha_i + \sum_{t\in[l]:\ q\notin S_t} \alpha_t = 0$$

$$\alpha_{i'} + \sum_{t\in[l]:\ q\notin S_t} \alpha_t = 0$$

Thus $\alpha_i = \alpha_{i'}$ for every $i, i' \in [l]$. So we get the equation $\binom{k-1}{\frac{2}{3}k-1}\alpha_1 = 0$ in every entry in the vector equation $\sum_{t=1}^{l} \alpha_t A_{Gt} = 0$, and Lemma 3.6 follows. □

We can now complete the proof of Theorem 1.4. Applying Lemma 3.2 to our bipartite graph G and its cutset-edge incidence matrix A_G, we get that every mimicking network G' of G has at least $l = 2^{\Omega(k)}$ edges. It follows that $|V(G')| \geq \sqrt{|E(G')|} \geq 2^{\Omega(k)}$. □

3.4 Lower bound for planar graphs

In this section we prove Theorem 1.5, which shows a planar k-terminal network, every mimicking network of which must have at least k^2 edges. The proof constructs a grid of size $O(k^2)$ with $2k$ terminals, and applies Lemma 3.2 on graph's cutset-edge incidence matrix.

Proof of Theorem 1.5. Construct a planar $2k$-terminal network G with $2k$ terminals $Q = \{v_1, \ldots, v_k, h_1, \ldots, h_k\}$ as follows. Consider a grid with k columns and k rows. Let $u_{i,j}$ be the non-terminal vertex at the ith column and jth row of the grid. To every vertex $u_{1,j}$, for $1 \leq j \leq k$, we attach a terminal vertex v_j of degree one, and at every vertex $u_{i,1}$, for $1 \leq i \leq k$, we attach a terminal vertex h_i of degree one. From now on, we will refer to i and j as indices between 1 to k, including 1, excluding k.

The costs associated with the edges of G are as follows: every edge that connects between a terminal to a non-terminal costs k^4. The cost of all the edges between the vertices $u_{i,k}$ and $u_{i+1,k}$, and between the vertices $u_{k,j}$ and $u_{k,j+1}$, is k^4. All the remaining vertical edges will have cost 1, i.e. all the edges between $u_{i,j}$ and $u_{i+1,j}$. All the remaining horizontal edges, i.e. every edge between $u_{i,j}$ and $u_{i,j+1}$, will cost $1 - \varepsilon_{i,j}$, where $\varepsilon_{i,j} = \frac{j}{k^4}$. Notice that for every $k > 2$ the sum of all the $\varepsilon_{i,j}$ in G is

$$\sum_{i,j=1}^{k-1} \varepsilon_{ij} \leq \frac{1}{k^4} \sum_{i,j=1}^{k} 2k = \frac{2k^3}{k^4} < 1 \tag{9}$$

Denote by $S_{i,j}$ the subset of the terminals $\{h_1, \ldots, h_i, v_1, \ldots, v_j\}$. We are interested in all the $(k-1)^2$ minimum $S_{i,j}$-separating cuts. See the grid G in Figure 5.

Lemma 3.7. *The minimum $S_{i,j}$-separating cut is obtained uniquely by the cut $(W, V(G) \setminus W)$ where $W = S_{i,j} \cup \{u_{\alpha,\beta} : 1 \leq \alpha \leq i, 1 \leq \beta \leq j\}$.*

Proof. Let $c_{i,j}$ be the cost of the $S_{i,j}$-separating cut $(W, V(G) \setminus W)$ described in the lemma. By a simple calculation, $c_{i,j} = i + j - \sum_{\alpha=1}^{i} \varepsilon_{\alpha,j}$. Assume towards contradiction that the above cut $(W, V(G) \setminus W)$ is not the minimum $S_{i,j}$-separating cut in G, i.e.

