

Min-Max Graph Partitioning and Small Set Expansion*

Nikhil Bansal[†] Uriel Feige[‡] Robert Krauthgamer[§] Konstantin Makarychev[¶]
Viswanath Nagarajan^{||} Joseph (Seffi) Naor^{**} Roy Schwartz^{††}

Abstract

We study graph partitioning problems from a *min-max* perspective, in which an input graph on n vertices should be partitioned into k parts, and the objective is to minimize the maximum number of edges leaving a single part. The two main versions we consider are where the k parts need to be of equal-size, and where they must separate a set of k given terminals. We consider a common generalization of these two problems, and design for it an $O(\sqrt{\log n \log k})$ -approximation algorithm. This improves over an $O(\log^2 n)$ approximation for the second version due to Svitkina and Tardos [ST04], and roughly $O(k \log n)$ approximation for the first version that follows from other previous work. We also give an $O(1)$ -approximation algorithm for graphs that exclude any fixed minor.

Our algorithm uses a new procedure for solving the Small-Set Expansion problem. In this problem, we are given a graph G and the goal is to find a non-empty set $S \subseteq V$ of size $|S| \leq \rho n$ with minimum edge-expansion. We give an $O(\sqrt{\log n \log(1/\rho)})$ bicriteria approximation algorithm for Small-Set Expansion in general graphs, and an improved factor of $O(1)$ for graphs that exclude any fixed minor.

1 Introduction

We study graph partitioning problems from a *min-max* perspective. Typically, graph partitioning problems ask for a partitioning of the vertex set of an undirected graph under some problem-specific constraints on the different parts, e.g., balanced partitioning or separating terminals, and the objective is *min-sum*, i.e., minimizing the total weight of the edges connecting different parts. In the *min-max* variant of these problems, the goal is different — minimize the weight of the edges leaving a single part, taking the maximum over the different parts. A canonical example, that we

*An extended abstract of this paper appeared in Proceedings of 52nd FOCS, 2011.

[†]Eindhoven University of Technology, The Netherlands. Work supported in part by the Netherlands Organisation for Scientific Research (NWO) grant #639.022.211. E-mail: n.bansal@tue.nl

[‡]Weizmann Institute of Science, Rehovot, Israel. Work supported in part by The Israel Science Foundation (grant #873/08). Email: uriel.feige@weizmann.ac.il

[§]Weizmann Institute of Science, Rehovot, Israel. Work supported in part by the Israel Science Foundation (grant #452/08), the US-Israel BSF (grant #2010418), and by a Minerva grant. Email: robert.krauthgamer@weizmann.ac.il

[¶]Microsoft Research, One Microsoft Way, Redmond, WA 98052. Email: konstantin.makarychev@microsoft.com

^{||}IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598. Email: viswanath@us.ibm.com

^{**}Computer Science Dept., Technion, Haifa, Israel. Supported by ISF grant 954/11 and BSF grant 2010426. Email: naor@cs.technion.ac.il

^{††}Microsoft Research, One Microsoft Way, Redmond, WA 98052. Email: roysch@microsoft.com

consider throughout the paper, is the Min-Max k -Partitioning problem: given an undirected graph $G = (V, E)$ with nonnegative edge-weights and $k \geq 2$, partition the vertices into k (roughly) equal parts S_1, \dots, S_k so as to minimize $\max_i \delta(S_i)$, where $\delta(S)$ denotes the sum of edge-weights in the cut $(S, V \setminus S)$. We design for this problem a bicriteria approximation algorithm. Throughout, let $w : E \rightarrow \mathbb{R}^+$ denote the edge-weights and let $n = |V|$.

Min-max partitions arise naturally in many settings. Consider the following application in the context of cloud computing, which is a special case of the general graph-mapping problem considered in [BLNZ11] (and also implicit in other previous works [ZA06, YYRC08, CRB09]). There are n processes communicating with each other, and there are k machines, each having a bandwidth capacity C . The goal is to allocate the processes to machines in a way that balances the load (roughly n/k processes per machine), and meets the outgoing bandwidth requirement. Viewing the processes as vertices and the traffic between them as edge-weights, we get the Min-Max k -Partitioning problem. In general, balanced partitioning (either min-sum or min-max) is at the heart of many heuristics that are used in a wide range of applications, including VLSI layout, circuit testing and simulation, parallel scientific processing, and sparse linear systems.

Balanced partitioning, particularly in its min-sum version, has been studied extensively during the last two decades, with impressive results and connections to several fields of mathematics, see e.g. [LR99, ENRS99, LLR95, AR98, ARV08, KNS09, LN06, CKN09]. The min-max variants, in contrast, have received much less attention. Kiwi, Spielman and Teng [KST98] formulated the Min-Max k -Partitioning problem and designed algorithms achieving absolute bounds for special classes of graphs such as meshes and planar graphs. Prior to our work, no approximation algorithm for the Min-Max k -Partitioning problem was given explicitly for general graphs, and the approximation that follows from known results is not smaller than $O(k\sqrt{\log n})$.¹ However, Raghavendra, Steurer and Tulsiani [RST12, Theorem IV.5] recently proved that a constant factor approximation for this problem, even for $k = 2$ and even a bicriteria approximation, is SSE-hard, meaning that it would refute the so-called SSE hypothesis (which asserts a certain promise version of the SSE problem defined below does not have a polynomial-time algorithm). A different min-max partitioning problem, *Min-Max Multiway Cut* was studied by Svitkina and Tardos [ST04], who designed an $O(\log^2 n)$ -approximation algorithm.

An important tool in our result above is an approximation algorithm for the Small-Set Expansion (SSE) problem. This problem was suggested recently by Raghavendra and Steurer [RS10] (see also [RST10, RST12]) in the context of the unique games conjecture. Recall that the *edge-expansion* of a subset $S \subseteq V$ with $0 < |S| \leq \frac{1}{2}|V|$ is

$$\Phi(S) := \frac{\delta(S)}{|S|}.$$

The input to the SSE problem is an edge-weighted graph and $\rho \in (0, \frac{1}{2}]$, and the goal is to compute

$$\Phi_\rho := \min_{|S| \leq \rho n} \Phi(S).$$

Raghavendra, Steurer and Tetali [RST10] designed for the SSE problem in d -regular graphs an algorithm that approximates the expansion of small sets within a factor of $O(\sqrt{(d/\Phi_\rho) \log(1/\rho)})$,

¹One could reduce the problem to the min-sum version of k -partitioning. The latter admits bicriteria approximation $O(\sqrt{\log n \log k})$ [KNS09], but the reduction loses another factor of $k/2$. Another possibility is to repeatedly remove n/k vertices from the graph, paying again a factor of $k/2$ on top of the approximation in a single iteration, which is, say, $O(\log n)$ by [Räc08].

while violating the bound on $|S|$ by no more than a constant multiplicative factor, i.e., a bicriteria approximation.² Notice that the approximation factor depends on Φ_ρ ; this is not an issue if every small set expands well, but in general Φ_ρ can be as small as $1/\text{poly}(n)$, in which case this guarantee is quite weak.

Using Räcke's technique of converting the graph into a tree [Räc08], one can obtain an $O(\log n)$ -approximation for SSE for any value of ρ , without violating the bound $|S| \leq \rho n$.³ A better approximation factor of $O(\sqrt{\log n})$ is achieved by Arora, Rao and Vazirani [ARV08] at the price of slightly violating the size constraint, namely a bicriteria approximation algorithm. However, this latter result works only for $\rho = \Omega(1)$. In our context of min-max problems we need the case $\rho = 1/k$, where $k = k(n)$ is part of the input. Therefore, it is desirable to extend the $O(\sqrt{\log n})$ bound of [ARV08] to a large range of values for ρ .

