Quasipolynomial multicut-mimicking networks and kernels for multiway cut problems*

MAGNUS WAHLSTRÖM, Royal Holloway, University of London, UK

We show the existence of an exact mimicking network of $k^{O(\log k)}$ edges for minimum multicuts over a set of terminals in an undirected graph, where k is the total capacity of the terminals, i.e., the sum of the degrees of the terminal vertices. Furthermore, using the best available approximation algorithm for Small Set Expansion, we show that a mimicking network of $k^{O(\log^3 k)}$ edges can be computed in randomized polynomial time. As a consequence, we show quasipolynomial kernels for several problems, including Edge Multiway Cut, Group Feedback Edge Set for an arbitrary group, and Edge Multicut parameterized by the solution size and the number of cut requests. The result combines the matroid-based irrelevant edge approach used in the kernel for s-Multiway Cut with a recursive decomposition and sparsification of the graph along sparse cuts. This is the first progress on the kernelization of Multiway Cut problems since the kernel for s-Multiway Cut for constant value of s (Kratsch and Wahlström, FOCS 2012).

CCS Concepts: • **Mathematics of computing → Extremal graph theory**; **Graph algorithms**; • **Theory of computation → Parameterized complexity and exact algorithms**.

Additional Key Words and Phrases: kernelization, sparsification, graph separation

1 INTRODUCTION

Graph separation questions are home to some of the most intriguing open questions in theoretical computer science. In approximation algorithms, the well-known *unique games conjecture* (UGC) has been central to the area for close to two decades, and is closely related to graph separation problems. Even more directly, the *small set expansion hypothesis*, proposed by Raghavendra and Steurer [37], roughly states that it is NP-hard to approximate the Small Set Expansion problem (SSE) up to a constant factor, where SSE is the problem of finding a small-sized set in a graph with minimum expansion. (More precise statements are given below.) For the general case, despite significant research, the best polynomial-time result is an $O(\log n)$-approximation due to Räcke [36], but stronger results are known for special cases. In particular, if the size bound on the desired set S is $|S| \leq s$, then Bansal et al. [3] show an algorithm with an approximation ratio of $O(\log n/\sqrt{\log s})$.

Another interesting notion from parameterized complexity is *kernelization*. Informally, a kernelization algorithm is a procedure that takes an input of a parameterized, usually NP-hard problem and *reduces* it in polynomial time to an equivalent instance of size bounded in the parameter, e.g., by discarding irrelevant parts of the input or transforming some part of the input into a smaller object with equivalent behaviour. For example, the seminal Nemhauser-Trotter theorem on the half-integrality of Vertex Cover [33] implies that an instance of Vertex Cover can be reduced to have at most $2k$ vertices, where k is the bound given on the solution size. On the flip side, Fortnow and Santhanam [12] and Bodlaender et al. [4] gave a framework to exclude the existence of a kernel of any polynomial size, under a standard complexity-theoretic conjecture. An extensive collection of upper and lower bounds for kernelization exists (see, e.g., the recent book of Fomin et al. [11]), but a handful of central "hard questions" remain unanswered. One of the most notorious is Multiway Cut.

*A preliminary version of this paper with weaker results was presented at ICALP 2020 [42]. Compared to that version, the current paper has a much simpler correctness argument for the marking procedure, as well as adding the constructive version of the result. We have also observed an error in [42] regarding the claimed 0-Extension application; see remark at the end of Section 3.

Author's address: Magnus Wahlström, Royal Holloway, University of London, Department of Computer Science, Egham Hill, Egham, UK, TW20 0EX, Magnus.Wahlstrom@rhul.ac.uk.

Let $G = (V, E)$ be a graph and $T \subseteq V$ a set of terminals in G. An *(edge) multiway cut* for T in G is a set of edges $X \subseteq E$ such that no two terminals are connected in $G - X$, and Multiway Cut is the problem of finding a multiway cut of at most k edges, given a parameter k. The problem is FPT [30] and NP-hard for $|T| \geq 3$ [8]. Using methods from matroid theory, Kratsch and Wahlström [20] were able to show that if $|T| \leq s$, then Multiway Cut has a kernel with $O(k^{s+1})$ vertices, hence the problem has a polynomial kernel for every constant s. However, if $|T|$ is unbounded, the only known size bound for a kernel is $2^{O(k)}$, following from the FPT algorithm [30], and the question of whether Multiway Cut has a polynomial kernel in the general case is completely open.

We make progress on this question by showing that Multiway Cut and several related problems have *quasipolynomial* kernels, i.e., kernels of size $k^{\log^{O(1)} k}$. Furthermore, the degree in the exponent depends on the best available approximation algorithm for Small Set Expansion (see Theorem 1.1 for exact dependence). With the current state of the art, we are able to show kernels of size $k^{O(\log^3 k)}$; and if the small set expansion problem has a constant-factor approximation, the result would be kernels of size $k^{O(\log k)}$.

The result goes via showing the existence of a kind of *mimicking network* for the problem; or more generally, a network of quasipolynomial size mimicking the behaviour of (G, T) for all multicut instances over T. We review these notions next.

1.1 Mimicking networks and multiway cut sparsifiers

Although kernelization is most commonly described in terms of polynomial-time preprocessing as above, there is also a clear connection with *succinct information representation*. For example, consider a graph $G = (V, E)$ with a set of p terminals $T \subseteq V$. The pair (G, T) is referred to as a *terminal network*. A *mimicking network* for (G, T) is a graph $G' = (V', E')$ with $T \subseteq V'$ such that for any sets $A, B \subseteq T$, the min-cut between A and B in G and G' have the same value. A mimicking network of size bounded in p always exists [14], but the size of G' can be significant. The initial upper bound given was 2^{2^p} [14], only slightly improved since [17], and there is an exponential lower bound [22]. Better bounds are known for special graph classes, but even for planar graphs the best possible general bound has $2^{\Theta(p)}$ vertices [16, 22] (see also recent improvements by Krauthgamer and Rika [21]).

A related notion is *cut sparsifiers*, which solve the same task up to some approximation factor $q \geq 1$ [24, 32], typically $q = \omega(1)$ in the general case. We focus on mimicking networks; see Krauthgamer and Rika [21] for an overview of cut sparsifiers. However, we note that for general graphs, constant-factor cut sparsifiers are not known with any size bound smaller than the size guarantee for an exact mimicking network.

However, if we include the capacity of the set of terminals in the bound (and if edges have integer capacity), then significantly stronger results are possible. Chuzhoy [5] showed that if the total capacity of T is $\mathrm{cap}_G(T) = \sum_{t \in T} d(t) = k$, then there exists an $O(1)$-approximate cut sparsifier of size $O(k^3)$. Kratsch and Wahlström [20] sharpened this to an exact mimicking network with $O(k^3)$ edges, which furthermore can be computed in randomized polynomial time. This is particularly remarkable given that the network has to replicate the exact cut-value for exponentially many pairs (A, B). The network can be constructed via contractions in G.[1] This built on an earlier result that used linear representations of matroids to encode the sizes of all (A, B)-min cuts into an object using $O(k^3 \log^{O(1)} k)$ bits of space [19], although this earlier version did not produce an explicit graph, i.e., not a mimicking network.

These results had significant consequences for kernelization. The succinct representation in [19] was used to produce a (randomized) polynomial kernel for the Odd Cycle Transversal problem,

[1]The results of [20] are phrased in terms of vertex cuts, but the above follows easily from [20].

thereby solving a notorious open problem in parameterized complexity [19]; and the mimicking network of [20] brought further (randomized) polynomial kernels for a range of problems, in particular including Almost 2-SAT, i.e., the problem of satisfying all but at most k clauses of a given 2-CNF formula.

Similar methods are relevant for the question of separating a set of terminals into more than two parts. Let (G, T) be a terminal network, and let $\mathcal{T} = T_1 \cup \ldots \cup T_s$ be a partition of T. A *multiway cut for* $\mathcal{T}$ is a set of edges $X \subseteq E(G)$ such that $G - X$ contains no path between any pair of terminals $t \in T_i$ and $t' \in T_j$, $i \neq j$. Let us tentatively define a *multicut-mimicking network* for (G, T) as a terminal network (G', T) where $T \subseteq V(G')$ and for every partition $\mathcal{T} = T_1 \cup \ldots \cup T_s$ of T, the size of a minimum multiway cut for $\mathcal{T}$ is identical in G and G'. (The term *multicut-mimicking*, as opposed to *multiway cut-mimicking*, is justified; see Prop. 2.2.) The minimum size of a multicut-mimicking network, in terms of $k = \mathrm{cap}_G(T)$, appears to lie at the core of the difficulty of the question of a polynomial kernelization of Multiway Cut. The kernel for s-Multiway Cut mentioned above builds on the computation of a mimicking network of size $O(k^{s+1})$ for partitions of T into at most s parts [20]. The kernel for s-Multiway Cut then essentially follows from considering the partition $\mathcal{T} = \{t_1\} \cup \ldots \cup \{t_s\}$ of a set T of $|T| = s$ terminals, along with known reduction rules bounding $\mathrm{cap}_G(T)$. We are not aware of any non-trivial lower bounds on the size of a multicut-mimicking network in terms of k; it seems completely consistent with known bounds that every terminal network (G, T) could have a multicut-mimicking network of size $\mathrm{poly}(k)$, even for partitions into an unbounded number of sets.

In this paper, we show that any terminal network (G, T) with $\mathrm{cap}_G(T) = k$ admits a multicut-mimicking network (G', T) where $|V(G')| = k^{O(\log k)}$; and furthermore, a network with $|V(G')| = k^{O(\log^{O(1)} k)}$ can be computed in randomized polynomial time, using a sufficiently good approximation algorithm for a graph separation problem similar to Small Set Expansion (SSE). We also see a tradeoff between the quality of the approximation algorithm and the size of (G', T). Using the algorithm of Bansal et al. [3], we achieve $|V(G')| = k^{O(\log^3 k)}$; and if the small set expansion hypothesis were false and SSE had a constant-factor approximation algorithm, then the bound $|V(G')| = k^{O(\log k)}$ would be achievable in polynomial time. We leave open the questions of whether there always exists a multicut-mimicking network of size $k^{O(1)}$, as well as the question of whether a network of size $k^{O(\log k)}$ can be computed through means other than a constant-factor approximation for SSE.