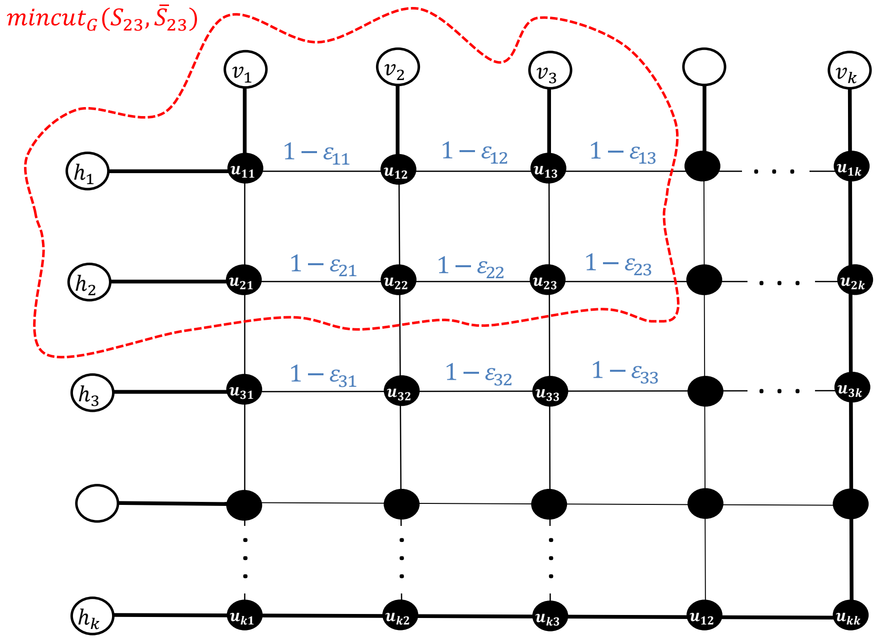

Figure 5: The $2k$-terminal network, which used in Theorem 1.5, with minimum S_{23}-separating cut (the red dashed line). All the vertical and horizontal bold edges has cost k^4, the remaining horizontal edges has cost $1-\varepsilon_{i,j}$ and all the remaining vertical edges has cost 1.

$\text{mincut}_G(S_{i,j}, \bar{S}_{i,j}) < c_{i,j} < k$. Thus all the edges that are contained in $\text{min-cutset}_G(S_{i,j}, \bar{S}_{i,j})$ have costs less then k. In particular, the edges with cost k^4 are not contained in $\text{mincut}_G(S_{i,j}, \bar{S}_{i,j})$, so the two terminals v_k and h_k are connected (which means, not disconnected when we remove that cutset).

The cut $(W, V(G) \setminus W)$ contains i horizontal edges and j vertical edges. This is the minimal number of vertical and horizontal edges that need to be removed in the minimum cut in order to separate $S_{i,j}$ from $\bar{S}_{i,j}$. Otherwise, if we remove less then i horizontal edges, there must be some terminal, h_α, in $S_{i,j}$, such that no horizontal edges were removed from its row, thus h_α connected to the terminals v_k and h_k that in $\bar{S}_{i,j}$. The argument for j vertical edges is similar.

Another observation is that the total cost of every $i+j+1$ or more edges in G (with cost less then k^4) is not less than $i+j+1-\sum_{\alpha,\beta=1}^{k} \varepsilon_{\alpha,\beta} > i+j$, where the inequality is by Equation (9). We conclude that the minimum cut has exactly j vertical edges and i horizontal edges.

By now we know that the cutset $\text{min-cutset}_G(S_{i,j}, \bar{S}_{i,j})$ contains i horizontal edges and j vertical edges. Furthermore, we know that the cutset $(W, V(G) \setminus W)$ contains the first i horizontal edges between the jth column to the $(j+1)$st column , and the first j vertical edges between the ith row to the $(i+1)$st row. Thus, $\text{min-cutset}_G(S_{i,j}, \bar{S}_{i,j})$ must contains at least one different edge than the cut $(W, V(G) \setminus W)$. There are two cases:

1. If $\text{min-cutset}_G(S_{i,j}, \bar{S}_{i,j})$ contains at least one vertical edge on some column $\beta > j$, then it contains no more than $j-1$ vertical edges from the columns between 1 to j. As before, there exist some terminal that is connected to at least one terminal from

$\bar{S}_{i,j}$. The same argument works for horizontal edge that removed from row $\alpha > i$. Hence, this case is impossible.

2. If all the edges that participate in min-cutset$_G(S_{i,j}, \bar{S}_{i,j})$ are from the first i rows and first j columns. We will calculate the minimal value of a cut that we can obtain. As mentioned above, in order to separate we need to remove one edge from every column and from every row. The cost of all the vertical edges is identical so already need to pay j. Notice that in every row α the following inequality chain holds

$$\varepsilon_{\alpha,1} < \varepsilon_{\alpha,2} < \ldots < \varepsilon_{\alpha,k-1}$$

Therefore, the cost of the cheapest edge that we can take from that row is $1 - \varepsilon_{\alpha,j}$. Summing all these costs we get $j + \sum_{\alpha=1}^{i}(1 - \varepsilon_{\alpha,j})$.

From the second case we get that mincut$_G(S_{i,j}, \bar{S}_{i,j}) = c_{i,j}$, and that the cut $(W, V(G)\backslash W)$ is the only cut with that value as we wanted. □

Proceeding with the proof of Theorem 1.5, let A_G be a cutset-edge incidence matrix of G (see Definition 3.1).

Lemma 3.8. $\text{rank}(A_G) \geq (k-1)^2$

Proof. Assume without loss of generality that the first $(k-1)^2$ columns of A_G correspond to all the horizontal edges that their cost involve an $\varepsilon_{i,j}$ variable. We will order them according to their order in the grid from left to right, up to down. i.e. the first $(k-1)^2$ columns of A_G will correspond to the edge costs in the following order:

$$1 - \varepsilon_{1,1}\ , \ldots ,\ 1 - \varepsilon_{1,k-1}\ ,\ 1 - \varepsilon_{2,1}\ , \ldots ,\ 1 - \varepsilon_{2,k-1}\ , \ldots ,\ 1 - \varepsilon_{k-1,1}\ , \ldots ,\ 1 - \varepsilon_{k-1,k-1}$$