1.1 Main Results

Our two main results are bicriteria approximation algorithms for the Min-Max k -Partitioning and the Small-Set Expansion problems; here, the word bicriteria means that the output generated by our algorithms violates the size constraints by at most a multiplicative constant, while the cost of the solution is compared to that of an optimal solution (that does satisfy the size constraints). The notation $O_\varepsilon(t)$ hides multiplicative factors depending on ε , i.e., stands for $O(f(\varepsilon) \cdot t)$.

Definition 1.1. An (α, β) bicriteria approximate solution for the Min-Max k -Partitioning problem is a partition S_1, \dots, S_k of the input graph such that:

1. $\max_{1 \leq i \leq k} \delta(S_i) \leq \alpha \cdot \text{OPT}$ (where OPT is the value of an optimal solution); and
2. $\max_{1 \leq i \leq k} |S_i| \leq \beta \cdot \frac{n}{k}$.

Theorem 1.1. For every positive constant $\varepsilon > 0$, Min-Max k -Partitioning admits a bicriteria approximation of $(O_\varepsilon(\sqrt{\log n \log k}), 2 + \varepsilon)$.

Theorem 1.1 provides a polynomial-time algorithm that with high probability outputs a partition S_1, \dots, S_k such that $\max_i |S_i| \leq (2 + \varepsilon) \frac{n}{k}$ and $\max_i \delta(S_i) \leq O(\sqrt{\log n \log k}) \text{OPT}$. We note that the guarantee on part size can be improved slightly to $2 - \frac{1}{k} + \varepsilon$.

Definition 1.2. An (α, β) bicriteria approximate solution for the Small-Set Expansion problem is a non-empty cut $S \subseteq V$ such that:

1. $\Phi(S) \leq \alpha \cdot \text{OPT}$ (where OPT is the value of an optimal solution); and
2. $|S| \leq \beta \cdot \rho n$.

Theorem 1.2. For every constant $\varepsilon > 0$, Small-Set Expansion admits a bicriteria approximation of $(O_\varepsilon(\sqrt{\log n \log(1/\rho)}), 1 + \varepsilon)$.

Theorem 1.2 provides a polynomial-time algorithm that with high probability outputs a cut $S \subseteq V$ of size $0 < |S| \leq (1 + \varepsilon) \rho n$ whose edge-expansion is $\delta(S)/|S| = O(\sqrt{\log n \log(1/\rho)}) \text{OPT}$. We point out that our algorithm actually handles a more general version, called Weighted Small-Set Expansion, which is required in the proof of Theorem 1.1. So in a sense, Theorem 1.1 is proved by reducing Min-Max k -Partitioning to Weighted Small-Set Expansion. We defer the precise details to Section 2.

²The algorithm of [RST10] works also in general (non-regular) graphs for the closely related objective of *conductance*.

³For very small values of ρ , roughly $\rho n \leq O(\log^2 n)$, a better approximation ratio is known [FKN03].

1.2 Additional Results and Extensions

1.2.1 ρ -Unbalanced Cut

Closely related to the SSE problem is the ρ -Unbalanced Cut problem. The input is again a graph $G = (V, E)$ with nonnegative edge-weights and a parameter $\rho \in (0, \frac{1}{2}]$, and the goal is to find a subset $S \subseteq V$ of size $|S| = \rho n$ that minimizes $\delta(S)$. The relationship between this problem and SSE is similar to the one between Balanced Cut and Sparsest Cut.

Definition 1.3. *An (α, γ, β) bicriteria approximate solution for the ρ -Unbalanced Cut problem is a cut $S \subseteq V$ such that:*

1. $\delta(S) \leq \alpha \cdot \text{OPT}$ (where OPT is the value of an optimal solution); and
2. $\gamma \cdot \rho n \leq |S| \leq \beta \cdot \rho n$.

Theorem 1.3. *For every constant $0 < \varepsilon < 1$, the ρ -Unbalanced Cut problem admits a bicriteria approximation of $(O_\varepsilon(\sqrt{\log n \log(1/\rho)}), \Omega(1), 1 + \varepsilon)$.*

Theorem 1.3 provides a polynomial-time algorithm that with high probability finds a cut $S \subseteq V$ of size $\Omega(\rho n) \leq |S| \leq (1 + \varepsilon)\rho n$ and value $\delta(S) \leq O_\varepsilon(\sqrt{\log n \log(1/\rho)})\text{OPT}$. This result generalizes the bound of [ARV08] from $\rho = \Omega(1)$ to *any* value of $\rho \in (0, \frac{1}{2}]$. Our factor is better than the $O(\log n)$ approximation ratio that follows from [Räc08], at the price of slightly violating the size constraint. Our algorithm actually handles a more general version, called Weighted ρ -Unbalanced Cut, which is required in Theorem 1.1. We defer the precise details to Section 2.4.

1.2.2 Min-Max-Multiway-Cut

We also consider the Min-Max-Multiway-Cut problem [ST04]. The input is an undirected graph with nonnegative edge-weights and k terminal vertices t_1, \dots, t_k . The goal is to partition the vertices into k parts S_1, \dots, S_k (not necessarily balanced), under the constraint that each part contains exactly one terminal, so as to minimize $\max_i \delta(S_i)$. Svitkina and Tardos designed an $O(\alpha \log n)$ -approximation algorithm for this problem, where α is the approximation factor known for Minimum Bisection. Plugging $\alpha = O(\log n)$, due to Räcke [Räc08], the algorithm of Svitkina and Tardos achieves an $O(\log^2 n)$ -approximation. Using a similar algorithm to the one in Theorem 1.1, we obtain a better approximation factor.

Theorem 1.4. *Min-Max-Multiway-Cut admits an $O(\sqrt{\log n \log k})$ -approximation algorithm.*

Somewhat surprisingly, we show that removing the dependence on n for Min-Max-Multiway-Cut appears hard, even though no balance is required. This stands in contrast to its min-sum version, known as Multiway Cut, which admits an $O(1)$ -approximation [CKR00, KKS⁺04]. In particular, we show that if there is a guarantee independent of n , then it would imply a similar (independent of n) guarantee for the min-sum version of k -partitioning. Thus, for large but constant $k = k(\varepsilon)$, we would get an $(O(1), O(1))$ -bicriteria approximation for Min-Sum k -Partitioning⁴, which is a significantly better than current state of art approximation algorithms [ARV08, AR06, KNS09], and is known, in fact, to be SSE-hard [RST12, Theorem IV.5].

Theorem 1.5. *If there is a $k^{1-\varepsilon}$ -approximation algorithm for Min-Max-Multiway-Cut for some constant $\varepsilon > 0$, then there is a (k^2, γ) bicriteria approximation algorithm for Min-Sum k -Partitioning with $\gamma \leq 3^{2/\varepsilon}$.*

⁴Bicriteria approximation for Min-Sum k -Partitioning is defined analogously to Definition 1.1, see also Section 5.

1.2.3 Min–Max Cut

We also consider a common generalization of Min-Max k -Partitioning and Min-Max-Multiway-Cut, which we call Min–Max Cut. In fact we obtain Theorems 1.1 and 1.4 as a special case of our result for Min–Max Cut. The input for the Min–Max Cut problem is an undirected graph $G = (V, E)$ with nonnegative edge-weights, a collection of disjoint terminal sets $T_1, T_2, \dots, T_k \subseteq V$ (possibly empty), and parameters $\rho \in [1/k, 1]$ and $C, D > 0$. The goal is to find (if possible) a partition S_1, \dots, S_k of the input graph such that:

1. $T_i \subseteq S_i, \forall i = [k]$.
2. $|S_i| \leq \rho n, \forall i = [k]$.
3. $\max_{1 \leq i \leq k} \delta(S_i) \leq C$.
4. $\sum_{i=1}^k \delta(S_i) \leq D$.