As a side note, we note that a 2-approximate "multicut sparsifier" of size $O(k^3)$ can be computed efficiently using known methods. Specifically, we observe that the above-mentioned mimicking network of $O(k^3)$ edges for standard terminal cuts in a terminal network (G, T) with $\mathrm{cap}(T) = k$ [20] implies a terminal network (G', T) where $\mathrm{cap}(T) = k$, $|E(G')| = O(k^3)$, and where for every partition $\mathcal{T}$ of T the costs of minimum multiway cuts for $\mathcal{T}$ in (G', T) and (G, T) differ by at most a factor of 2 (see Lemma 2.4).

Flow sparsifiers. Finally, similarly to cut sparsifiers, there is a notion of a *flow sparsifier* of a terminal network (G, T). Here the goal is to approximately preserve the minimum congestion for any multicommodity flow on (G, T). Chuzhoy [5] showed flow sparsifiers with quality $O(1)$ and with $k^{O(\log \log k)}$ vertices, where k is the total terminal capacity; for further results on achievable bounds for flow sparsifiers, see [2, 9]. However, the notion is incomparable to multicut-mimicking networks, because even an exact flow sparsifier would be subject to the corresponding multicommodity flow-multicut approximation gap, which is $\Theta(\log k)$ in the worst case [13].

Further related results. The general approach of decomposing a graph along sparse cuts is well established; cf. Räcke [35] and follow-up work.

The Small Set Expansion (SSE) problem is defined as follows. Let $G = (V, E)$ be a graph and $S \subseteq V$ a set of vertices. The *edge expansion* of S is

$$\Phi(S) := \frac{|E(S, V \setminus S)|}{|S|},$$

where $E(A, B)$ denotes the set of edges with one endpoint in A and one in B. For a real number $\rho \in (0, 1/2]$, one also defines the *small set expansion*

$$\Phi_\rho(G) := \min_{S \subseteq V, |S| = \rho n} \Phi(S).$$

In particular, for a value $s \in [n/2]$, $\Phi_{s/n}(G)$ denotes the worst (i.e., minimum) expansion among subsets of G of size at most s. The Small Set Expansion problem is to compute the value of $\Phi_s(G)$. The *Small Set Expansion Hypothesis* (SSEH) states that for every constant $\eta > 0$ there exists a value $\epsilon > 0$ such that it is NP-hard to distinguish between the cases that $\Phi_\epsilon(G) \leq \eta$ and $\Phi_\epsilon(G) \geq 1 - \eta$. SSEH was defined by Raghavendra and Steurer [37], who also showed that SSEH implies UGC [37]; in fact it is equivalent to a stronger version of UGC [38]. Like UGC, SSEH has also been used as the foundation for hardness of approximation bounds not known to follow by other methods [29, 38].

For further applications of matroid tools to kernelization, see Hols and Kratsch [15], Kratsch [18], and Reidl and Wahlström [39].

1.2 Our results

We show the following.

Theorem 1.1. *Let A be an approximation algorithm for Small Set Expansion for instances with integral, strictly positive capacities with an approximation ratio of $\alpha(n, k)$, where k is the number of edges cut in the optimal solution. Let (G, T) be a terminal network with $\mathrm{cap}_G(T) = k$. Then there is a set $Z \subseteq E(G)$ with $|Z| = k^{O(\alpha(n,k) \log k)}$ such that for every partition $\mathcal{T} = T_1 \cup \ldots \cup T_s$ of T, there is a minimum multiway cut X for $\mathcal{T}$ such that $X \subseteq Z$. Furthermore, Z can be computed in randomized polynomial time using calls to A.*

Note that this result, and the corollaries below, are given in terms of the number of edges of the output, where the more common size measure for a sparsifier or mimicking network (or, indeed, for a polynomial kernel) is the number of vertices of the output graph. However, since we generally leave the exponent open up to a constant-factor term, this is not a significant difference.

Unfortunately, the best known ratio for SSE which can be stated purely in terms of k and n is $\alpha(n, k) = O(\log n)$ [36], and plugging this into the above formula yields a vacuous result. However, by a more careful analysis we are able to show a better bound using the algorithm of Bansal et al. [3]. In summary, we get the following.

Corollary 1.2. *Let (G, T) be a terminal network with $\mathrm{cap}_G(T) = k$. The following holds.*

(1) *There is a multicut-mimicking network for (G, T) with $k^{O(\log k)}$ edges.*
(2) *A multicut-mimicking network for (G, T) with $k^{O(\log^3 k)}$ edges can be computed in randomized polynomial time.*

The latter implies several breakthrough results in kernelization, as follows. We refer to previous kernelization work [20] for the necessary definitions.

Corollary 1.3. *The following problems have randomized quasipolynomial kernels.*

(1) *Edge Multiway Cut parameterized by solution size.*
(2) *Edge Multicut parameterized by the solution size and the number of cut requests.*
(3) *Group Feedback Edge Set parameterized by solution size, for any group.*
(4) *Subset Feedback Edge Set with undeletable edges, parameterized by solution size.*

2 PRELIMINARIES

A *parameterized problem* is a decision problem where inputs are given as pairs (X, k), where k is the *parameter.* A *polynomial kernelization* is a polynomial-time procedure that maps an instance (X, k) to an instance (X', k') where (X, k) is positive if and only if (X', k') is positive, and $|X'|, k' \leq g(k)$ for some function $g(k)$ referred to as the *size* of the kernel. A problem has a *polynomial kernel* if it has a kernel where $g(k) = k^{O(1)}$. We extend this to discuss *quasipolynomial kernels*, which is the case that $g(k) = k^{\log^{O(1)} k}$. For more on parameterized complexity and kernelization, see [6, 11].

For a graph $G = (V, E)$ and sets $A, B \subseteq V$, we let $E_G(A, B) = \{uv \in E \mid u \in A, v \in B\}$. As shorthand for $S \subseteq V$ we also write $E(S) = E(S, S)$, $\partial_G(S) = E_G(S, V \setminus S)$, and $\delta_G(S) = |\partial_G(S)|$. For a vertex $v \in V$, $d_G(v)$ is the degree of v in G, also written $d(v)$ when the graph G is understood from context. The *total capacity* of a set of vertices S in a graph G is

$$\mathrm{cap}_G(S) := \sum_{v \in S} d(v).$$

In all cases, we may omit the index G if understood from context.

For a vertex set $S \subseteq V$ in a graph $G = (V, E)$, we let $N_G[S] = S \cup \{v \in V \mid \exists uv \in E, u \in S\}$ denote the closed neighbourhood of S. We write $N[S]$ in place of $N_G[S]$ when G is understood from context.

2.1 Multicut-mimicking networks

Let $G = (V, E)$ be a graph and $T \subseteq V$ a set of terminals with $\mathrm{cap}(T) = k$. An *edge multiway cut* for T in G is a set of edges $X \subseteq E$ such that no two vertices in T are connected in $G - X$. More generally, let $\mathcal{T} = \{T_1, \ldots, T_r\}$ be a partition of T. Then an *edge multiway cut for $\mathcal{T}$ in G* is a set of edges $X \subseteq E$ such that in $G - X$ every connected component contains terminals from at most one part of $\mathcal{T}$. Hence a multiway cut for (G, T) is equivalent to a multiway cut for $(G, \{\{t\} \mid t \in T\})$. Further, let $R \subseteq \binom{T}{2}$ be a set of pairs over T, referred to as *cut requests.* A *multicut for R in G* is a set of edges $X \subseteq E$ such that every connected component in $G - X$ contains at most one member of every pair $\{u, v\} \in R$. A *minimum multicut for R in G* is a multicut for R in G of minimum cardinality. Similarly, a *minimum multiway cut for $\mathcal{T}$ in G* is a multiway cut for $\mathcal{T}$ in G of minimum cardinality. For the rest of the paper, we let all cuts implicitly be edge cuts, unless otherwise specified, hence we generally refer simply to multiway cuts.

We now come to the central definition of the paper.

Definition 2.1 (Multicut-mimicking network). Let (G, T) be a terminal network, i.e., $G = (V, E)$ is a graph and $T \subseteq V$ is a set of terminals in G. A *multicut-mimicking network* for T in G is a graph $G' = (V', E')$ such that $T \subseteq V'$ and such that for every set of cut requests $R \subseteq \binom{T}{2}$, the size of a minimum multicut for R is equal in G and in G'.

We observe that preserving this is equivalent to preserving the sizes of minimum multiway cuts over all partitions of T.

Proposition 2.2. *A graph G' with $T \subseteq V(G')$ is a multicut-mimicking network for T in G if and only if, for every partition $\mathcal{T}$ of T, the size of a minimum multiway cut for $\mathcal{T}$ is equal in G and in G'.*

Proof. It is clear that the condition is necessary, since for any partition $\mathcal{T}$ of T we could form the set R of all pairs over T which lie in distinct parts of $\mathcal{T}$, and a multicut for R is then necessarily a multiway cut for $\mathcal{T}$. To see that the condition is also sufficient, consider an arbitrary set of cut requests $R \subseteq \binom{T}{2}$ and let X be a minimum multicut for (G, R). Let $\mathcal{T}$ be the partition of T in $G - X$ according to connected components. Then X is a multiway cut for $\mathcal{T}$, and any multiway cut for

$\mathcal{T}$ is also a multicut for R. Hence the size of a minimum multicut for R is precisely the size of a minimum multiway cut for $\mathcal{T}$. □

We also consider a slightly sharper notion.