In addition, without loss of generality the first $(k-1)^2$ rows of A_G correspond to the $(k-1)^2$ minimum $S_{i,j}$-separating cuts in G which deals with the $(k-1)^2$ subsets of terminals we are interested in according to the following order:

$$S_{1,1}\ , \ldots ,\ S_{1,k-1}\ ,\ S_{2,1}\ , \ldots ,\ S_{2,k-1}\ , \ldots ,\ S_{k-1,1}\ , \ldots ,\ S_{k-1,k-1}$$

We will show that the sub matrix of A_G formed by first $(k-1)^2$ rows and columns of A_G is a lower triangular matrix, which imply that the first $(k-1)^2$ columns are linearly independent. Given column t that corresponds to $1 - \varepsilon_{ij}$, we need to show that the entry t, t is 1, and all the $t-1$ first entries are 0. As we set above, the t-th row of A_G corresponds to the minimum $S_{i,j}$-separating cut. According to Lemma 3.7 the total costs of the horizontal edges that participate in the minimum $S_{i,j}$-separating cut is $\sum_{\alpha=1}^{i}(1 - \varepsilon_{\alpha,j})$. Thus it is clear that entry t, t is 1, because the edge $1 - \varepsilon_{ij}$ participates in the minimum $S_{i,j}$-separating cut. It remains to show that all the $t-1$ first entries are 0. All the first $t-1$ rows correspond to subsets of terminals $S_{\alpha,\beta}$ such that $\alpha < i$ or $\alpha = i$ and $\beta < j$. As we saw above, the edge $1 - \varepsilon_{i,j}$ participates only in all the minimum cuts of the subsets $S_{\alpha,j}$ where $\alpha \geq i$. Thus, there is 0 in all the first $t-1$ entries in the t-th column. So we prove that the first $(k-1)^2$ rows and columns of A_G form a lower triangular matrix as we wanted, and the Lemma follows. □

To complete the proof of Theorem 1.5, we apply Lemma 3.2 to our grid G and its cutset-edge incidence matrix A_G. We get that there exists an edge-costs function for G such that every mimicking network of G has at least $\text{rank}(A_G) = \Omega(k^2)$ edges and the theorem follows.

$\square$

4 Lower Bounds for Data Structures

We can extend the definition of a (deterministic) TC scheme to a randomized one by letting the two operations access a common source of random bits. (We do not assume the random bits are stored explicitly in M, even though it might be required in some implementations.) We then change the requirement from the query operation to be

$$\Pr[Q(S;M) = \text{mincut}_{G,c}(S,\bar{S})] \geq 2/3,$$

where the probability is taken over the data structure's random bits. Our lower bound in Theorem 1.7 holds also for randomized schemes, even those with shared randomness (that is not stored explicitly).

4.1 Proof for deterministic schemes

We now prove Theorem 1.7, which asserts that a terminal-cuts scheme requires $2^{\Omega(k)}$ words in the worst-case. Fix k and let (G,c) be the k-terminal bipartite graph constructed in Section 3.3. Recall that $l := \binom{k}{2k/3}$ is the number of subsets of terminals of size $2k/3$, each corresponding to a non-terminal in G. The number of vertices in G is $n := k + l = 2^{\Theta(k)}$, and size of a machine word is $O(\log n) = \Theta(k)$ bits. Assume towards contradiction there is a terminal-cuts scheme that can handle every k-terminal network using less than $l/100$ bits. For now, let us assume the scheme is deterministic.

Let $A_{G,c}$ be the cutset-edge incidence matrix of (G,c). By Lemma 3.6, $\text{rank}(A_{G,c}) \geq l$. Let us assume that the first l columns of $A_{G,c}$ are linearly independent (otherwise, we just reorder them), and let e_j denote the edge of G corresponding to the j-th column of $A_{G,c}$.

Let $\mathcal{W}$ denote the collection of 2^l edge-costs functions $w : E(G) \to \{0, \frac{1}{6k^2 l}\}$ satisfying that $w(e_j) = 0$ for all $j > l$. As in Section 3.3, every function $w \in \mathcal{W}$ defines a graph $(G, c+w)$, whose cutset-edge incidence matrix is denoted $A_{G,c+w}$. We can now apply Lemma 3.3, since $6k > \Delta_{G,c}$ and $|E(G)| = kl$, and obtain that for all $w \in \mathcal{W}$ the network $(G, c+w)$ has the same cutset-edge incidence matrix as (G,c), i.e. $A_{G,c} = A_{G,c+w}$. Using the above bound on the rank of $A_{G,c}$ we can deduce that for every two different functions $w \neq w' \in \mathcal{W}$, we have $A_{G,c} \cdot (\vec{c} + \vec{w}) \neq A_{G,c} \cdot (\vec{c} + \vec{w'})$, i.e. there exists $S \subset Q$ such that $\text{mincut}_{G,c+w}(S,\bar{S}) \neq \text{mincut}_{G,c+w'}(S,\bar{S})$.