This problem models the aforementioned cloud computing scenario, where in addition, certain processes are pre-assigned to machines (each set T_i maps to machine $i \in [k]$). The goal is to assign the processes V to k machines according to the pre-assignment and machine load constraints, while bounding both the bandwidth per machine C and the total volume of communication D .

Theorem 1.6. *There is a randomized polynomial time algorithm that given any feasible instance of the Min–Max Cut problem with parameters k, ρ, C, D and any constant $\varepsilon > 0$, finds a partition S_1, \dots, S_k with the following properties:*

1. $T_i \subseteq S_i, \forall i \in [k]$.
2. $|S_i| \leq (2 + \varepsilon)\rho n, \forall i \in [k]$.
3. $\mathbb{E}[\max_{1 \leq i \leq k} \delta(S_i)] \leq O_\varepsilon(\sqrt{\log n \log k})C$.
4. $\mathbb{E}\left[\sum_{i=1}^k \delta(S_i)\right] \leq O_\varepsilon(\sqrt{\log n \log k})D$.

1.2.4 Excluded-Minor Graphs

Finally, we obtain an improved approximation – *constant factor* – for SSE in graphs excluding a fixed minor.

Theorem 1.7. *For every constant $\varepsilon > 0$, Small-Set Expansion admits:*

1. *A bicriteria approximation of $(O_\varepsilon(r^2), 1 + \varepsilon)$ on graphs excluding a $K_{r,r}$ -minor.*
2. *A bicriteria approximation of $(O_\varepsilon(\log g), 1 + \varepsilon)$ on graphs of genus $g \geq 1$.*

These bounds extend to the ρ -Unbalanced Cut problem, and by plugging them into the proofs of Theorems 1.1, 1.4, and 1.6 we achieve an improved (bicriteria) approximation ratio of $O(r^2)$ for Min-Max k -Partitioning, Min-Max-Multiway-Cut and Min–Max Cut in graphs excluding a $K_{r,r}$ -minor, and an improved approximation ratio of $O(\log g)$ for the same problems in genus g graphs.

1.3 Techniques

For clarity, we restrict the discussion here mostly to our main application, Min-Max k -Partitioning. Our approach has two main ingredients. First, we reduce the problem to a weighted version of SSE, showing that an α (bicriteria) approximation for the latter can be used to achieve an $O(\alpha)$ (bicriteria) approximation for Min-Max k -Partitioning. Second, we design an $O_\epsilon(\sqrt{\log n \log(1/\rho)})$ (bicriteria) approximation for Weighted Small-Set Expansion (our applications will use $\rho = 1/k$).

The Small-Set Expansion Problem: Let us first examine SSE, and assume for simplicity of presentation that $\rho = 1/k$. Note that SSE bears obvious similarity to both Balanced Cut and Min-Sum k -Partitioning—its solution contains a single cut (vertex-subset) with a size condition, as in Balanced Cut, but the size of this cut is n/k similarly to the k pieces in Min-Sum k -Partitioning). Thus, our algorithm is inspired by, but different from, the approximation algorithms known for these two problems [ARV08, KNS09]. As in these two problems, we use a semidefinite programming (SDP) relaxation to compute an ℓ_2^2 metric on the graph vertices. However, new spreading constraints are needed since SSE is highly asymmetric in its nature — it contains only a *single* cut of size n/k . We devise a randomized rounding procedure based on the *orthogonal separator* mechanics, first introduced by Chlamtac, Makarychev, and Makarychev [CMM06] in the context of the unique games problem. These ideas lead to an algorithm that computes a cut S of *expected* size $|S| \leq O(n/k)$ and of *expected* cost $\delta(S) \leq O(\sqrt{\log n \log k})$ times the SDP value. An obvious concern is that both properties occur only in expectation and might be badly correlated, e.g., the expected edge-expansion $\mathbb{E}[\delta(S)/|S|]$ might be extremely large. Nevertheless, we prove that with good probability, both $|S| = O(n/k)$ and $\delta(S)/|S|$ is sufficiently small.

Excluded-Minor and Bounded-Genus Graphs: For SSE on excluded-minor and bounded-genus graphs, we give a constant factor approximation guarantee, by extending the notion of orthogonal separators to linear programs (LPs) and designing such “LP separators” of low-distortion for these special graph families. The proof uses the probabilistic decompositions of Klein, Plotkin, and Rao [KPR93] and Lee and Sidiropoulos [LS10]. We believe that this result may be of independent interest – the LP formulation for SSE is non-trivial and requires new spreading constraints. We remark that even on planar graphs, the decomposition of Räcke [Räc08] suffers an $\Omega(\log n)$ loss in the approximation guarantee, and thus does not yield a $o(\log n)$ ratio for SSE on this class of graphs.

From SSE to Min-Max Partitioning: Several natural approaches for designing an approximation algorithm for Min-Max k -Partitioning fail. First, reducing the problem to trees à la Räcke [Räc08] is not very effective, because there might not be a single tree in the distribution that preserves *all* the k cuts simultaneously. Standard arguments show that the loss might be a factor of $O(k \log n)$ in the case of k different cuts. Second, one can try and formulate a relaxation for the problem. However, the natural linear and semidefinite relaxations both have large integrality gaps. As a case study, consider for a moment Min-Max-Multiway-Cut. The standard linear relaxation of Calinescu, Karloff and Rabani [CKR00] was shown by Svitkina and Tardos [ST04] to have an integrality gap of $k/2$. In Appendix A we extend this gap to the semidefinite relaxation that includes all ℓ_2^2 triangle inequality constraints. A third attempt is a greedy approach that repeatedly removes from the graph, using SSE, pieces of size $\Theta(n/k)$. However, by removing the “wrong” vertices from the graph, this process might arrive at a subgraph in which every cut of $\Theta(n/k)$

vertices has edge-weight greater by a factor of $\Theta(k)$ than the original optimum (see Appendix B for details). Thus, a different approach is needed.

Our Approach: Our approach is to use multiplicative weight-updates on top of the algorithm for weighted SSE. This yields a collection \mathcal{S} of sets S , each of size $|S| = \Theta(n/k)$ and cost $\delta(S) \leq O(\sqrt{\log n \log k})\text{OPT}$, that covers every vertex $v \in V$ at least $\Omega(|\mathcal{S}|/k)$ times. Alternatively, by assigning each set $S \in \mathcal{S}$, a fractional value of $k/|S|$, we can view this as a fractional covering of vertices by valid configurations, where valid configurations correspond to sets S with $|S| = \Theta(n/k)$ and $\delta(S) = O(\sqrt{\log n \log k}) \cdot \text{OPT}$.

Next, we randomly sample sets S_1, \dots, S_t from \mathcal{S} until V is covered, and derive a partition given by $P_1 = S_1$, $P_2 = S_2 \setminus S_1$, and in general $P_i = S_i \setminus (\cup_{j < i} S_j)$. This method of converting a cover into a partition was previously used in [AK09, Theorem 2], although it is somewhat counter-intuitive, since the sets P_i may have a very large cost $\delta(P_i)$. We use the fact that S_1, \dots, S_t are chosen randomly (and in particular are in random order), to show that the *total* expected boundary of the partition is not very large, i.e.,

$$\mathbb{E} \left[\sum_i \delta(P_i) \right] \leq O(k\sqrt{\log n \log k}) \text{OPT}.$$

Then, we start fixing the partition by the following local operation: find a P_i violating the constraint $\delta(P_i) \leq O(\sqrt{\log n \log k})\text{OPT}$, replace it with the (unique) S_i which defined it, and adjust all other sets P_j accordingly. Somewhat surprisingly, we prove that this local fixing procedure terminates quickly. Finally, the resulting partition consists of sets P_i , each of which satisfies all the necessary properties. However, the number of these sets might be very large, i.e., much larger than k . Thus, the last step of the rounding is to merge small sets together. We show that this can be done while maintaining simultaneously the constraints on the sizes and on the costs of the sets.