Definition 2.3 (Multicut-covering set). Let (G, T) be a terminal network. A *multicut-covering set* for (G, T) is a set $Z \subseteq E(G)$ such that for every set of cut requests $R \subseteq \binom{T}{2}$, there is a minimum multicut X for R in G such that $X \subseteq Z$.

Note that a multicut-covering set Z is essentially equivalent to a multicut-mimicking network formed by contraction (contracting all edges of $E(G) \setminus Z$). Our main result in this paper is the existence of a multicut-covering set of size quasipolynomial in $k = \mathrm{cap}(T)$ in any undirected graph G. Furthermore, such a set can be computed in polynomial time, subject to the existence of certain approximation algorithms that we will make precise later in this section.

A multicut-covering set is a generalization of a *cut-covering set*, used in previous work [20]. Formally, a cut-covering set for (G, T) is a set $Z \subseteq E(G)$ such that for any partition $T = A \cup B$ there is an (A, B)-min cut X in G with $X \subseteq Z$. By previous work, if (G, T) is a terminal network with $\mathrm{cap}(T) = k$, then a cut-covering set of $O(k^3)$ edges can be computed in randomized polynomial time [20]. We observe that this gives us a 2-approximate multicut-covering set.

Lemma 2.4. *Let (G, T) be a terminal network and let Z be a cut-covering set for edge cuts over T. Then Z is a 2-approximate multicut-covering set.*

Proof. Let $\mathcal{T} = T_1 \cup \ldots \cup T_s$ be a partition of T and let X be a minimum multiway cut for $\mathcal{T}$. For $i \in [s]$, let $\lambda_i = \lambda(T_i, T \setminus T_i)$ be the size of an isolating min-cut for T_i in G. Then Z contains a $(T_i, T \setminus T_i)$-cut of cardinality λ_i for every $i \in [s]$, and by taking their union we get a solution $X' \subseteq Z$ with $|X'| \le \sum_{i=1}^{s} \lambda_i$.

It is known [40, Cor. 73.2e] that there exists a half-integral multiflow in G for $\mathcal{T}$ of value $\frac{1}{2} \sum_{i=1}^{s} \lambda_i \ge |X'|/2$. Hence $|X| \ge |X'|/2$, and X' is a 2-approximate multiway cut for $\mathcal{T}$. By the argument of Prop. 2.2, X' is also a 2-approximate multicut-mimicking network for (G, T). □

Thus, in randomized polynomial time we can compute a 2-approximate multicut-covering set for a terminal network (G, T) with $O(\mathrm{cap}(T)^3)$ edges.

2.2 Graph separation algorithms

The central technical approximation assumption needed in this paper is the following. For a graph G with a set of terminals T, define the *T-capacity of S in G* as

$$\mathrm{cap}_T(S) = \mathrm{cap}_G(T \cap S) + \delta_G(S).$$

For the purposes of the paper, we may consider a graph "dense" if there are no sets S with $|S| \le |V(G)|/2$ where $\mathrm{cap}_T(S)$ is very small compared to $|S|$. More precisely, we work with the following definition of a form of approximate denseness guarantee.

Definition 2.5 ((α, c)-dense). Let α and c be constants and let (G, T) be a terminal network. We say that (G, T) is *(α, c)-dense* if for every set $S \subseteq V(G)$ with $0 < |S| \le |V(G)|/2$ and $N[S] \neq V(G)$ we have $\mathrm{cap}_T(S) \ge |S|^{1/c}/\alpha$. More generally, let $\alpha \colon \mathbb{N} \to \mathbb{N}$ be a function. In this case, (G, T) is (α, c)-dense if for every set S as above we have $\mathrm{cap}_T(S) \ge |S|^{1/c}/\alpha(|S|)$.

We define the following notion.[2]

[2]For the main results of the paper, it suffices to assume a specialised version that focuses on cuts that cut through the terminal set. The details were worked out in the preliminary version of this paper [42], under the name *sublogarithmic terminal expansion tester*. We use here a simplified definition that suffices for our results.

Definition 2.6 (Quasipolynomial expansion tester). Let (G,T) be a terminal network with $\mathrm{cap}_G(T) = k$. A *quasipolynomial expansion tester (with approximation ratio α)* is a (possibly randomized) algorithm that, given as input (G,T) and an integer $c \in \mathbb{N}$, with $c = \Omega(\log k)$, does one of the following.

(1) Either returns a set $S \subset V$ such that $N[S] \neq V(G)$ and $\mathrm{cap}_T(S) < |S|^{1/c}$,
(2) or guarantees that (G,T) is (α,c)-dense.

More generally, we allow $\alpha\colon \mathbb{N} \to \mathbb{N}$ to be a function depending on $|S|$ in addition to n and k, in which case the alternate definition of (α,c)-denseness applies.

We note that quasipolynomial expansion testers follow from approximation algorithms for Small Set Expansion. Indeed, this should be no great surprise, as the problem definitions are almost identical, except for the parameter c.

Lemma 2.7. *Assume that Small Set Expansion has a bicriteria approximation algorithm that on input (G,ρ) returns a set S with $|S| \leq \beta \cdot \rho n$ and $\Phi(S) \leq \alpha(\rho n) \cdot \Phi_\rho$, for some $\alpha,\beta \geq 1$ which may depend on n, $k = \Phi_\rho \rho n$, and ρ. Also assume $c = \Omega(\log k)$ and $\alpha \leq O(\log n)$. Then there is a quasipolynomial expansion tester with an approximation function $\alpha'(|S|) = \Theta(\alpha(|S|)\beta)$.*

Proof. Let a terminal network (G,T) and parameter $c \in \mathbb{N}$ be given, $c = \Omega(\log k)$. Let $\alpha'(s) = 2\alpha(s)\beta$. We need to show that if (G,T) is not (α',c)-dense, then we can find a set $S \subset V$ with $N[S] \neq V$ such that $\mathrm{cap}_T(S) < |S|^{1/c}$. Hence assume that (G,T) is not (α',c)-dense, and let $S \subset V$ be a set witnessing this, i.e., $0 < |S| \leq |V(G)|/2$, $N[S] \neq V(G)$, and $\mathrm{cap}_T(S) < |S|^{1/c}/\alpha'(|S|)$. For shorthand write $\alpha' = \alpha'(|S|)$. We argue that the set $S \setminus T$ is also a legal return value for the algorithm. Note

$$\mathrm{cap}_T(S \setminus T) = \delta(S \setminus T) \leq \delta(S) + \mathrm{cap}_G(T \cap S) = \mathrm{cap}_T(S).$$

We also have $|S| > (\alpha' \mathrm{cap}_T(S))^c \geq (\alpha' \mathrm{cap}_T(S \setminus T)^c$. Now, recall that Minimum Bisection is FPT parameterized by the solution value (i.e., the number of edges cut by an optimal solution), with the fastest FPT algorithm running in time $O^*(2^{O(p \log p)})$ for parameter p [7]. Hence we can in polynomial time check for a bisection with $p = O(\log n/\log\log n)$ edges, and by replacing a vertex with a suitably large clique we can also check for a set S' of cardinality s with $\delta(S') \leq p$. Hence in the remaining case we assume $\mathrm{cap}_T(S) \geq \delta(S) \geq \Omega(\log n/\log\log n)$. Furthermore, by assumption $c = \Omega(\log k)$. Hence

$$|S| \geq (\alpha' \mathrm{cap}_T(S))^c \geq (\log n/\log\log n)^{\log k} = k^{\Omega(\log\log n)},$$

and the difference in size between $|S|$ and $|S \setminus T| \geq |S| - k$ is negligible. Hence

$$\Phi(S \setminus T) = \frac{\delta(S \setminus T)}{|S \setminus T|} \leq \frac{\mathrm{cap}_T(S)}{(1-o(1))|S|} < (1+o(1))(1/\alpha')|S|^{1/c-1}.$$

Now attach a large clique to every terminal in G, say of size $\beta|S|+1$, forming a graph G', and call an approximation algorithm for Small Set Expansion with a parameter of $\rho = |S|/|V(G')|$. Assume that the algorithm returns a set $S' \subseteq V(G')$. Then $|S'| \leq \beta|S|$, hence $S' \cap T = \emptyset$ and $\mathrm{cap}_T(S') = \delta(S')$. Furthermore $\Phi(S') \leq \alpha''\Phi(S \setminus T)$ where α'' is the approximation guarantee of the algorithm on input (G',ρ). Here, the only relevant difference between α'' and α lies in the difference between $|V(G)| = n$ and $|V(G'| \leq n + k(\beta|S|+1) \leq O(k\beta n)$. But since α by assumption depends on n as $O(\log n)$ or slower, this difference is a lower-order term. Then

$$\mathrm{cap}_T(S') = \delta(S') = |S'|\Phi(S') \leq |S'|\alpha\cdot\Phi(S\setminus T) < \frac{1+o(1)}{2\beta}|S'|\cdot|S|^{1/c-1}. = \frac{1+o(1)}{2\beta}(|S'|/|S|)^{1-1/c}|S'|^{1/c}.$$

Now, since $|S'|/|S| \leq \beta$ we get $\mathrm{cap}_T(S') < (1/2)(1+o(1))|S'|^{1/c}$, and S' is a valid return value. By repeating the above for all target sizes $|S| = \rho|V(G')|$ from 1 to $|V(G)|$, we can be sure to identify such a set S' if one exists. □

Recall that our main result (Theorem 1.1) states roughly that if Small Set Expansion has an approximation ratio of $\alpha(n,k)$, then we can find a multicut-covering set with $k^{O(\alpha(n,k)\log k)}$ edges. The strongest general-case approximation (stated purely in terms of n) is an $O(\log n)$-approximation due to Räcke [36]. Unfortunately, this is not a useful bound for us, since $k^{\log n \log k} = n^{\log^2 k}$ is a trivial size guarantee. However, Bansal et al. [3] gave a bicriteria algorithm with an approximation ratio of $O(\sqrt{\log n \log(1/\rho)})$, giving $\alpha(|S|) = O(\log n/\sqrt{\log |S|})$. In Section 3.3, we show that this bound is (just barely) strong enough for a constructive result, yielding a kernel with $k^{O(\log^3 k)}$ edges. Stronger bounds, e.g., $\alpha = O(\log k)$ or better, would naturally give improved results.