Now, the assumed terminal-cuts scheme uses less than $l/100$ bits, and thus, by the pigeonhole principle, there must be $w \neq w' \in \mathcal{W}$, whose preprocessing results with the exact same memory image $M = P(G, c+w) = P(G, c+w')$. Consequently, for all queries $S \subset Q$, the scheme will report the same answer under inputs $(G, c+w)$ and $(G, c+w')$, which means that $\text{mincut}_{G,c+w}(S,\bar{S}) = \text{mincut}_{G,c+w'}(S,\bar{S})$ and is a contradiction.

Notice that the edge costs of the graphs $(G, c+w)$ for $w \in \mathcal{W}$ can be easily scaled so that they are all in the range $\{0, 1, \ldots, n^{O(1)}\}$. We conclude that a terminals-cut scheme for k terminals requires, in the worst case, storage of at least $\frac{l/100}{O(\log n)} \geq 2^{\Omega(k)}$ words. This proves Theorem 1.7 for deterministic schemes.

4.2 Proof for randomized schemes

The proof for randomized schemes follows the same outline, the main difference being that we replace the simple collision argument between $w \neq w'$, with well-known entropy (information) bounds. First, the data structure's success probability can be amplified to at least (say) $1 - \frac{1}{2^{10k}}$, by straightforward independent repetitions and using Chernoff bound, while increasing the storage requirement by a factor of $O(k)$. More formally, denote by Q' the TC scheme that runs the TC scheme Q independently $O(k)$ (say ck times for big enough constant $c > 0$) times, store all the $O(k)$ different values it get as an estimation for $mincut_{G,c}(S, \bar{S})$, and output as an answer the median value among them.

Lemma 4.1. $\Pr[Q'(S; M) = mincut_{G,c}(S, \bar{S})] \geq 1 - 2^{ck}$.

Proof. Fix a partition $Q = S \cap \bar{S})$. Define a random variable X_i for repetition i of the scheme such that $X_i = 1$ if $Q(S; M) = \text{mincut}_{G,c}(S, \bar{S})$, otherwise $X_i = 0$, and define $X = \sum_i X_i$. Thus,

$$\Pr[X_i = 1] = \Pr[Q(S; M) = \text{mincut}_{G,c}(S, \bar{S})] \geq 2/3$$

and

$$\mathbf{E}[X] = 1 \cdot \frac{2}{3}ck + 0 \cdot \frac{1}{3}ck \geq \frac{2}{3}ck.$$

Notice that

$$\Pr[Q'(S; M) \neq \text{mincut}_{G,c}(S, \bar{S})] \leq \Pr[X \leq \frac{1}{2}ck] \leq \Pr[X \leq (1-\frac{1}{4})\mathbf{E}[X]] \leq e^{-(1/4)^2\mathbf{E}[X]/2} \leq 2^{-10k}.$$

Where the first equality is due to the fact that if more than half of the ck values are equal to $\text{mincut}_{G,c}(S, \bar{S})$ then $Q'(S; M) = mincut_{G,c}(S, \bar{S})$. Using a Chernoff bound and by setting $c \geq 336$ we get that the two last inequalities, and the lemma follows.

□

So assume henceforth this high probability event does occur, and let us choose $w \in \mathcal{W}$ at random - which corresponds to choosing a random string of l bits. Using the data structure, one can retrieve with very high probability $(1-2^{-10k})$ the value $\text{mincut}_{G,c+w}(S, \bar{S}) = A_{G,c} \cdot (\vec{c} + \vec{w})$. Applying a union bound over all 2^k subsets $S \subset Q$ as follows,

$$\begin{aligned}
\Pr[\forall S,\ Q(S; M) = \text{mincut}_{G,c}(S, \bar{S})] &\geq 1 - \Pr[\exists S,\ Q(S; M) \neq \text{mincut}_{G,c}(S, \bar{S})] \\
&\geq 1 - \sum_S \Pr[Q(S; M) \neq \text{mincut}_{G,c}(S, \bar{S})] \\
&\geq 1 - 2^{-10k} \cdot 2^k \geq 0.9
\end{aligned}$$

with probability at least 0.9 one would retrieve correctly all these values. In this case, since the first l columns of $A_{G,c}$ yield an invertible matrix, we could actually recover the vector w itself (with probability at least 0.9). But since w is effectively a random string of l bits, it follows by standard entropy bounds that M must have at least $\Omega(l)$ bits.

More formally, denote by g the procedure described above that uses M to predict w with error probability at most $\epsilon = 0.1$, i.e. $\Pr[g(M) = w] \geq 1 - \epsilon = 0.9$, then we turn to state and prove the following lemma.

Lemma 4.2. *The size of the data structure is* $|M| \geq \Omega(l)$.

Proof. By Fano's inequality [CT06] $H(w|M) \leq H(\epsilon, 1-\epsilon) + \epsilon \log(2^l - 1) = \epsilon l + O(1)$. Now using mutual information we get the following

$$H(w) - H(w|M) = I(w; M) = H(M) - H(M|w) \leq H(M) \leq |M|$$

Since w is a random vector we know that $H(w) = l$. Thus,

$$|M| \geq H(w) - H(w|M) \geq l(1-\epsilon) - O(1) = 0.9l - O(1) \geq \Omega(l).$$

□

And the proof is completed just like for a deterministic scheme.