Organization. We first show in Section 2 how to approximate Weighted Small-Set Expansion in both general and excluded-minor graphs, and then show (Section 2.4) that an approximation algorithm for Weighted Small-Set Expansion also yields an approximation algorithm for Weighted ρ -Unbalanced Cut. In Section 3 we present an approximation algorithm for Min-Max k -Partitioning that uses the aforementioned algorithm for ρ -Unbalanced Cut. The common generalization of both Min-Max k -Partitioning and Min-Max-Multiway-Cut, Min-Max Cut, appears in Section 4. Finally, Theorem 1.5, the hardness result for Min-Max-Multiway-Cut, is proved in Section 5.

2 Approximation Algorithms for Small-Set Expansion

In this section we design approximation algorithms for the Small-Set Expansion problem. Our main result is for general graphs and uses an SDP relaxation. It actually holds for a slight generalization of the problem, where expansion is measured with respect to vertex weights (see Definition 2.1 and Theorem 2.1). This generality is useful later in using this algorithm as a subroutine for min-max partitioning. We further obtain improved approximation for certain graph families such as planar graphs (see Section 2.3).

To simplify notation, we shall assume that vertex weights are normalized: we consider measures μ and η on a vertex set V with $\mu(V) = \eta(V) = 1$. For $u \in V$, we denote $\mu(u) = \mu(\{u\})$ and

$\eta(u) = \eta(\{u\})$. We let (V, w) denote a complete (undirected) graph on vertex set V with edge-weight $w(u, v) = w(v, u) \geq 0$ for every pair of vertices $u \neq v$ in V . In our context, such (V, w) can easily model a specific edge set E , by simply setting $w(u, v) = 0$ for every non-edge $(u, v) \notin E$. Recall that we denote

$$\delta(S) := \sum_{u \in S, v \in V \setminus S} w(u, v),$$

to be the total weight of edges crossing the cut $(S, V \setminus S)$.

Definition 2.1 (Weighted Small-Set Expansion). *Let $G = (V, w)$ be an edge-weighted graph, and let μ and η be two measures on the vertex set V with $\mu(V) = \eta(V) = 1$. The weighted small-set expansion of G with respect to $\rho \in (0, 1/2]$ is*

$$\Phi_{\rho, \mu, \eta}(G) := \min \left\{ \frac{\delta(S)}{\eta(S)} : \eta(S) > 0, \mu(S) \leq \rho \right\}.$$

Intuitively, η in the above definition is used to measure the sparsity of the cut, thus, the objective is to minimize $\delta(S)/\eta(S)$ and it is required that $\eta(S) > 0$ (i.e. S is not an empty cut with respect to η). On the other hand, μ is used to measure the size of S , hence, it is required that $\mu(S)$ is not too large. Note that when both η and μ are uniform measures, we obtain the usual SSE problem.

Throughout, we restrict attention to cases where $\Phi_{\rho, \mu, \eta}$ is defined (excluding e.g. the case where μ, η are supported on the same single vertex). Part I of the following theorem immediately implies Theorem 1.2 since we can choose both η and μ to be the uniform measures.

Theorem 2.1 (Approximating SSE). *I. For every fixed $\varepsilon > 0$, there is a polynomial-time algorithm that given as input an edge-weighted graph $G = (V, w)$, two measures μ and η on V ($\mu(V) = \eta(V) = 1$), and some $\rho \in (0, 1/2]$, finds a set $S \subseteq V$ satisfying $\eta(S) > 0$, $\mu(S) \leq (1 + \varepsilon)\rho$ and*

$$\frac{\delta(S)}{\eta(S)} \leq D \cdot \Phi_{\rho, \mu, \eta}(G),$$

where $D = O_\varepsilon(\sqrt{\log n \log(1/\rho)})$ and $n = |V|$.

II. Furthermore, when the input contains also a parameter $H \in (0, 1)$, there is an algorithm to find a set $S \subseteq V$ satisfying $\eta(S) \in [\Omega(H), 2(1 + \varepsilon)H]$, $\mu(S) \leq (1 + \varepsilon)\rho$, and

$$\frac{\delta(S)}{\eta(S)} \leq D \cdot \min \left\{ \frac{\delta(T)}{\eta(T)} : \eta(T) \in [H, 2H], \mu(T) \leq \rho \right\},$$

where $D = O_\varepsilon(\sqrt{\log n \log(\max\{1/\rho, 1/H\})})$.

Remark 2.2. *Let us clarify why we need part II of the theorem. In Section 3.1, we repeatedly apply this theorem to generate a family of sets $S \in \mathcal{S}$ that uniformly covers all vertices in V . The measure μ will be uniform and $\rho = \frac{1}{k}$, corresponding to the requirement that each part in Min-Max k -Partitioning contains n/k vertices. Each set $S \in \mathcal{S}$ must satisfy two conditions: $\mu(S) \leq \rho$ and $\delta(S) \leq D \cdot \text{OPT}$. The second measure η is used to ensure that we uniformly cover all vertices: we dynamically update η (starting with the uniform measure) so that yet uncovered vertices have larger weight. We need the lower bound $\eta(S) \geq \Omega(H)$ to ensure adequate new coverage. We also need an upper bound on $\eta(S)$ to ensure that $\delta(S)$ is small:*

$$\delta(S) \leq D \eta(S) \cdot \frac{\text{OPT}}{H} \leq 2(1 + \varepsilon) D \cdot \text{OPT}.$$

We prove part I of the theorem in Section 2.1, and part II in Section 2.2. These algorithms require the following notion of m -orthogonal separators due to Chlamtac, Makarychev, and Makarychev [CMM06].

Definition 2.3 (Orthogonal Separators). *Let X be an ℓ_2^2 metric (i.e., a finite collection of vectors satisfying ℓ_2^2 triangle inequalities) and $m > 0$. A distribution over subsets $S \subseteq X$ is called an m -orthogonal separator of X with distortion $D > 0$, probability scale $\alpha > 0$ and separation threshold $0 < \beta < 1$, if the following conditions hold:*

1. For all $u \in X$ we have $\Pr(u \in S) = \alpha \|u\|^2$.
2. For all $u, v \in X$ with $\|u - v\|^2 \geq \beta \min\{\|u\|^2, \|v\|^2\}$,

$$\Pr(u \in S \text{ and } v \in S) \leq \frac{\min\{\Pr(u \in S), \Pr(v \in S)\}}{m}.$$

3. For all $u, v \in X$ we have $\Pr(I_S(u) \neq I_S(v)) \leq \alpha D \cdot \|u - v\|^2$, where I_S is the indicator function for the set S .

Orthogonal separators is a distribution over subsets of vector which has three properties. The first property states that the probability that a vector u is chosen by this distribution is proportional to its squared norm. The second property intuitively states that for any two vectors u and v , which are far from each other, the event that u is in a random subset is almost independent from the event that v is in a random subset. The third and last property states that the probability that two vectors u and v are separated by a random subset is upper bounded by a quantity which is proportional to the ℓ_2^2 distance of these two vectors.

Theorem 2.2 ([CMM06]). *There exists a polynomial-time randomized algorithm that given a set of vectors X which includes the origin and satisfies the ℓ_2^2 triangle inequality constraints, positive number m , and $0 < \beta < 1$, generates an m -orthogonal separator with distortion $D = O_\beta(\sqrt{\log |X| \log m})$ and scale $\alpha \geq 1/p(|X|)$ for some polynomial p .*

The second requirement in the definition of orthogonal separators was slightly different originally (in [CMM06]), however, exactly the same algorithm and proof works in our case.⁵

2.1 Algorithm I: Small-Set Expansion in General Graphs

In this section we prove part I of Theorem 2.1.