3 MULTICUT-COVERING SETS

We now present the main result of the paper, namely the existence of quasipolynomial multicut-mimicking networks for terminal networks (G,T), and a method for computing them in randomized polynomial time using an appropriate quasipolynomial expansion tester.

At a high level, the process works through recursive decomposition of the graph G across very sparse cuts, treating each piece $G[S]$ of the recursion as a new instance of multicut-covering set computation, where the edges of $\partial(S)$ are considered as additional terminals. The process repeatedly finds a single edge $e \in E(G)$ with a guarantee that for every set of cut requests $R \subseteq \binom{T}{2}$ there is a minimum multicut X for R in G such that $e \notin X$. We may then contract the edge e and repeat the process. Thus the end product is a multicut-mimicking network, and the edges that survive until the end of the process form a multicut-covering set.

In somewhat more detail, the process uses a variant of the representative sets approach, which was previously used in the kernel for s-Multiway Cut [20]. Refer to an edge e as *essential for* R, for some $R \subseteq \binom{T}{2}$, if every minimum multicut for R in G contains e, and *essential for* (G,T) if it is essential for R for some $R \subseteq \binom{T}{2}$. We use a representative sets approach to return a set of at most k^c edges which is guaranteed to contain every essential edge, *if* (G,T) is already (α,c)-dense, for an appropriate value $c = \Omega(\alpha \log k)$. On the other hand, if (G,T) is not (α,c)-dense, then (by careful choice of parameters) we can identify a cut through G which is sufficiently sparse that we can reduce the size of one side of this cut via a recursive call. This gives a tradeoff between the size of the resulting multicut-covering set and the denseness-guarantee we may assume through the approximation algorithm. When α is constant (or, more precisely, independent of $|S|$) then this analysis gives a quite simple bound for an algorithm that computes a multicut-covering set of k^c edges.

Unfortunately, the best bound on α independent of $|S|$ is just $\alpha = O(\log n)$, in which case the bound k^c with $c = \Omega(\log n)$ is vacuous. We therefore also perform a more careful analysis using the Small Set Expansion- approximation algorithm of Bansal et al. [3], and show that it allows us to compute a multicut-covering set of $k^{O(\log^3 k)}$ edges in polynomial time.

3.1 Recursive replacement

We now present the recursive decomposition step in detail. Let (G,T) be a terminal network with $\mathrm{cap}_G(T) = k$. For a set $S \subseteq V$, we define the graph

$$G_S = G[N_G[S]] - E(N_G(S)),$$

i.e., G_S equals the graph $G[S]$ with the edges of $\partial(S)$ added back in. We also denote

$$T(S) = (T \cap S) \cup N_G(S)$$

as the *terminals of S*. Under these definitions, the *T-capacity of S in G* has two equivalent definitions as

$$\mathrm{cap}_T(S) = \mathrm{cap}_{G_S}(T(S)) = \mathrm{cap}_G(T \cap S) + \delta_G(S).$$

The *recursive instance at S* consists of the terminal network $(G_S, T(S))$. This is the basis of our recursive replacement procedure. Indeed, we show the following. Note that we consider $E(G_S) \subseteq E(G)$ in the following.

Lemma 3.1. *Let $(G_S, T(S))$ be the recursive instance at S for some $S \subseteq V(G)$. Let Z_S be a multicut-covering set for $(G_S, T(S))$ and let $e \in E(G_S) \setminus Z_S$. Then e is not essential for (G, T).*

Proof. By Prop. 2.2, it is sufficient to consider partitions $\mathcal{T}$ of T and minimum multiway cuts X for $\mathcal{T}$. Let $\mathcal{T}$ be some partition of T, and let X be a minimum multiway cut for $\mathcal{T}$ in G. Let $\mathcal{T}'$ be the partition of $T(S)$ induced by the connected components of $G - X$, and let $X_S = X \cap E(G_S)$. Then X_S is a multiway cut for $\mathcal{T}'$ in G_S. Indeed, any path P in $G_S - X_S$ between distinct parts of $\mathcal{T}'$ also exists in $G - X$. If $\mathcal{T}'$ consists of a single part, then we have $X_S = \emptyset$, as otherwise either X contains an edge uv whose both endpoints lie in the same connected component of $G - X$, or $G - X$ contains a connected component with no terminals, both of which contradict that X is of minimum cardinality. Otherwise, by assumption there is a minimum multiway cut X'_S for $\mathcal{T}'$ in G_S such that $e \notin X'_S$. We claim that $X' := (X \setminus X_S) \cup X'_S$ is a minimum multiway cut for $\mathcal{T}$ in G. Note that $|X'| \leq |X|$, hence it remains to show that X' is a multiway cut. Assume for a contradiction that $G - X'$ contains a path P connecting different parts of $\mathcal{T}$, and consider the partition of P into subpaths induced by splitting at every vertex of $T(S)$ that P intersects. Note that every such subpath is either contained in $E(G_S)$ or disjoint from $E(G_S)$, and by assumption at least one such subpath is contained in $E(G_S)$, as otherwise P uses only edges also present in $G - X$. But every such subpath goes between two vertices of $T(S)$ which lie in the same connected component of $G - X$ by definition of $\mathcal{T}'$. Thus every such subpath starts and ends in a single connected component of $G - X$, contradicting that P starts and ends in different components. Therefore X' is a minimum multiway cut for $\mathcal{T}$ in G. Since $e \notin X'$ we are done. □

Let us also briefly note the formal correctness of contracting a non-essential edge. Let G/e denote the result of contracting e in G.

Proposition 3.2. *Let $e \in E(G)$ be a non-essential edge. Then for every $X \subseteq E(G)$ with $e \notin X$, and every partition $\mathcal{T}$ of T, X is a multiway cut for $\mathcal{T}$ in G if and only if it is a multiway cut for $\mathcal{T}$ in G/e. Furthermore, G/e is a multicut-mimicking network for (G, T), and any multicut-covering set $Z \subseteq E(G/e)$ for $(G/e, T)$ is also multicut-covering for (G, T).*

Proof. The first part is clear, since the contraction of an edge in $G - X$ does not change the structure of the connected components. Since e is non-essential, by assumption there exists such an optimal X with $e \notin X$ for every partition $\mathcal{T}$, hence $(G/e, T)$ is a multicut-mimicking network. It also follows that an optimal solution for G always exists in $E(G/e)$, hence a solution-covering set for $(G/e, T)$ is also solution-covering for (G, T). □

The process now works as follows. Recall that (G, T) is (α, c)-dense if $\mathrm{cap}_T(S) \geq |S|^{1/c}/\alpha$ for every set S with $S \cap T \neq \emptyset$ and $|S| \leq |V|/2$ (Def. 2.5). The main technical result is a marking process that marks all essential edges for (G, T) on the condition that (G, T) is (α, c)-dense, and which marks at most k^c edges in total. In such a case, we are clearly allowed to select and contract any unmarked edge of G. Now, assume that (G, T) is not (α, c)-dense. Then by definition there exists a set $S \subset V$ such that $\mathrm{cap}_T(S) < |S|^{1/c}/\alpha$. If we can detect a set S such that $\mathrm{cap}_T(S) < |S|^{1/c}$, then we can recursively compute a multicut-covering set Z_S for $(G_S, T(S))$, consisting of at most $\mathrm{cap}_T(S)^c < |S|$ edges. By the above, we may again select any single edge $e \in E(G_S) \setminus Z_S$ and contract e in G. In either case, we replace G by a strictly smaller graph until $|E(G)| \leq k^c$, at which point we are done.

The two ingredients in the above are thus the marking process for (α, c)-dense graphs, which we present next, and the ability to distinguish the two cases, which has been formalized in the notion of a quasipolynomial expansion tester (Def. 2.6).

3.2 The dense case

Let us now focus on the marking procedure. Let a terminal network (G, T) with $\mathrm{cap}_G(T) = k$ and an integer c be given. We show a process that marks essential edges, on the condition that (G, T) is (α, c)-dense, where we initially assume that α is constant. That is, we prove the following result. The proof takes up the rest of the subsection.

Lemma 3.3. *Let α be a constant. There is a function $c = \Theta(\alpha \log k)$ and a randomized polynomial-time procedure that returns a set of edges $Z \subseteq E(G)$ such that $|Z| \leq k^c$ and if (G, T) is (α, c)-dense then every essential edge for (G, T) is contained in Z.*

In the preliminary version of this paper [42], we gave a multi-phase marking procedure for this purpose, with a relatively complex correctness proof. In this paper, we give a simplified proof, based around the following observation.

Proposition 3.4. *Let $\mathcal{T}$ be a partition of T and X a minimum multiway cut for $\mathcal{T}$. Let $V = V_1 \cup \ldots \cup V_s$ be the partition of V according to the connected components of $G - X$, ordered so that $\mathrm{cap}_T(V_1) \geq \ldots \geq \mathrm{cap}_T(V_s)$. Then for any $i \in [s]$, $\mathrm{cap}_T(V_i) \leq 3k/i$.*

Proof. Since every edge of X is incident with at most two components of $G - X$, we have $\sum_{i=1}^{s} \delta(V_i) \leq 2k$. The additional contribution to $\mathrm{cap}_T(V_i)$ from terminals of T is precisely k in total. Hence $\sum_i \mathrm{cap}_T(V_i) \leq 3k$. On the other hand, if there exists $i \in [s]$ such that $\mathrm{cap}_T(V_i) > 3k/i$ then $\sum_{j=1}^{s} \mathrm{cap}_T(V_j) \geq \sum_{j=1}^{i} \mathrm{cap}_T(V_j) \geq i \cdot \mathrm{cap}_T(V_i) > i \cdot (3k/i) = 3k$. □

Since (G, T) is by assumption (α, c)-dense, for any index i such that $|V_i| \leq |V|/2$ (i.e., for all but at most one index i), it holds that $|V_i| \leq (\alpha \cdot 3k/i)^c$. If $|V(G)| > k^c$, and if c is large enough then it follows that almost all vertices of G are found in the first few components. We shall see that this suffices to allow for a simple marking procedure to capture all essential edges of (G, T).