5 Concluding Remarks

We studied the problem of preserving minimum terminal cuts, while our main goal was to narrow the double exponential gap on the size of mimicking networks. We first handled the case of mimicking networks for general graphs, and then proceeded to work on mimicking networks for planar graphs. In both cases we narrowed the gap into only exponential one. In general graphs the improvement was in the lower bound, and in the graphs the improvement was in the upper bound on the size of the mimicking network. In this section we discuss and propose potential directions to make further progress on the mimicking network problem.

5.1 Upper Bounds

We can try to improve the upper bound, in order to narrow the exponential gap. For general graphs we wish to achieve an exponential upper bound, and for planar graphs we wish to achieve a polynomial upper bound in terms of the number of terminals.

General Graphs We can start to explore the impact of the terminal min-cuts on the structure of the graph using the cutset-edge incidence matrix (Definition 3.1) and the vector of all the terminal min-cuts. Given a vector of terminal min-cuts, it will be interesting to construct some graph that realizes the terminal min-cuts in the vector. Then we can ask for efficiency (in terms of size and running time) while constructing this graph. For $k < 6$ there are known algorithms that given such vector of terminal cuts, constructs a graph of size $O(k)$ [CSWZ00]. The problem arise when $k \geq 6$, and it is no longer possible to determine the relationship (bigger or smaller) between every terminal min-cut values of some subsets of terminals S and T with some common terminals (for example if $Q = \{q_1, \ldots, q_6\}$, then for $S = \{q_1, q_2, q_3\}$ we do not know for sure whether the min-cut that separates $\{q_1, q_2\}$ and $Q \setminus \{q_1, q_2\}$ is smaller then the min-cut that separates $\{q_1, q_3\}$ and $Q \setminus \{q_1, q_3\}$).

In addition, we could try to use the properties of the cutset-edge incidence matrix of the graph and its terminal min-cut vector in order to construct a mimicking network of smaller size as follows. Explore what can the linear dependency between the rows or the columns of the matrix teach us about the structure of the original graph, and how can we use this information in order to construct a smaller mimicking network.

A cutset-edges incidence matrix has $2^{k-1}-1$ rows (the number of different bipartitions of the terminals), and the number of columns (i.e. the number of edges in the graph) is significantly larger than 2^k (otherwise we do not need to compress it). Therefore the rank of the matrix is at most exponential in k, which suggests a lot of redundant information in the columns. Consider the following graph compression algorithm. Given a k terminal network (G, c), $B_{G,c}$ is a full-rank submatrix of $A_{G,c}$, constructed by selecting the maximum possible number of rows. Pad with 0 all the relevant places, and multiply both sides of the linear equations (mentioned in Definition 3.1) by $B_{G,c}^{-1}$ in order to retrieve the costs of the edges that corresponds to the independent columns of $B_{G,c}$ (the rest of the edges in the original graph will get cost 0). Call this new graph (G', c'). Since the rank of every cutset-edge incidence matrix is at most $2^{k-1}-1$ we will get at most $2^{k-1}-1$ edges with positive cost, and so the number of vertices (after removing edges with cost

0) will be at most exponential in k as well. The last question that need to be answered in order to prove that (G', c') is a mimicking network of (G, c) is that the terminal min-cuts in (G, c) that (G', c') preserves are also the minimum ones in (G', c').

There are two other different graph parameters that we can explore in order to attack the problem. One of them is independent sets of vertices. Try to simplify the structure of the general graph by considering all its independent sets, then explore the relations between them in terms of minimum terminal cuts (note that since we can always preserve all the edges between the terminals we can place them in the same independent set). Another way is to determine some local neighborhood around every terminal and around some non-terminals, then explore the connectivity between them (maybe in terms of minimum matching - in intention to find the specific edges that participate in the terminal min-cuts).

Planar Graphs In order to understand the structure of a planar graph better in terms of minimum terminal cuts we suggest to study the structure of the dual graph first as follows. Using Theorem 2.3 that deals with the duality between minimum cuts and circuits, consider minimum terminal circuits instead of cuts. Construct a plane graph such that all its 2^k different minimum terminal circuits are obtained by only $poly(k)$ different parts. Then turn to study the structure of its primal plane graph while looking for some special pattern in it.

Excluded Minors Having proved a new upper bound result for planar graphs, which is a minor-closed family, it is naturally to ask whether these results can be extend for any minor-closed graph family with the complete graph K_l as a forbidden minor.

One way to extend our proof of Theorem 1.3 (upper bound for planar graphs) is to amplify Lemma 2.2 in the following way. Instead of bounding the number of connected components by $3k$ we will try to bound it by $O(f(l)k)$, where f is some function of l. Then need to use the excluded minor properties in order to bound efficiently the number of connected component after removing all the terminal cuts from the graph.