SDP Relaxation. In our relaxation we introduce a vector \bar{v} for every vertex $v \in V$. The objective is to minimize the total weight of cut edges:

$$\min \sum_{(u,v) \in E} w(u,v) \|\bar{u} - \bar{v}\|^2.$$

⁵Let $\|u - v\|^2 \geq \beta \|u\|^2$ and assume w.l.o.g that $\|u\|^2 \leq \|v\|^2$. Using arithmetics one can prove that $\langle u, v \rangle \leq (1 - \beta/2)\|v\|^2$. Applying Lemma 4.1 in [CMM06] implies that $\langle \varphi(u), \varphi(v) \rangle \leq (1 - \beta/2)$. Hence, $\|\varphi(u) - \varphi(v)\|^2 \geq \beta > 0$ and using Corollary 4.6 in [CMM06] finishes the proof since $\|\psi(u) - \psi(v)\| \geq 2\gamma = \sqrt{\beta}/4 > 0$.

We could constrain all vectors \bar{v} to have length at most 1, i.e. $\|\bar{v}\|^2 \leq 1$, but it turns out our algorithm never uses this constraint. We require that the vectors $\{\bar{v} : v \in V\} \cup \{\mathbf{0}\}$, i.e., all the vectors corresponding to V and the origin, satisfy the ℓ_2^2 triangle inequalities. For every $u, v, w \in V$:

$$\begin{aligned}\|\bar{u} - \bar{w}\|^2 &\leq \|\bar{u} - \bar{v}\|^2 + \|\bar{v} - \bar{w}\|^2 \\ \|\bar{u}\|^2 &\leq \|\bar{u} - \bar{v}\|^2 + \|\bar{v}\|^2 \\ \|\bar{u} - \bar{w}\|^2 &\leq \|\bar{u}\|^2 + \|\bar{w}\|^2.\end{aligned}$$

Note that any valid integral solution (where each vector is either zero or a fixed unit vector) satisfies these conditions. Suppose now that we have approximately guessed the measure H of the optimal solution $H \leq \eta(S) \leq 2H$ (this step is not necessary but it simplifies the exposition). This can be done since the measure of every set S lies in the range from $\eta(u)$ to $n \cdot \eta(u)$, where u is the heaviest element in S , hence H can be chosen from the set $\{2^t \eta(u) : u \in V, t = 0, \dots, \lfloor \log_2 n \rfloor\}$ of size $O(n \log n)$. Therefore, we add a constraint:

$$\sum_{v \in V} \|\bar{v}\|^2 \eta(v) \geq H. \quad (1)$$

We also ignore all vertices $v \in V$ with $\eta(v) > 2H$, since they do not participate in the optimal solution, and thus write the constraint (equivalent to removing them from the graph)

$$\|\bar{v}\|^2 = 0, \quad \text{whenever } \eta(v) > 2H \quad (2)$$

Finally, we introduce new spreading constraints which state that for every $u \in V$:

$$\sum_{v \in V} \mu(v) \cdot \min\{\|\bar{u} - \bar{v}\|^2, \|\bar{u}\|^2\} \geq (1 - \rho)\|\bar{u}\|^2. \quad (3)$$

We remark that one could alternatively use a slightly simpler, almost equivalent, constraint:

$$\sum_{v \in V} \langle \bar{u}, \bar{v} \rangle \mu(v) \leq \rho \|\bar{u}\|^2.$$

We chose to use the former formulation because it has an analogous constraint in linear programming, see Section 2.3.

The SDP relaxation used in our algorithm is presented below in its entirety. We also note that the second constraint can be equivalently written as $\langle \bar{u}, \bar{v} \rangle \leq \|\bar{u}\|^2$, and the third constraint can be equivalently written as $\langle \bar{u}, \bar{v} \rangle \geq 0$.

$$\begin{aligned} \mathbf{min} \quad & \sum_{(u,v) \in E} w(u,v) \|\bar{u} - \bar{v}\|^2 \\ \mathbf{s.t.} \quad & \|\bar{u} - \bar{w}\|^2 + \|\bar{w} - \bar{v}\|^2 \geq \|\bar{u} - \bar{v}\|^2, \quad \forall u, v, w \in V, \\ & \|\bar{u} - \bar{v}\|^2 \geq \|\bar{u}\|^2 - \|\bar{v}\|^2, \quad \forall u, v \in V, \\ & \|\bar{u}\|^2 + \|\bar{v}\|^2 \geq \|\bar{u} - \bar{v}\|^2, \quad \forall u, v \in V, \\ & \sum_{v \in V} \mu(v) \cdot \min\{\|\bar{u} - \bar{v}\|^2, \|\bar{u}\|^2\} \geq (1 - \rho)\|\bar{u}\|^2, \quad \forall u \in V, \\ & \|\bar{v}\|^2 = 0, \quad \text{whenever } \eta(v) > 2H, \\ & \sum_{v \in V} \eta(v) \|\bar{v}\|^2 \geq H. \end{aligned} \quad (4)$$

Lemma 2.3. *Assuming a correct guess of H , the optimal value of SDP (4) is at most $2H \cdot \Phi_{\rho, \mu, \eta}(G)$.*

Proof. Let S^* be an optimal solution to the Small-Set Expansion problem, i.e. $\Phi_{\rho, \mu, \eta}(G) = \delta(S^*)/\eta(S^*)$. Construct the following solution to SDP (4). If $v \in S^*$ set $\bar{v} = \mathbf{1}$ for some fixed unit vector, and if $v \notin S^*$ set $\bar{v} = \mathbf{0}$. Assuming that H was guessed correctly (i.e. $H \leq \eta(S^*) \leq 2H$), each vertex $v \in S^*$ has $\eta(v) \leq 2H$, and constraint (1) is also satisfied. It is clear that all the ℓ_2^2 triangle inequality constraints with the origin are satisfied. Let us focus now on the spreading constraints. If $u \in S^*$, then $\bar{u} = \mathbf{1}$ and the sum in the spreading constraint equals $\mu(V \setminus S) \geq 1 - \rho$. Otherwise, if $u \notin S^*$, then $\bar{u} = \mathbf{0}$ and both sides of the spreading constraint equal 0. Hence, we can conclude that the solution we defined for (4) is feasible.

Note that for every edge $(u, v) \in E$, if the edge crosses the cut defined by S^* then it contributes to the sum in the objective of (4) its weight $w(u, v)$. Otherwise, its contribution is 0. Thus, $\delta(S^*)$ equals the objective value of this SDP solution and the lemma follows. \square

We now describe the approximation algorithm.

Approximation Algorithm I. We first informally describe the main idea behind the algorithm. The algorithm solves the SDP relaxation and obtains a set of vectors $\{\bar{u}\}_{u \in V}$, whose value we denote by *SDP*. Now it samples an orthogonal separator, a random set $S \subseteq V$, and returns it. Assume for the moment that the probability scale α equals 1. Since $\Pr(v \in S) = \|\bar{v}\|^2$, we get $\mathbb{E}[\eta(S)] \geq H$. The expected size of the cut is at most $D \cdot \text{SDP}$ by the third property of orthogonal separators; and thus the ratio of expectations

$$\frac{\mathbb{E}[\delta(S)]}{\mathbb{E}[\eta(S)]} \leq D \cdot \frac{\text{SDP}}{H} \leq 2D \cdot \Phi_{\rho, \mu, \eta}(G).$$

We will show that the expected ratio is also bounded by a similar quantity. Moreover, using the second property of orthogonal separators and the spreading constraints, we can show that $\mu(S) \leq (1 + \varepsilon)\rho$. We now proceed to the formal argument.