3.2.1 *Matroid constructions.* Before we show the marking procedure, we need some additional preliminaries. We refer to Oxley [34] for more background on matroids, and to Marx [31] for a more concise, technical presentation, including the presentation of the representative sets lemma. For further examples of usage of representative sets in kernelization„ see Kratsch and Wahlström [20].

A *matroid* is a pair $M = (E, \mathcal{I})$ where $\mathcal{I} \subseteq 2^E$ is the *independent sets* of M, subject to the following axioms.

(1) $\emptyset \in \mathcal{I}$;
(2) if $B \in \mathcal{I}$ and $A \subseteq B$ then $A \in \mathcal{I}$; and
(3) if $A, B \in \mathcal{I}$ with $|B| > |A|$ then there exists an element $x \in B \setminus A$ such that $A + x \in \mathcal{I}$.

A *basis* of M is a maximum independent set of M; the *rank* of M is the size of a basis.

Let A be a matrix, and let E label the columns of A. The *column matroid* of A is the matroid $M = (E, \mathcal{I})$ where $S \in \mathcal{I}$ for $S \subseteq E$ if and only if the columns indexed by S are linearly independent. A matrix A *represents* a matroid M if M is isomorphic to the column matroid of A. We refer to A as a *linear representation* of M.

We need three classes of matroids to build from. First, for a set E, the *uniform matroid* over E of rank r is the matroid

$$U(E,r) := (E, \{S \subseteq E \mid |S| \leq r\}).$$

Uniform matroids are representable over any sufficiently large field.

The second class is a *truncated graphic matroid*. Given a graph $G = (E, V)$, the *graphic matroid* of G is the matroid $M(G) = (E, \mathcal{I})$ where a set $F \subseteq E$ is independent if and only if F is the edge set of a forest in G. Graphic matroids can be deterministically represented over all fields. The *r-truncation* of a matroid $M = (E, \mathcal{I})$ for some $r \in \mathbb{N}$ is the matroid $M' = (E, \mathcal{I}')$ where $S \in \mathcal{I}'$ if and only if $S \in \mathcal{I}$ and $|S| \leq r$. Given a linear representation of M, over some field $\mathbb{F}$, a truncation of M can be computed in randomized polynomial time, possibly by moving to an extension field of $\mathbb{F}$ [31]. There are also methods for doing this deterministically [25], but the basic randomized form will suffice for us.

The final class is more involved. Let $D = (V, A)$ be a directed graph and $S \subseteq V$ a set of source vertices. A set $T \subseteq V$ is *linked to S in D* if there are $|T|$ pairwise vertex-disjoint paths starting in S and ending in T. Let $U \subseteq V$. Then

$$M(D, S, U) = (U, \{T \subseteq U \mid T \text{ is linked to } S \text{ in } D\})$$

defines a matroid over U, referred to as a *gammoid*. Note that by Menger's theorem, a set T is dependent in M if and only if there is an (S, T)-vertex cut in D of cardinality less than $|T|$ (where the cut is allowed to overlap S and T). Like uniform matroids, gammoids are representable over any sufficiently large field, and a representation can be computed in randomized polynomial time [31, 34]. We will work over a variant of gammoids we refer to as *edge-cut gammoids*, which are defined as gammoids, except in terms of edge cuts instead of vertex cuts. Informally, for a graph $G = (V, E)$ and a set of source vertices $S \subseteq V$, the edge-cut gammoid of (G, S) is a matroid on a ground set of edges, where a set F of edges is independent if and only if it can be linked to S via pairwise edge-disjoint paths. However, we also need to introduce the "edge version" of *sink-only copies* of vertices, as used in previous work [20]. That is, we introduce a second set $E' = \{e' \mid e \in E\}$ containing copies of edges $e \in E$ which can only be used as the endpoints of linkages, not as initial or intermediate edges.

More formally, for a graph $G = (V, E)$ and a set of source vertices $S \subseteq E$ we perform the following transformation.

(1) Let $L(G)$ be the line graph of G, i.e., the vertices of $L(G)$ are $V(L(G)) = \{z_e \mid e \in E(G)\}$, and $z_e z_f \in E(L(G))$ if and only if $e \cap f \neq \emptyset$. Let $S_D \subseteq V(L(G))$ be the vertices of $L(G)$ corresponding to the edges $E(S, V)$ in G.
(2) Convert $L(G)$ to a directed graph D_G by replacing every edge $z_e z_f \in E(L_G)$ by a pair of directed edges (z_e, z_f), (z_f, z_e) in $E(D_G)$.
(3) Finally, for every vertex $z_e \in V(D_G)$ introduce a new vertex z'_e, and create a directed edge (v, z'_e) for every edge (v, z_e) in D_G.

Slightly abusing notation, we let E refer to the vertices z_e in D_G, and we let E' refer to the vertices z'_e in D_G. The *edge-cut gammoid of* (G, S) is the gammoid $(D_G, S_D, E \cup E')$. Let us observe the resulting notion of independence.

Proposition 3.5. *Let $G = (V, E)$ and $S \subseteq V$ be given. Let $M = (E \cup E', \mathcal{I})$ be the edge-cut gammoid of (G, S). Let $X \subseteq E \cup E'$ be given, and let $F = (X \cap E) \cup \{e \mid e' \in X \cap E'\}$. Then X is independent in M if and only if there exists a set $\mathcal{P}$ of $|X|$ paths linking F to S, where paths are pairwise edge-disjoint except that if $\{e, e'\} \subseteq X$ for some edge e, then two distinct paths in $\mathcal{P}$ end in e.*

We let $U(E, p)$ denote the uniform matroid of rank p on ground set $E(G)$, $M_G(p)$ the p-truncated graphic matroid of G, and $M(T)$ the edge-cut gammoid of (G, T).

If $M_1 = (E_1, \mathcal{I}_1)$ and $M_2 = (E_2, \mathcal{I}_2)$ are two matroids with $E_1 \cap E_2 = \emptyset$, then their *disjoint union* is the matroid

$$M_1 \uplus M_2 = (E_1 \cup E_2, \{I_1 \cup I_2 \mid I_1 \in \mathcal{I}_1, I_2 \in \mathcal{I}_2\}).$$

If M_1 and M_2 are represented by matrices A_1 and A_2 over the same field, then $M_1 \uplus M_2$ is represented by the matrix

$$A = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$$

We will define matroids M as the disjoint union over several copies of the base matroids $M(T)$, $M_G(p)$ and $U(E, p)$ defined above. We refer to the individual base matroids making up M as the *layers* of M.

Representative sets. Our main technical tool is the representative sets lemma, due to Lovász [28] and Marx [31]. This result has been important in FPT algorithms [10, 31] and has been central to the previous kernelization algorithms for cut problems, including variants of Multiway Cut [20]. We also introduce some further notions.

Definition 3.6. Let $M = (E, \mathcal{I})$ be a matroid and $X, Y \in \mathcal{I}$. We say that Y *extends* X *in* M if $r(X \cup Y) = |X| + |Y|$, or equivalently, if $X \cap Y = \emptyset$ and $X \cup Y \in \mathcal{I}$. Furthermore, let $c = O(1)$ be a constant and let $\mathcal{Y} \subseteq \binom{E}{c}$. We say that a set $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$ *represents* $\mathcal{Y}$ *in* M if the following holds: For every $X \in \mathcal{I}$ for which there exists some $Y \in \mathcal{Y}$ such that Y extends X in M, then there exists some $Y' \in \hat{\mathcal{Y}}$ such that Y' extends X in M.

The representative sets lemma now says the following.

Lemma 3.7 (representative sets lemma [28, 31]). *Let $M = (E, \mathcal{I})$ be a linear matroid represented by a matrix A of rank $r + s$, and let $\mathcal{Y} \subseteq \binom{E}{s}$ be a collection of independent sets of M, where $s = O(1)$. In time polynomial in the size of A and the size of $\mathcal{Y}$, we can compute a set $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$ of size at most $\binom{r+s}{s}$ which represents $\mathcal{Y}$ in M.*

We use the following *product form* of the representative sets lemma, with stronger specialized bounds. Assume that the rank of M is $r = r_1 + \ldots + r_c$, where r_i is the rank of layer i of M. Then Lemma 3.7 gives a bound on $|\hat{\mathcal{Y}}|$ as $\Theta((r_1 + \ldots + r_c)^c)$, but the following bound is significantly better when the layers of M have different ranks.

Lemma 3.8 ([20, Lemma 3.4]). *Let $M = (E, \mathcal{I})$ be a linear matroid, given as the disjoint union of c matroids $M_i = (E_i, \mathcal{I}_i)$, where M_i has rank r_i. Let $\mathcal{Y} \subseteq \binom{E}{c}$ be such that every set $Y \in \mathcal{Y}$ contains precisely one member in each layer M_i of M. Then the representative set $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$ computed by the representative sets lemma will have $|\hat{\mathcal{Y}}| \le \prod_{i=1}^{c} r_i$.*

3.2.2 The marking step. For the marking process, fix an integer i_0 (to be specified later). We define a matroid M as a function of i_0 as the disjoint union of $i_0 - 1$ copies of the edge-cut gammoid $M(T)$ (on disjoint copies of the ground set), one copy of $M_G(k^{c-i_0})$, and one copy of $U(E, k)$. That is, $M = M_1 \uplus \cdots \uplus M_{i_0+1}$, where M_1 through M_{i_0-1} are copies of $M(T)$ (on disjoint copies of the ground set), $M_{i_0} = M_G(k^{c-i_0})$, and $M_{i_0+1} = U(E, k)$. We refer to the first $i_0 - 1$ layers in M as the *gammoid layers* and the last two as the *graphic matroid layer* and the *uniform layer*. Note that a linear representation of M over some common field $\mathbb{F}$ can be computed in randomized polynomial time, since every layer of M can be represented over any sufficiently large field.