Another way to improve the upper bound for minor-closed graph family with the complete graph K_l as a forbidden minor involves Hadwiger number and the chromatic number as follow. It is known that every graph G with Hadwiger number $l-1$ (i.e. K_{l-1} is the largest complete graph that is a minor of G) has a vertex with at most $O((l-1)\sqrt{\log(l-1)})$ incident edges [Kos84]. By applying a greedy coloring algorithm that removes this low-degree vertex, colors recursively the remaining graph, and then adds back the removed vertex and colors it, one can show that the chromatic number of G is at most $O((l-1)\sqrt{\log(l-1)})$. Thus, we can conclude that the graph can be partitioned into $O((l-1)\sqrt{\log(l-1)})$ independent sets of vertices in every graph that does not contain the complete graph K_l as a minor. As mentioned in the discussion above for general graphs, maybe the simplification of the graph structure to only a bounded number of independent sets can assist to construct a smaller size mimicking networks.

5.2 Lower Bounds

In order to further narrow the exponential gap, we can try to improve the exponential lower bound for general graphs (Theorem 1.4) to match the known doubly exponential upper bound $O(2^{2^k})$ of [HKNR98]. We may also try to improve the lower bound for planar graphs to match the exponential upper bound $O(k^2 2^{2k})$ (Theorem 1.3).

General Graphs The technique presented in Section 3.1 can be used only to prove a lower bound of at most $2^{k-1}-1$. This is due to the fact that the number of rows in every cutset-edge incidence matrix (Definition 3.1) is equal to $2^{k-1}-1$, which is the number of different bipartitions of the terminals. Thus the rank of that matrix is at most $2^{k-1}-1$, and so by Lemma 3.2 we can only prove that the size of every mimicking network is at least $\Omega(2^k)$.

Thereupon, we tried to study more carefully the structure of the graph. In particular, we wanted to construct a graph which achieves the upper bound in [HKNR98], i.e. a graph that all its terminal min-cuts create 2^{2^k} different equivalence class (see proof sketch of Theorem 1.2). We wondered whether some special structure between the non-terminal vertices can increase the number of equivalence classes. Thus, we started with the k-terminal graph constructed in Section 3.3 (lower bound for general graphs) which has only $2^{\Omega(k)}$ equivalence classes, and extended it (by adding non-terminals as a new independent set in the graph) to be a tripartite graph. Sadly, this tripartite graph did not help us to confirm or disprove our assumption and the question remains open.

Planar Graphs Since in the thesis we proved an upper bound that is roughly exponential in k (Theorem 1.3), we can hope to improve the lower bound to be (roughly) exponential in k. Thus, we can perhaps use Lemma 3.2 to close the gap almost completely. The only constraint we need to fulfil is to construct a planar graph, which yields a cutset-edges incidence matrix of rank r to prove a lower bound of $\sqrt{r}$. Notice that the mimicking network of a planar graph is not necessarily planar, and so we can not use the planarity properties on the mimicking network - the linear relation between the number of vertices and the number of edges etc..

Excluded Minors Again, similarly to the upper bound case, we inquire whether the lower bound result for planar graphs can be extended to any minor-closed graph family with the complete graph K_l as a forbidden minor or with the complete bipartite graph $K_{l,l}$, where the terminals in the minor are independent set, as a forbidden minor. We will discuss the second case.

Consider a bipartite graph with the complete bipartite graph $k_{l,l}$ as a forbidden minor, when all the terminals are an independent set and all the non-terminals form an independent set. The easiest case is when all the terminals are divided into disjoint sets of size at most l, all the non-terminals are divided to a disjoint sets of size at most l, and every set of terminals connects to a distinct set of non-terminals, which lead (using the known upper bound) to a mimicking network of size $O(k2^{2^l}/l)$.

The construction of the mimicking network begins to be more complicated when terminals connect to some non-terminals from different sets. In particular, consider the terminal min-cuts of $(S, \bar{S})$ and $(T, \bar{T})$, where S and T have some terminals in common, and these terminals connect to some common set of non-terminals. Maybe considering these cases can help to improve the lower bound.

5.3 Extensions of Mimicking Networks

Directed Graphs The upper bound of Hagerup, Katajainen, Nishimura, and Ragde [HKNR98] holds for both directed and undirected graphs. We studied only the case of undirected graphs; our lower bounds actually hold for directed graphs as well. It is an interesting question whether there is a significant difference between the maximum size of a mimicking network in the directed and undirected versions of the problem, either for general graphs or for some natural family of graphs.

A Generalization The following definition increases the number of cuts that must be preserved. Yet, it equivalent to the original mimicking network definition (Definition 1.1).

Definition 5.1. *Let (G, c) be a k-terminal network. A* generalized mimicking network *of (G, c) is a k-terminal network (G', c') with the same set of terminals Q, such that for all disjoint $S, T \subset Q$,* $\text{mincut}_{G',c'}(S, T) = \text{mincut}_{G,c}(S, T)$.

We turn to prove the equivalence of Definition 1.1 and the above one. Let (G, c) be a k-terminal network, let (G', c') be a mimicking network of (G, c) and let (G'', c'') be a generalized mimicking network of (G, c). Since Definition 1.1 is a special case of the above definition, its easy to verify that (G'', c'') is also a mimicking network of (G, c). In the following claim we prove the other direction.