We may assume that ε is sufficiently small, i.e., $\varepsilon \in (0, 1/4)$. The approximation algorithm guesses a value H satisfying $H \leq \eta(S) \leq 2H$. Then, it solves the SDP while further constraining $\|\bar{u}\| = 0$ for every $u \in V$ with $\eta(u) > 2H$, to obtain a set of vectors $X = \{\bar{v}\}_{v \in V}$. After that, it samples an orthogonal separator S with $m = \varepsilon^{-1}\rho^{-1}$ and $\beta = \varepsilon$ (from Theorem 2.2). For convenience, we let S be the set of vertices corresponding to vectors belonging to the orthogonal separator rather than the vectors themselves. The algorithm repeats the previous step $\lceil \alpha^{-1}n^2 \rceil$ times (recall α is the probabilistic scale of the orthogonal separator). It outputs the set S having minimum $\delta(S)/\eta(S)$ value among those satisfying $0 < \mu(S) < (1 + 10\varepsilon)\rho$. If no S satisfies this constraint (which we shall see happens with an exponentially small probability), the algorithm outputs an arbitrary set satisfying the constraints.

Analysis. We first estimate the probability of the event $\{u \in S \text{ and } \mu(S) < (1 + 10\varepsilon)\rho\}$ for a fixed vertex $u \in V$. Let $A_u = \{v : \|\bar{u} - \bar{v}\|^2 \geq \beta\|\bar{u}\|^2\}$ be all the vertices which are far from u and $B_u = \{v : \|\bar{u} - \bar{v}\|^2 < \beta\|\bar{u}\|^2\}$ all the vertices which are close to u . Assume for now that $u \in S$, and let us show that only a small fraction of A_u is likely to belong to S , and that the entire set B_u has small measure $\mu(B_u)$. Recall that u satisfies the spreading constraint,

$$\sum_{v \in V} \min\{\|\bar{u} - \bar{v}\|^2, \|\bar{u}\|^2\} \cdot \mu(v) \geq (1 - \rho)\|\bar{u}\|^2. \quad (5)$$

The left hand side above is at most $\beta\|\bar{u}\|^2 \cdot \mu(B_u) + \|\bar{u}\|^2 \cdot \mu(A_u)$. Since $\mu(A_u) = 1 - \mu(B_u)$, this implies that $\mu(B_u) \leq \rho/(1 - \beta) = \rho/(1 - \varepsilon) \leq (1 + 2\varepsilon)\rho$.

For an arbitrary $v \in A_u$ (for which $\bar{v} \neq 0$) we have $\|\bar{u} - \bar{v}\|^2 \geq \beta\|\bar{u}\|^2 \geq \beta \min(\|\bar{u}\|^2, \|\bar{v}\|^2)$. By the second property of orthogonal separators, $\Pr(v \in S \mid u \in S) \leq 1/m = \varepsilon\rho$, and thus, the expected measure is $\mathbb{E}[\mu(A_u \cap S) \mid u \in S] \leq \varepsilon\rho$. Now, still conditioning on $u \in S$, the event $\{\mu(S) \geq (1 + 10\varepsilon)\rho\}$ implies the event $\{\mu(A_u \cap S) \geq 8\varepsilon\rho\}$ because $\mu(B_u \cap S) \leq \mu(B_u) \leq (1 + 2\varepsilon)\rho$. Thus by Markov's inequality, $\Pr[\mu(S) \geq (1 + 10\varepsilon)\rho \mid u \in S]$ is at most $1/8$.

Recall that each vertex $u \in V$ belongs to S with probability $\alpha\|\bar{u}\|^2$. Hence, with probability at least $7/8 \alpha\|\bar{u}\|^2$, it holds that both $u \in S$ and $\mu(S) < (1 + 10\varepsilon)\rho$.

Finally, we use the third property of orthogonal separators to bound the cost of the cut $\delta(S)$,

$$\mathbb{E}[\delta(S)] = \sum_{(u,v) \in E} |I_S(u) - I_S(v)| \cdot w(u,v) \leq \alpha D \cdot \sum_{(u,v) \in E} \|\bar{u} - \bar{v}\|^2 \cdot w(u,v) = \alpha D \cdot \text{SDP}, \quad (6)$$

where $D = O_\beta(\sqrt{\log n \log m}) \leq O_\varepsilon(\sqrt{\log n \log(1/\rho)})$ is the distortion of m -orthogonal separators.

Consider the function f :

$$f(S) = \begin{cases} \eta(S) - \delta(S) \cdot \frac{H}{4D \cdot \text{SDP}}, & \text{if } S \neq \emptyset \text{ and } \mu(S) < (1 + 10\varepsilon)\rho; \\ 0, & \text{otherwise.} \end{cases}$$

As the event $\{u \in S \text{ and } \mu(S) < (1 + 10\varepsilon)\rho\}$ holds for each vertex u with probability at least $3/4 \alpha\|\bar{u}\|^2$, and by (6) we have $\mathbb{E}[\delta(S)] \leq \alpha D \cdot \text{SDP}$, we can bound the expectation as

$$\mathbb{E}[f(S)] \geq \sum_{u \in V} \frac{3\alpha\|\bar{u}\|^2 \eta(u)}{4} - \frac{\alpha H}{4} \geq \frac{\alpha H}{2}.$$

The second inequality uses the last SDP constraint $\sum_{u \in V} \|\bar{u}\|^2 \eta(u) \geq H$. The random variable $f(S)$ is always upper bounded by $2nH$, because by the SDP constraints, $\|\bar{u}\| = 0$ whenever $\eta(u) > 2H$. We note that the zeroing of all $u \in V$ such that $\eta(u) > 2H$ is not essential to the algorithm. However, this zeroing enables us to obtain a better bound on the expected value of f , which affects only the number of samples of orthogonal separators the algorithm needs to perform.

Thus, $\Pr[f(S) > 0] \geq \alpha/(4n)$. Therefore, after $\alpha^{-1}n^2$ iterations, with probability exponentially close to 1, the algorithm will find S with $f(S) > 0$. The latter implies that $\eta(S) > 0$, $\mu(S) < (1 + 10\varepsilon)\rho$, and

$$\frac{\delta(S)}{\eta(S)} \leq 4D \cdot \frac{\text{SDP}}{H}.$$

This finishes the proof of Theorem 2.1 part I, since $\text{SDP}/(2H) \leq \Phi_{\rho, \mu, \eta}(G)$.

2.2 Algorithm II: Small-Set Expansion in General Graphs

We now prove part II of Theorem 2.1.

SDP Relaxation. Our SDP relaxation is similar to part I, namely, we use SDP (4) with the following additional constraints. Recall that Algorithm II gets H (an approximate value of $\eta(S)$)

in the optimal solution) as input, and thus does not need to guess it. First, we add a spreading constraint also on the η -measure of the solution

$$\sum_{v \in V} \min\{\|\bar{u} - \bar{v}\|^2, \|\bar{u}\|^2\} \cdot \eta(v) \geq (1 - 2H)\|\bar{u}\|^2, \quad \forall u \in V. \quad (7)$$

Second, we complement the two spreading constraints using different normalizations

$$\|\bar{v}\| \leq 1, \quad \forall v \in V, \quad (8)$$

$$\sum_{v \in V} \|\bar{v}\|^2 \mu(v) \leq \rho. \quad (9)$$

It is not difficult to verify (analogously to Lemma 2.3) that this SDP is indeed a relaxation of the problem stated in Theorem 2.1 part (II).