For each edge $e \in E$, let $t(e)$ be the set that contains a copy of z'_e in every gammoid layer, and a copy of e in the graphic matroid and uniform layers. Let

$$F = \{t(e) \mid e \in E\}.$$

We compute a representative set $\hat{F} \subseteq F$ in the matroid M, and let $Z \subseteq E$ be the set of edges represented in $\hat{F}$. An edge $e \in E$ is *marked* if $e \in Z$. We finish the description by observing the bound on the number of marked edges.

Lemma 3.9. *The total number of marked edges is at most k^c.*

Proof. Follows directly from the product form of the representative sets lemma. □

Finally, we note the correctness condition for the marking. Consider a partition $\mathcal{T}$ of T and a corresponding minimum multiway cut $X \subseteq E$. Note that $|X| \leq k$ since $E(T, V)$ is a multiway cut for every partition, and say that X is *covered* if all edges essential for $\mathcal{T}$ are marked. We then have the following.

Lemma 3.10. *Let $V = V_1 \cup \ldots \cup V_s$ be the partition of $G - X$ into connected components, where $|V_1| \geq \ldots \geq |V_s|$. If $|\bigcup_{i=i_0}^{s} V_i| \leq k^{c-i_0}$, then X is covered.*

Proof. Let $e \in X$ be an edge which is essential for $\mathcal{T}$. Let $\mathcal{T} = \{T_1, \ldots, T_s\}$ where $T_i = T \cap V_i$ for $i \in [s]$. Finally, define an independent set I in M as follows. In the i:th gammoid layer, $i < i_0$, I contains copies of vertices z_e from the edges of $\partial(T_i) \cup X$. In the graphic matroid layer, I contains a spanning forest for components V_{i_0} through V_s. In the final layer, I contains the edges of $X - e$. We claim that $t(f)$ extends I if and only if $f = e$.

For the easier direction, we note that $t(f)$ cannot extend I if $e \neq f$. If $f \in E(V_i)$ for some $i < i_0$, then f fails to extend I in layer i. Indeed, independence of $\partial(T_i) \cup X \cup \{f\}$ in $M(T)$ would require that $X \cup \{f\}$ is linked to $T \setminus T_i$, since the vertices of T_i are blocked off by the edges $\partial(T_i)$. But X separates f from $T \setminus T_i$, hence no such linkage is possible. If $f \in E(V_i)$ for $i \geq i_0$, then f fails to extend I in the graphic matroid layer. Finally, if $f \in X$ then f fails to extend I in the uniform matroid layer. Hence it remains to show that $t(e)$ extends I.

For the gammoid layers, this works precisely as in [20]. As noted in [20] (Prop. 1), whether a sink-only copy v' extends a set U in a gammoid (D, S) depends on whether the original copy v is contained in the (S, U)-min cut *closest to* S. Here, including $\partial(T_i)$ in I in layer i effectively turns this condition into a cut between X and $\delta(T \setminus T_i)$. Hence if e' does not extend $X \cup \partial(T_i)$, then there is a min-cut X_2 between X and $T \setminus T_i$ that is closer to $T \setminus T_i$ than X, and $e \notin X_2$. This contradicts that e is essential for $\mathcal{T}$.

For the last two layers, the statement is trivial. Hence $t(f)$ extends I if and only if $f = e$, as promised, and $e \in Z$. □

3.2.3 Correctness. We finish this section by showing that a choice of $c = \Theta(\alpha \log k)$ suffices, when α is constant.

Lemma 3.11. *Set $i_0 = 4\alpha + 1$. There is a value $c = \Theta(\alpha \log k)$ such that the following holds: If (G, T) is (α, c)-dense, then Z contains all essential edges.*

Proof. Let $\mathcal{T}$ be a partition of T and let X be a minimum multiway cut for $\mathcal{T}$. Let $V = V_1 \cup \ldots \cup V_s$ be the connected components of $G - X$ sorted by decreasing value of $\mathrm{cap}_T(V_i)$, and let $j \in [s]$ be the index that maximizes $|V_j|$. By Prop. 3.4, $\mathrm{cap}_T(V_i) \leq 3k/i_0$ for $i \geq i_0$. If furthermore $i \neq j$, then $|V_i| \leq (\alpha \cdot 3k/i_0)^c = (3k/4)^c$ by the denseness guarantee on (G, T). Let $I = \{i_0 - 1, \ldots, s\} \setminus \{j\}$. Then

$$\sum_{i \in I} |V_i| \leq s \cdot (3k/4)^c \leq k \cdot k^c \cdot (3/4)^c.$$

For $c = \log_{4/3}(k^{i_0+1}) = \Theta(\alpha \log k)$, this is upper-bounded by k^{c-i_0}. Thus I indexes a set of at least $s - i_0 + 1$ components whose total size is bounded by k^{c-i_0}, and certainly the $s - i_0 + 1$ smallest components are not larger than this. Thus Lemma 3.10 applies, X is covered, and Z contains all essential edges. □

In Theorem 3.14, we combine this with the decomposition described previously to show the existence of a multicut-covering set with $k^{O(\log k)}$ edges.

3.3 A constructive result

Now, with some slightly more involved calculations and a higher degree in the exponent, we show that this can be achieved constructively. The marking process is the same as in the last section, but we need to be more careful with the constants.

For this section, let $c = \lfloor (\log n)/(\log k) \rfloor - 1$, to ensure that the number of edges marked is $k^c < n$. We will treat this as $c = (1 - o(1)) \log n / \log k$. We show that such a marking process is possible until we reach a bound of $c = O(\log^3 k)$. Hence, we assume $|V| = k^{\Omega(\log^3 k)}$ in the sequel.

We use the Small Set Expansion approximation algorithm of Bansal et al. [3], which has a ratio of $\alpha(|S|) = O(\log n / \sqrt{\log |S|})$. By Lemma 2.7 we may then assume that for any non-empty set $S \subseteq V$, $|S| \leq |V|/2$, we have $\mathrm{cap}_T(S) \geq |S|^{1/c}/\alpha(|S|)$.

By Lemma 3.10, we wish to find a threshold value i_0 such that if the components are sorted in decreasing order of $|V_i|$, then the total number of vertices in components V_i for $i \geq i_0$ is at most k^{c-i_0}. For this, it suffices to show that $|V_i| \leq k^{c-i_0-1}$ for every $i \geq i_0$. We set $i_0 = \Theta(\sqrt{c \log k})$ with a constant factor to be decided. As above, we use $\mathrm{cap}_T(V_i)$ and the denseness bound as a proxy to upper-bound $|V_i|$, making a possible exception for a component V_j with $\mathrm{cap}_T(V_j)$ small but $|V_j| > n/2$.

Let (G, T) be (α, c)-dense with α and c as above. Let $\mathcal{T}$ be a partition of T and X a minimum multiway cut for $\mathcal{T}$. We show that the marking process with parameters c and i_0 cover all essential edges of X.

Lemma 3.12. *Let $V = V_1 \cup \ldots \cup V_s$ be the partition corresponding to connected components of $G - X$, ordered in decreasing value of $\mathrm{cap}_T(V_i)$. With the above parameters, $|V_i| \leq k^{c-i_0-1}$ for all but at most one index $i \geq i_0 - 1$.*

Proof. Let V_j be the largest component, and assume for a contradiction that for some $i \geq i_0 - 1$, $i \neq j$, we have $|V_i| > k^{c-i_0-1}$. By the SSE approximation we use, for some $p = O(1)$ we then have

$$\alpha(|V_i|) \leq \frac{p \log n}{\sqrt{(c - i_0 - 1) \log k}} \leq (1 + o(1)) \frac{pc\sqrt{\log k}}{\sqrt{c - i_0 - 1}},$$

where we can absorb the $1 + o(1)$ factor into p. Furthermore fix the constant in i_0 so that $i_0 = 6p\sqrt{c \log k} + 1$. By Prop. 3.4 we have $\mathrm{cap}_T(V_i) \leq 3k/i \leq 3k/(i_0 - 1)$, hence the denseness guarantee is

$$|V_i| \leq (3\alpha k/(i_0 - 1))^c = k^c \cdot \left(\frac{3pc\sqrt{\log k}}{6p\sqrt{c \log k}\sqrt{c - i_0 - 1}} \right)^c = k^c \cdot \left(\frac{1}{2} \cdot \sqrt{\frac{c}{(c - i_0 - 1)}} \right)^c.$$

Since $c = \Omega(\log^3 k)$, we have $i_0 = \Theta(\sqrt{c \log k}) = o(c)$. Hence $c/(c - i_0 - 1) = 1 + (i_0 + 1)/(c - i_0 - 1) = 1 + o(1)$ and we claim

$$|V_i| \leq k^c \cdot \left(\frac{1 + o(1)}{2} \right)^c \leq k^{c-i_0-1},$$

contradicting our assumption. Indeed, $k^{i_0} = 2^{\Theta(c^{1/2} \log^{3/2} k)} \leq 2^c$ for some $c = \Omega(\log^3 k)$. Thus the contradiction is complete and $|V_i| \leq k^{c-i_0-1}$. □

Since there are at most k components, it follows that after the $i_0 - 1$ largest components are removed, the remaining components contain fewer than k^{c-i_0} vertices in total, as required. Hence we have the following.

Lemma 3.13. *If $|V(G)| \geq k^{\Omega(\log^3 k)}$ and (G,T) is (α,c)-dense as above, then there is a process that marks all essential edges of (G,T) while leaving at least one edge unmarked.*

3.4 Completing the result

We now put the pieces together to show the non-constructive and constructive bounds on the size of a multicut-covering set.