Claim 5.2. *(G', c') is a generalized mimicking network of (G, c).*

Proof. Let S and T be a two disjoint subsets of terminals, i.e. $S, T \subset Q$ such that $S \cap T = \emptyset$. Every cut that separates S and T must also separate some subset of terminals $S \subseteq W$ and the rest of the terminals $T \subseteq Q \setminus W$. Thus, we can express the value of the minimum cut that separates S and T in G and G' as follows:

$$\text{mincut}_{G,c}(S, T) = \min_{W:\ S \subseteq W \subseteq Q \setminus T} \{\text{mincut}_{G,c}(W, Q \setminus W)\}$$

$$\text{mincut}_{G',c'}(S, T) = \min_{W:\ S \subseteq W \subseteq Q \setminus T} \{\text{mincut}_{G',c'}(W, Q \setminus W)\}$$

Since G' is a mimicking network of G, we know that for every $W \subseteq Q \setminus T$ the equality $\text{mincut}_{G,c}(W, Q \setminus W) = \text{mincut}_{G',c'}(W, Q \setminus W)$ holds. Thus,

$$\min_{W:\ S \subseteq W \subseteq Q \setminus T} \{\text{mincut}_{G,c}(W, Q \setminus W)\} = \min_{W:\ S \subseteq W \subseteq Q \setminus T} \{\text{mincut}_{G',c'}(W, Q \setminus W)\}$$

and the claim follows. □

Accordingly, all the results that were discussed and asserted in the thesis (the upper bounds and the lower bounds) hold for the more general definition as well.

Special Graphs Families Usually, graphs that belong to some family $\mathcal{F}$ have some special structure which is sometimes important to preserve, for example planar graphs, trees and bounded treewidth. Thus, the above definition restricts the mimicking network G' to be with a special (limited) structure.

Definition 5.3. *Let $\mathcal{F}$ be a family of graphs, and let $(G, c) \in \mathcal{F}$ be a k-terminal network. A* special mimicking network *of (G, c) is a k-terminal network (G', c') with the same set of terminals Q, such that $(G', c') \in \mathcal{F}$, and for all $S \subset Q$,* $\text{mincut}_{G',c'}(S, \bar{S}) = \text{mincut}_{G,c}(S, \bar{S})$.

For example, the mimicking network we construct for planar graph (Theorem 1.3) is a minor, and thus is itself planar. It is also known that every k-terminal network which is outerplanar has a mimicking network of size $O(k)$ which is also outerplanar [CSWZ00]. It is interesting to ask how many non-terminals we need to "add" to an optimal mimicking network of some graph in order to preserve the structure of the graph in addition to its terminal cuts. A different variation of this question is to study the tradeoff (the relation) between the size of the mimicking network to its structure similarity of the original graph. Notice that the answers to these two questions can differ from one graph family to another.

Terminal Vertex-Cuts Let (G, c) be a k-terminal network with a non-negative cost associated to each vertex, and assume the terminals Q is an independent set. A minimum vertex-cut of (G, c) that separates between a subset of terminals S and its complement $\bar{S}$ is a set of vertices $W \subseteq V(G)$ with the minimum total cost, such that the vertex deletion $G \setminus W$ disconnects the terminals S from the terminals $\bar{S}$. Denote its cost by $\text{vertex-cut}_{G,c}(S, \bar{S})$. We can now define the vertex-cut version of mimicking network.

Definition 5.4. *Let (G, c) be a k-terminal network. A* vertex-cut mimicking network *of (G, c) is a k-terminal network (G', c') with the same set of terminals Q, such that for all $S \subset Q$,* $\text{vertex-cut}_{G',c'}(S, \bar{S}) = \text{vertex-cut}_{G,c}(S, \bar{S})$.

Unfortunately, our upper bound proof for planar graphs can not be extended to the above definition of vertex-cut mimicking network. The first lemma in the proof (Lemma 2.1) bounds the number of connected components in the graph $G \setminus W$ by k, where W is a cutset edges. But in our case, W is a set of vertices. Thus, we can only argue that the number of connected components in $G \setminus W$ is bounded by k times the highest degree in G (take for example the star graph). Since the double exponential upper bound result for general graphs (Theorem 1.2) holds for that definition, it is interesting to ask whether our upper bound result for planar graphs (Theorem 1.3) holds as well, maybe with a different proof.

Approximate Mimicking Network The following definition relaxes the requirements from a mimicking network. It preserves the terminal cuts approximately instead of exactly, which in a sense add more freedom.

Definition 5.5. *Let (G, c) be a k-terminal network and let $\epsilon > 0$. An ϵ-*mimicking network *of (G, c) is a k-terminal network (G', c') with the same set of terminals Q, such that for all $S \subset Q$,* $\text{mincut}_{G,c}(S, \bar{S}) \leq \text{mincut}_{G',c'}(S, \bar{S}) \leq (1 + \epsilon)\,\text{mincut}_{G,c}(S, \bar{S})$.