Approximation Algorithm II. The algorithm solves the SDP relaxation and then executes several iterations. Each iteration computes a vertex subset S by using a variant of Algorithm I from Section 2.1 detailed below, and then “removes” this S from the graph G and the SDP solution, by zeroing the weight of edges incident to S (i.e., discarding these edges) and zeroing the SDP vectors corresponding to vertices in S (i.e., setting them to the origin $\mathbf{0}$). The algorithm maintains the vertices removed so far in a set $T \subset V$, by initializing $T = \emptyset$, and then at each iteration updating $T = T \cup S$. The iterations repeat until either $\mu(T) \geq \rho/4$ or $\eta(T) \geq H/4$, at which point the algorithm returns a vertex subset F determined as follows: if $\mu(T) \leq \rho$ and $\eta(T) \leq H$, then $F = T$; otherwise, $F = S$ (as computed in the last iteration).

To perform a single iteration, repeatedly sample an orthogonal separator S , by applying Theorem 2.2 on the current SDP vectors with $m = 1/(\varepsilon\rho)$ and $\beta = \varepsilon$, until obtaining a set S with a positive f' value according to the definition in (10) below. (Alternatively, one can return an arbitrary feasible S if the number of samples exceeds a certain threshold.)

Remark 2.4. *To handle terminals in the extended version of the problem (see Section 4), we guess a single terminal u that belongs to the optimal solution S (if any), and set $\|\bar{u}\| = 1$ and $\|\bar{v}\| = 0$ for all other terminals v . Since an orthogonal separator never contains the zero vector, the returned solution F will always contain at most one terminal.*

Analysis. Let us examine the effect of the algorithm’s changes to the SDP solution (by zeroing some vectors and removing edges). Notice that the objective value of the SDP may only decrease, and that constraints (8), (9) and the triangle inequalities clearly remain satisfied. Constraint (1) may get violated, but not by too much; the removed vertices have total measure $\eta(T) \leq H/4$, and due to (8) we still have the slightly weaker constraint

$$\sum_{u \in V} \|\bar{u}\|^2 \eta(u) \geq 3H/4. \quad (1')$$

Let us show that the spreading constraints (3) and (7) are still satisfied. Consider the constraint (3) for a given vertex $u \in V$:

$$\sum_{v \in V} \mu(v) \cdot \min\{\|\bar{u} - \bar{v}\|^2, \|\bar{u}\|^2\} \geq (1 - \rho)\|\bar{u}\|^2.$$

If vertex $u \in T$, the constraint holds trivially since the righthand side is 0. If $u \notin T$, we claim that zeroing some vectors \bar{v} can only increase the lefthand side: if $v \notin T$, then the term $\min\{\|\bar{u} - \bar{v}\|^2, \|\bar{u}\|^2\}$ does not change; and if $v \in T$, then it is replaced with $\min\{\|\bar{u} - 0\|^2, \|\bar{u}\|^2\} = \|\bar{u}\|^2$, which is at least as large. The same argument applies to constraint (7).

Recall that Algorithm II repeatedly samples S from the orthogonal separator until obtaining a set S with $f'(S) > 0$, where the function f' is defined (analogously to f in Algorithm I) by

$$f'(S) = \begin{cases} \eta(S) - \delta(S) \cdot \frac{H}{4D \cdot SDP} - \frac{\mu(S)}{4\rho} \cdot H & \text{if } S \neq \emptyset, \mu(S) < (1 + 10\varepsilon)\rho, \eta(S) \leq 2(1 + 10\varepsilon)H; \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

Notice that f' has an extra term and an extra condition involving $\eta(S)$ when compared to f .

Consider now a single sampling of S from the orthogonal separator. The same argument as before shows, using constraints (5) and (7), that for every given $u \in V$,

$$\Pr \left[\mu(S) \leq (1 + 10\varepsilon)\rho \text{ and } \eta(S) \leq 2(1 + 10\varepsilon)H \mid u \in S \right] \geq \frac{3}{4}.$$

Using the new constraint (9), we get $\mathbb{E}[\mu(S)] \leq \alpha\rho$, and recalling the bound (6) on $\mathbb{E}[\delta(S)]$, we obtain altogether

$$\mathbb{E}[f'(S)] \geq \sum_{u \in V} \frac{3\alpha\|\bar{u}\|^2\eta(u)}{4} - \frac{\alpha H}{4} - \frac{\alpha H}{4} \geq \frac{3\alpha \cdot 3H/4}{4} - \frac{\alpha H}{2} \geq \frac{\alpha H}{16},$$

where the second inequality used (1'). As before, we can bound $\Pr[f'(S) > 0] \geq \Omega(\alpha/n)$, because with probability 1 we have $f'(S) \leq \eta(S) \leq 2nH$ (recall that $\|\bar{u}\| = 0$ whenever $\eta(u) > 2H$). So after $O(n^2/\alpha)$ samples, with probability that is exponentially close to 1, the algorithm finds a set of vertices S satisfying $f'(S) > 0$.⁶

At every iteration we have $f'(S) > 0$, which implies that

$$\delta(S) \leq 4D \cdot \frac{SDP}{H} \eta(S), \quad (11)$$

$$\eta(S) \geq H \cdot \mu(S)/(4\rho), \quad (12)$$

and thus also (throughout the iterations)

$$\eta(T) \geq H \cdot \mu(T)/(4\rho). \quad (13)$$

Let T' denote the set T at the end of the algorithm, and let T'' denote the penultimate set T . Also let S' denote the last set S ; so $T' = T'' \cup S'$. Note that $\mu(T'') < \rho/4$ and $\eta(T'') < H/4$ by the iterations' stopping condition. In addition, either $\mu(T') \geq \rho/4$ or $\eta(T') \geq H/4$, but in both cases $\eta(T') \geq H/16$ follows from (13). We consider the two possible outputs:

- $F = T'$. This implies $\mu(T') \leq \rho$ and $\eta(T') \leq H$. And also $\eta(T') \geq H/16$ from above.
- $F = S'$. This means $\mu(T') > \rho$ or $\eta(T') > H$, which implies, respectively, $\mu(S') > \frac{3}{4}\rho$ or $\eta(S') > \frac{3}{4}H$; in either case, $\eta(S') > 3H/16$ follows from (12). Also our algorithm ensures that $\mu(S') \leq (1 + 10\varepsilon)\rho$ and $\eta(S') \leq 2(1 + 10\varepsilon)H$.

We thus see that either output F satisfies $\mu(F) \leq (1 + 10\varepsilon)\rho$, $\eta(F) \leq 2(1 + 10\varepsilon)H$ and $\eta(F) \geq H/16$. Finally, inequality (11) holds for every set S added, and thus a similar inequality holds for the output F as well. This completes the proof of Theorem 2.1 part (II).

⁶In fact, now $f'(S) \leq 2(1 + 10\varepsilon)H$, thus $\Pr[f'(S) > 0] \geq \Omega(\alpha/H)$ and we need only $O(n/\alpha H)$ iterations.

2.3 Small-Set Expansion in Minor-Closed Graph Families

In this subsection we prove Theorem 1.7. We start by writing the LP relaxation (14). For every vertex $u \in V$ we introduce a variable $x(u)$ taking values in $[0, 1]$; and for every pair of vertices $u, v \in V$ we introduce a variable $z(u, v) = z(v, u)$ also taking values in $[0, 1]$. The intended integral solution corresponding to a set $S \subseteq V$ has $x(u) = 1$ if $u \in S$, and $x(u) = 0$ otherwise; $z(u, v) = |x(u) - x(v)|$. Intuitively, $x(u)$ is the distance to a fictional vertex o that never belongs to S . The analogous object in the SDP relaxation (4) is the origin, and indeed, it is instructive to think of $x(u)$ as an analogue of $\|\bar{u}\|^2$ and of $z(u, v)$ as an analogue of $\|\bar{u} - \bar{v}\|^2$.