Theorem 3.14 (Theorem 1.1 restated). *Let A be a quasipolynomial expansion tester with ratio $\alpha(n,k)$. Let (G,T) be a terminal network with $\mathrm{cap}_G(T) = k$. There is a multicut-covering set $Z \subseteq E(G)$ with $|Z| \leq k^{O(\alpha(n,k)\log k)}$, which furthermore can be computed in randomized polynomial time using calls to A.*

Proof. Set $c = \Theta(\alpha \log k)$ as in Lemma 3.11. If $|E(G)| \leq k^c$ then return $Z = E(G)$. If there is a vertex $v \in V(G) \setminus T$ with $d(v) \leq 1$, then delete it. This is clearly a correct reduction rule. Otherwise, we compute a non-essential edge e as follows. Call A on (G,T,c). If A reports that (G,T) is (α,c)-dense, then Lemma 3.3 applies. Compute a set Z containing all essential edges, with $|Z| \leq k^c$, guaranteeing that there is a non-essential edge $e \in E(G) \setminus Z$.

If A returns a set $S \subseteq V(G)$, let $k_S = \mathrm{cap}_T(S)$. Let $(G_S, T(S))$ be the recursive instance at S, and note that $|V(G_S)| = |N_G[S]| < |V|$ and $|S| > k_S^c$ by definition of A. We may now proceed by induction on $|V|$ and assume that we can compute a multicut-covering set $Z_S \subseteq E(G_S)$ of size $|Z_S| \leq k_S^c < |V(G_S)|$. On the other hand, we have

$$|E(G_S)| = \frac{1}{2} \sum_{v \in V(G_S)} d_{G_S}(v) \geq |S|$$

since $d_G(v) = d_{G_S}(v) \geq 2$ for every $v \in S$ and $S \subseteq V(G_S)$. Thus $|E(G_S)| \geq |S| > k_S^c \geq |Z_S|$, and there is an edge $e \in E(G_S) \setminus Z_S$. By construction e corresponds directly to an edge in G, and we have found a non-essential edge.

By Prop. 3.2 we may now contract e in G and repeat. This yields a graph G' with $|V(G')| < |V|$, hence by induction we can create a multicut-covering set Z for G', which is also a multicut-covering set of G by Prop. 3.2. Hence we can compute a multicut-covering set Z with $|Z| \leq k^c$. □

We observe the following consequences.

Corollary 3.15. *Let (G,T) be a terminal network with $\mathrm{cap}_G(T) = k$. The following holds.*

1. *There is a multicut-mimicking network for (G,T) with $k^{O(\log k)}$ edges.*
2. *A multicut-mimicking network with $k^{O(\log^3 k)}$ edges can be computed in randomized polynomial time.*

Proof. The first is immediate using $\alpha(n,k) = 1$. For the second, we need to use Lemma 3.13. We assume that $|E(G)| > k^{\Omega(\log^3 k)}$ as required by Lemma 3.13, or else we return (G,T). As in Theorem 3.14 it suffices to locate a single non-essential edge. By Bansal et al. [3] and Lemma 2.7, there is a quasipolynomial expansion tester with a ratio $\alpha(|S|) = O(\log n/\sqrt{\log |S|})$. Call this algorithm with (G,T,c) where $c = (1-o(1))\log n/\log k$ as in Section 3.3. If it finds a set S, recurse on S as in Theorem 3.14; otherwise Lemma 3.13 applies. In both cases we locate a non-essential edge. Eventually we reach the threshold where $|E(G)| = k^{O(\log^3 k)}$, and return the resulting terminal network (G,T) as multicut-covering set. □

Additionally, if better approximation ratios for Small Set Expansion exist, in particular ratios $\alpha = O(\log k)$ or better, then we can improve $k^{O(\log^3 k)}$ to a constructive bound of size $k^{O(\alpha \log k)}$.

3.5 Kernelization extensions and consequences

As noted, we get the following consequences.

Corollary 3.16. *The following problems have randomized quasipolynomial kernels.*

(1) *Edge Multiway Cut parameterized by solution size.*
(2) *Edge Multicut parameterized by the solution size and the number of cut requests.*
(3) *Group Feedback Edge Set parameterized by solution size, for any group.*
(4) *Subset Feedback Edge Set with undeletable edges, parameterized by solution size.*

Proof. For Edge Multiway Cut, let (G, T, k) be an input. Known reduction rules can reduce the instance to one with $\mathrm{cap}_G(T) \leq 2k$ [20]. From this point, the kernel follows.

For Edge Multicut, let the input be $(G, \{(s_1, t_1) \ldots, (s_r, t_r)\}, p)$ and let $k = p+r$. Create a set of $2r$ vertices $T = \{s'_1, \ldots, t'_r\}$ and $p+1$ subdivided parallel edges between s'_i and s_i, and between t'_i and t_i, for each $i \in [r]$. Let G' be the new graph. We claim that $I = (G, \{(s_1, t_1), \ldots, (s_r, t_r)\}, p)$ is a positive instance if and only if $I' = (G', \{(s'_1, t'_1), \ldots, (s'_r, t'_r)\}, p)$ is. Indeed, any multicut for I is a multicut for I', and every multicut for I' containing at most p edges leaves all new terminals s'_i, t'_i connected to the old terminals s_i, t_i and is hence a multicut for I. Furthermore $\mathrm{cap}_{G'}(T) = (p+1)r = O(k^2)$. Now it suffices to compute a multicut-covering set Z for (G', T) and contract all edges in $E(G') \setminus Z$.

For Group Feedback Edge Set (GFES), we follow the approach of [20]. The input to GFES is a tuple (G, ϕ, k), where ϕ is a direction-dependent labelling of the edges of G from some multiplicative group Γ, such that for every $uv \in E$, $\phi(uv) = \phi(vu)^{-1}$ (where the inverse is the group inverse). The goal is to remove k edges such that in the remaining graph, there is an assignment $\lambda\colon V(G) \to \Gamma$ such that for any $uv \in E(G)$ we have $\lambda(v) = \lambda(u) \cdot \phi(uv)$. We will not need any assumptions about how the group elements are represented, other than the ability to test whether a product of elements $\phi(e)$ equals the group identity 1_Γ or not. Refer to a simple cycle C as *unbalanced* if $\prod_{e \in E(C)} \phi(e) \neq 1_\Gamma$, with the product taken in order along C. We first note that GFES has an $O(\log k)$-approximation. Indeed, GFES reduces easily to Group Feedback Vertex Set, which in turn is a special case of the meta-problem Biased Graph Cleaning [41]. Lee and Wahlström [23] showed that Biased Graph Cleaning admits an $O(\log k)$-approximation, using an oracle for testing whether cycles are unbalanced. Let X_0 be an approximate solution with $|X_0| = O(k \log k)$. Let $T = V(X_0)$ be the endpoints of X_0. By assumption, $G - X_0$ admits an assignment $\lambda\colon V(G) \to \Gamma$ as above, and such an assignment λ can be computed by starting with an arbitrary value from one vertex of each connected component. We now follow [20] in *untangling* the group labels, so that every edge except those in X_0 receive the identity label by ϕ. As in [20], the solution to GFES now corresponds to a multiway cut for some unknown partition $\mathcal{T}$ of T, hence the multicut-mimicking network can be used for kernelization.

Subset Feedback Edge Set with undeletable edges, parameterized by solution size, is covered by the previous case, since it is a special case of GFES. Indeed, let $S \subseteq E(G)$ be the special edges. We use labels ϕ from the group Z_2^S where every edge is labelled by ϕ by identity except the edges of S, which flip one bit of the group element each. It is now easy to see that a cycle is balanced if and only if it contains no edge from S. It is furthermore easy to see that we can implement undeletable edges by creating parallel (subdivided) copies of edges, using the same group labels. □

Remark. In the preliminary version of this paper [42], we additionally claimed results for the 0-Extension problem, building on results of Reidl and Wahlström [39]. Unfortunately, the proof of this in [42] is incorrect, and we were unable to fix it. Concretely, using some terminology of [39], let $I = ((G, T), \mu, \tau)$ be an instance of 0-Extension for a terminal network (G, T) and a metric μ. Assume that I has an optimal solution λ with at most k crossing edges, and that G contains a sparse cut, i.e., a set $S \subseteq V(G)$ such that $\mathrm{cap}_T(S) < |S|^{1/c} < k$. Then the proof attempt in the conference version [42] implicitly assumes that λ requires only $O(\mathrm{cap}_T(S))$ crossing edges inside $G[S]$. This does not appear to hold. Therefore we retract our previous claims in [42] regarding quasipolynomial metric sparsifiers. The results of Reidl and Wahlström [39] are not affected by this.

4 DISCUSSION

We defined the notion of a *multicut-mimicking network*, and showed that every terminal network (G, T) with $k = \mathrm{cap}_G(T)$ admits one of size $k^{O(\log k)}$, and that a multicut-mimicking network of size $k^{O(\log^3 k)}$ can be computed in randomized polynomial time. The mimicking network is constructed via contractions on G, i.e., it simply consists of a set of edges which form a *multicut-covering set*. As a consequence of such a result, a range of parameterized problems, starting from Edge Multiway Cut, have randomized quasipolynomial kernels.

A first question is how to bridge the gap between $k^{O(\log k)}$ and $k^{O(\log^3 k)}$. This may be partially possible, depending on the precise approximation guarantee available for Small Set Expansion; but a complete bridging via this approach would require a constant-factor approximation for SSE, which has been conjectured not to exist under the *small set expansion hypothesis*. A more difficult question is what the correct size of a multicut-mimicking network is in general, and whether one of polynomial size exists and can be efficiently computed. A positive solution would confirm the existence of a polynomial kernel for Edge Multiway Cut, which is one of the most significant open questions in kernelization.