It can be asked whether the additional freedom leads to a significant improvement in the size of the optimal mimicking network of any graph.

Multi-Commodity Flow All our results deal with minimum cut, which is equivalent to single-commodity maximum flow. The following definition preserves the multi-commodity terminal flow in a network.

Definition 5.6 (Multi-commodity Mimicking Network). *Let (G, c) be a k-terminal network. A* mimicking network *of (G, c) is a k-terminal network (G', c') with the same set of terminals Q, such that for all pairs $\{s_i, t_i\} \in \binom{Q}{2}$ and for all $d_i \in \mathbb{R}^+$, the multi-commodity flow $\{(s_i, t_i, d_i) : \ i = 1, \ldots, \binom{k}{2}\}$ is feasible in G if and only if it is feasible in G'.*

Since we are not aware of previous results on this problem, the question is whether we can bound the size of the optimal multi-commodity mimicking network by a function of k. Since the number of constraints is infinite, we suggest to reduce the number of choices for the demands d_i that we need to deal with in the following way. Convert the problem into a system of linear inequalities, where the feasible demands in the graph form a polytope, and the number of demands that we need to take into account are these at the extreme points. Then try to bound the number of extreme points by a function of k. A related problem is addressed by Chuzhoy [Chu12], as the multi-commodity flow preserved approximately in the compressed graph. She proves that every k-terminal network has a compressed k-terminal network of exponential size that depends on the total capacity of all edges incident on the terminals, and preserves the multi-commodity flow by a constant factor.

References

[AKPW95] N. Alon, R. M. Karp, D. Peleg, and D. West. A graph-theoretic game and its application to the k-server problem. *SIAM J. Comput.*, 24(1):78–100, February 1995.

[Bar96] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science*, pages 184–193. IEEE, 1996.

[BK96] A. A. Benczúr and D. R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. In *28th Annual ACM Symposium on Theory of Computing*, pages 47–55. ACM, 1996.

[BSS09] J. D. Batson, D. A. Spielman, and N. Srivastava. Twice-ramanujan sparsifiers. In *41st Annual ACM symposium on Theory of computing*, pages 255–262. ACM, 2009.

[CE06] D. Coppersmith and M. Elkin. Sparse sourcewise and pairwise distance preservers. *SIAM J. Discrete Math.*, 20:463–501, 2006.

[Chu12] J. Chuzhoy. On vertex sparsifiers with Steiner nodes. In *44th symposium on Theory of Computing*, pages 673–688. ACM, 2012.

[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.

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

[CT06] T. M. Cover and J. A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

[EGK+10] M. Englert, A. Gupta, R. Krauthgamer, H. Räcke, I. Talgam-Cohen, and K. Talwar. Vertex sparsifiers: New results from old techniques. In *13th International Workshop on Approximation, Randomization, and Combinatorial Optimization*, volume 6302 of *Lecture Notes in Computer Science*, pages 152–165. Springer, 2010.

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

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

[GH61] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9:551–570, 1961.

[HKNR98] T. Hagerup, J. Katajainen, N. Nishimura, and P. Ragde. Characterizing multi-terminal flow networks and computing flows in networks of small treewidth. *J. Comput. Syst. Sci.*, 57:366–375, 1998.

[HS85] D. S. Hochbaum and D. B. Shmoys. An $O(|V|^2)$ algorithm for the planar 3-cut problem. *SIAM J. Algebraic Discrete Methods*, 6(4):707–712, 1985.

[Kos84] A. V. Kostochka. Lower bound of the hadwiger number of graphs by their average degree. *Combinatorica*, 4(4):307–316, 1984.

[KRTV12] A. Khan, P. Raghavendra, P. Tetali, and L. A. Végh. On mimicking networks representing minimum terminal cuts. *CoRR*, abs/1207.6371, 2012.

[KZ12] R. Krauthgamer and T. Zondiner. Preserving terminal distances using minors. In *39th International Colloquium on Automata, Languages, and Programming*, volume 7391 of *Lecture Notes in Computer Science*, pages 594–605. Springer, 2012.

[MM10] K. Makarychev and Y. Makarychev. Metric extension operators, vertex sparsifiers and lipschitz extendability. In *51st Annual Symposium on Foundations of Computer Science*, pages 255–264. IEEE, 2010.

[Moi09] A. Moitra. Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size. In *50th Annual Symposium on Foundations of Computer Science*, FOCS, pages 3–12. IEEE, 2009.

[Nao90] M. Naor. Succinct representation of general unlabeled graphs. *Discrete Applied Mathematics*, 28(3):303–307, 1990.

[PS89] D. Peleg and A. A. Schäffer. Graph spanners. *J. Graph Theory*, 13(1):99–116, 1989.

[Rao87] S. Rao. Finding near optimal separators in planar graphs. In *28th Annual Symposium on Foundations of Computer Science*, pages 225–237. IEEE, 1987.

[Tur84] G. Turán. On the succinct representation of graphs. *Discrete Applied Mathematics*, 8(3):289 – 294, 1984.