We finally note two questions in different directions. First, all results in this paper relate only to edge deletion problems. Can an extension of the method be used to prove the existence or efficient computability of a quasipolynomial mimicking network for *vertex deletion* multicut behaviour? This may introduce significant additional difficulties, especially considering that approximation algorithms for sparse vertex cuts are less well-developed than algorithms for Small Set Expansion [26] (see also Agrawal and Maity for related FPT-results [1], and Louis, Raghavendra and Vempala [27] on the approximability of Vertex Expansion).

Second, the preliminary version of this paper [42] contained mistaken claims about the existence of quasipolynomial kernels and "metric sparsifiers" for 0-Extension instances, subject to a bound on the number of crossing edges in a solution. Can such a result be established? On the other hand, can evidence be established against the existence of a polynomial-sized exact metric sparsifier, depending on a bound on the number of crossing edges k of a solution, or is even such a result plausible?

REFERENCES

[1] Garima Agrawal and Soumen Maity. 2021. The Small Set Vertex Expansion Problem. *Theor. Comput. Sci.* 886 (2021), 84–93. https://doi.org/10.1016/j.tcs.2021.07.017

[2] Alexandr Andoni, Anupam Gupta, and Robert Krauthgamer. 2014. Towards $1 + \varepsilon$-Approximate Flow Sparsifiers. In *SODA*. SIAM, 279–293. https://doi.org/10.1137/1.9781611973402.20

[3] Nikhil Bansal, Uriel Feige, Robert Krauthgamer, Konstantin Makarychev, Viswanath Nagarajan, Joseph Naor, and Roy Schwartz. 2014. Min-Max Graph Partitioning and Small Set Expansion. *SIAM J. Comput.* 43, 2 (2014), 872–904. https://doi.org/10.1137/120873996

[4] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. 2009. On problems without polynomial kernels. *J. Comput. Syst. Sci.* 75, 8 (2009), 423–434. https://doi.org/10.1016/j.jcss.2009.04.001

[5] Julia Chuzhoy. 2012. On vertex sparsifiers with Steiner nodes. In *STOC*. ACM, 673–688. https://doi.org/10.1145/2213977.2214039

[6] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer. https://doi.org/10.1007/978-3-319-21275-3

[7] Marek Cygan, Pawel Komosa, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, Saket Saurabh, and Magnus Wahlström. 2021. Randomized Contractions Meet Lean Decompositions. *ACM Trans. Algorithms* 17, 1 (2021), 6:1–6:30. https://doi.org/10.1145/3426738

[8] Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis. 1994. The Complexity of Multiterminal Cuts. *SIAM J. Comput.* 23, 4 (1994), 864–894. https://doi.org/10.1137/S0097539792225297

[9] Matthias Englert, Anupam Gupta, Robert Krauthgamer, Harald Räcke, Inbal Talgam-Cohen, and Kunal Talwar. 2014. Vertex Sparsifiers: New Results from Old Techniques. *SIAM J. Comput.* 43, 4 (2014), 1239–1262. https://doi.org/10.1137/130908440

[10] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. 2016. Efficient Computation of Representative Families with Applications in Parameterized and Exact Algorithms. *J. ACM* 63, 4 (2016), 29:1–29:60. https://doi.org/10.1145/2886094

[11] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. 2019. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press. https://doi.org/10.1017/9781107415157

[12] Lance Fortnow and Rahul Santhanam. 2011. Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.* 77, 1 (2011), 91–106. https://doi.org/10.1016/j.jcss.2010.06.007

[13] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. 1996. Approximate Max-Flow Min-(Multi)Cut Theorems and Their Applications. *SIAM J. Comput.* 25, 2 (1996), 235–251. https://doi.org/10.1137/S0097539793243016

[14] Torben Hagerup, Jyrki Katajainen, Naomi Nishimura, and Prabhakar Ragde. 1998. Characterizing Multiterminal Flow Networks and Computing Flows in Networks of Small Treewidth. *J. Comput. Syst. Sci.* 57, 3 (1998), 366–375. https://doi.org/10.1006/jcss.1998.1592

[15] Eva-Maria C. Hols and Stefan Kratsch. 2018. A Randomized Polynomial Kernel for Subset Feedback Vertex Set. *Theory Comput. Syst.* 62, 1 (2018), 63–92. https://doi.org/10.1007/s00224-017-9805-6

[16] Nikolai Karpov, Marcin Pilipczuk, and Anna Zych-Pawlewicz. 2019. An Exponential Lower Bound for Cut Sparsifiers in Planar Graphs. *Algorithmica* 81, 10 (2019), 4029–4042. https://doi.org/10.1007/s00453-018-0504-8

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

[18] Stefan Kratsch. 2018. A Randomized Polynomial Kernelization for Vertex Cover with a Smaller Parameter. *SIAM J. Discrete Math.* 32, 3 (2018), 1806–1839. https://doi.org/10.1137/16M1104585

[19] Stefan Kratsch and Magnus Wahlström. 2014. Compression via Matroids: A Randomized Polynomial Kernel for Odd Cycle Transversal. *ACM Trans. Algorithms* 10, 4 (2014), 20:1–20:15. https://doi.org/10.1145/2635810

[20] Stefan Kratsch and Magnus Wahlström. 2020. Representative Sets and Irrelevant Vertices: New Tools for Kernelization. *J. ACM* 67, 3 (2020), 16:1–16:50. https://doi.org/10.1145/3390887

[21] Robert Krauthgamer and Havana Inbal Rika. 2020. Refined Vertex Sparsifiers of Planar Graphs. *SIAM J. Discrete Math.* 34, 1 (2020), 101–129. https://doi.org/10.1137/17M1151225

[22] Robert Krauthgamer and Inbal Rika. 2013. Mimicking Networks and Succinct Representations of Terminal Cuts. In *SODA*. SIAM, 1789–1799. https://doi.org/10.1137/1.9781611973105.128

[23] Euiwoong Lee and Magnus Wahlström. 2020. LP-branching algorithms based on biased graphs. *CoRR* abs/1610.06060v2 (2020). Extended version of previous version (Wahlström, SODA 2017).

[24] Frank Thomson Leighton and Ankur Moitra. 2010. Extensions and limits to vertex sparsification. In *STOC*. ACM, 47–56. https://doi.org/10.1145/1806689.1806698

[25] Daniel Lokshtanov, Pranabendu Misra, Fahad Panolan, and Saket Saurabh. 2018. Deterministic Truncation of Linear Matroids. *ACM Trans. Algorithms* 14, 2 (2018), 14:1–14:20. https://doi.org/10.1145/3170444

[26] Anand Louis and Yury Makarychev. 2016. Approximation Algorithms for Hypergraph Small-Set Expansion and Small-Set Vertex Expansion. *Theory Comput.* 12, 1 (2016), 1–25. https://doi.org/10.4086/toc.2016.v012a017

[27] Anand Louis, Prasad Raghavendra, and Santosh S. Vempala. 2013. The Complexity of Approximating Vertex Expansion. In *FOCS*. IEEE Computer Society, 360–369. https://doi.org/10.1109/FOCS.2013.46

[28] László Lovász. 1977. Flats in matroids and geometric graphs. In *Proc. Sixth British Combinatorial Conf. (Combinatorial Surveys)*. 45–86.

[29] Pasin Manurangsi. 2018. Inapproximability of Maximum Biclique Problems, Minimum k-Cut and Densest At-Least-k-Subgraph from the Small Set Expansion Hypothesis. *Algorithms* 11, 1 (2018), 10. https://doi.org/10.3390/a11010010

[30] Dániel Marx. 2006. Parameterized graph separation problems. *Theor. Comput. Sci.* 351, 3 (2006), 394–406. https://doi.org/10.1016/j.tcs.2005.10.007

[31] Dániel Marx. 2009. A parameterized view on matroid optimization problems. *Theor. Comput. Sci.* 410, 44 (2009), 4471–4479. https://doi.org/10.1016/j.tcs.2009.07.027

[32] Ankur Moitra. 2009. Approximation Algorithms for Multicommodity-Type Problems with Guarantees Independent of the Graph Size. In *FOCS*. IEEE Computer Society, 3–12. https://doi.org/10.1109/FOCS.2009.28

[33] G. Nemhauser and L. Trotter. 1975. Vertex packing: structural properties and algorithms. *Mathematical Programming* 8 (1975), 232–248. https://doi.org/10.1007/BF01580444

[34] James Oxley. 2011. *Matroid Theory* (2nd ed.). Oxford University Press.

[35] Harald Räcke. 2002. Minimizing Congestion in General Networks. In *FOCS*. IEEE Computer Society, 43–52. https://doi.org/10.1109/SFCS.2002.1181881

[36] Harald Räcke. 2008. Optimal hierarchical decompositions for congestion minimization in networks. In *STOC*. ACM, 255–264. https://doi.org/10.1145/1374376.1374415

[37] Prasad Raghavendra and David Steurer. 2010. Graph expansion and the unique games conjecture. In *STOC*. ACM, 755–764. https://doi.org/10.1145/1806689.1806792

[38] Prasad Raghavendra, David Steurer, and Madhur Tulsiani. 2012. Reductions between Expansion Problems. In *Computational Complexity Conference*. IEEE Computer Society, 64–73.

[39] Felix Reidl and Magnus Wahlström. 2018. Parameterized Algorithms for Zero Extension and Metric Labelling Problems. In *ICALP (LIPIcs, Vol. 107)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 94:1–94:14. https://doi.org/10.4230/LIPIcs.ICALP.2018.94

[40] A. Schrijver. 2003. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer.

[41] Magnus Wahlström. 2017. LP-branching algorithms based on biased graphs. In *SODA*. SIAM, 1559–1570. https://doi.org/10.1137/1.9781611974782.102

[42] Magnus Wahlström. 2020. On Quasipolynomial Multicut-Mimicking Networks and Kernelization of Multiway Cut Problems. In *ICALP (LIPIcs, Vol. 168)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 101:1–101:14. https://doi.org/10.4230/LIPIcs.ICALP.2020.101