It is easy to verify that LP (14) below is a relaxation of the Small-Set Expansion problem. The first three constraints say that $z(u, v)$ is a metric (strictly speaking, a semi-metric), by requiring the triangle inequality, even with the fictional vertex o . The novelty of the LP is in the fourth constraint, which is a new spreading constraints for ensuring that the size of S is small.

$$\begin{array}{ll}
 \min & \sum_{(u,v) \in E} w(u, v) z(u, v) \\
 \text{s.t.} & z(u, v) + z(v, w) \geq z(u, w), \quad \forall u, v, w \in V, \\
 & |x(u) - x(v)| \leq z(u, v), \quad \forall u, v \in V, \\
 & x(u) + x(v) \geq z(u, v), \quad \forall u, v \in V, \\
 & \sum_{v \in V} \mu(v) \cdot \min\{x(u), z(u, v)\} \geq (1 - \rho)x(u), \quad \forall u \in V, \\
 & x(v) = 0, \quad \text{whenever } \eta(v) > 2H, \\
 & \sum_{v \in V} \eta(v)x(v) \geq H, \\
 & x(u), z(u, v) \in [0, 1], \quad \forall u, v \in V.
 \end{array} \tag{14}$$

We introduce an analogue of m -orthogonal separators for linear programming, which we call LP separators.

Definition 2.5 (LP separator). *Let $G = (V, E)$ be a graph, and let $\{x(u), z(u, v)\}_{u, v \in V}$ be a set of numbers. We say that a distribution over subsets $S \subseteq V$ is an LP separator of V with distortion $D \geq 1$, probability scale $\alpha > 0$ and separation threshold $\beta \in (0, 1)$ if the following conditions hold:*

1. For all $u \in V$ we have $\Pr(u \in S) = \alpha \cdot x(u)$.
2. For all $u, v \in V$ with $z(u, v) \geq \beta \cdot \min\{x(u), x(v)\}$ we have $\Pr(u \in S \text{ and } v \in S) = 0$.
3. For all $(u, v) \in E$ we have $\Pr(I_S(u) \neq I_S(v)) \leq \alpha D \cdot z(u, v)$, where I_S is the indicator function for the set S .

Below we present an efficient algorithm for generating an LP separator. The input to this algorithm is a graph $G = (V, E)$ that excludes $K_{r,r}$ minors, a parameter $\beta \in (0, 1)$, and numbers $\{x(u)\}_{u \in V}, \{z(u, v)\}_{u, v \in V}$ satisfying the first three constraints in LP (14) (namely, the triangle inequality even with the fictional vertex o). The algorithm then computes a sample of an LP

separator with distortion $O(r^2)$. For genus g graphs, the argument is similar and gives distortion $O(\log g)$.

Using this algorithm for LP separators, Theorem 1.7 is proved in the following way. Replace in Algorithm I and Algorithm II above the SDP relaxation (4) with the LP relaxation (14), and the orthogonal separators with LP separators. This provides an $O(r^2)$ approximation algorithm for Small-Set Expansion in $K_{r,r}$ excluded-minor graphs.

Computing LP Separators. We now describe an algorithm that samples an LP separator (see Definition 2.5) with respect to a feasible solution to LP (14). We recall a standard notion of low-diameter decomposition of a metric space, see e.g. [Bar96, GKL03, KR11] and references therein.

Let (V, d) be a finite metric space. Given a partition P of V and a point $v \in V$, we refer to the elements of P as *clusters*, and let $P(v)$ denote the unique cluster $S \in P$ that contains v . A *stochastic decomposition* of this metric is a probability distribution ν over partitions P of V .

Definition 2.6 (Separating Decomposition). *Let $D, \Delta > 0$. A stochastic decomposition ν of a finite metric space (V, d) is called a D -separating Δ -bounded decomposition if it satisfies:*

1. For every partition $P \in \text{supp}(\nu)$ and every cluster $S \in P$,

$$\text{diam}(S) := \max_{u, v \in S} d(u, v) \leq \Delta.$$

2. For every $u, v \in V$, the probability that a partition P sampled from ν separates them is

$$\Pr_{P \sim \nu} [P(u) \neq P(v)] \leq D \cdot \frac{d(u, v)}{\Delta}.$$

Theorem 2.4 ([KPR93, Rao99, FT03]). *Let $G = (V, E)$ be a graph excluding $K_{r,r}$ as a minor, equipped with nonnegative edge-lengths and let $\Delta > 0$. Then there exist an $O(r^2)$ -separating Δ -bounded decomposition ν of the shortest-path metric d_G on G and a polynomial-time algorithm that samples a partition from ν .*

Lee and Sidiropoulos [LS10] similarly show that graphs with genus $g \geq 1$ admit an $O(\log g)$ -separating decomposition. Alternative algorithms for both cases are shown in [KR11].

Theorem 2.5. *There exists an algorithm that, given a graph $G = (V, E)$ that excludes $K_{r,r}$ minors, numbers $\{x(u)\}_{u \in V}$, $\{z(u, v)\}_{u, v \in V}$ satisfying the first three constraints of LP (14), and a parameter $\beta \in (0, 1)$, returns an LP separator with distortion $D = O(r^2/\beta)$, probability scale $\alpha = \Omega(1/|V|)$ and separation threshold β .*

Proof. We slightly modify the graph G by adding a new vertex o and edges $\{(o, v) : v \in V\}$; let $V' = V \cup \{o\}$ and $E' = E \cup \{(o, v) : v \in V\}$. Note that the modified graph (V', E') excludes minor $K_{r+1, r+1}$, so we can still apply Theorem 2.4. We set the length of every edge $(u, v) \in E'$ to be

$$\ell(u, v) := \begin{cases} z(u, v) & \text{if } u, v \in V; \\ x(u) & \text{if } u \in V \text{ and } v = o. \end{cases}$$

For ease of notation, let the modified graph (V', E') be also called G , and let d_G denote the shortest-path metric according to these edge-lengths. It follows from the LP's triangle inequalities that $z(u, v) \leq d_G(u, v)$ for all $u, v \in V'$. The following algorithm outputs a subset $S \subset V$ which is an LP separator.

1. Choose $\delta \in [0, 1]$ uniformly at random, and sample a random partition P_δ from Theorem 2.4 applied to the metric (V', d_G) and parameter $\Delta = \beta\delta/2$.
2. Let $S_1, \dots, S_t \in P_\delta$ denote the clusters that contain at least one vertex $u \in V$ with $x(u) \in [\delta, 2\delta]$. Clearly $t \leq |V|$.
3. Output cluster $S = S_i$ with probability $1/|V|$ for each $i \in \{1, \dots, t\}$; with the remaining probability $1 - t/|V|$ output the empty cluster $S = \emptyset$.

Let us show that the output cluster S is an LP separator with $\alpha = 1/(2|V|)$ and $D = O(r^2/\beta)$. For the first requirement from an LP separator, we see that for every $u \in V$,

$$\Pr[u \in S] \geq \Pr\left[\frac{x(u)}{2} \leq \delta \leq x(u)\right] \cdot \frac{1}{|V|} = \frac{x(u)}{2|V|} = \alpha \cdot x(u).$$

Observe that conditioned on δ , if the output cluster $S \neq \emptyset$ then there is some $w \in S$ with $x(w) \in [\delta, 2\delta]$ by Step 2. Since the diameter of cluster S under the metric d_G is bounded by $\Delta = \beta\delta/2$, for all $v \in S$ we have $|x(w) - x(v)| \leq z(w, v) \leq d_G(w, v) \leq \beta\delta/2$, and therefore

$$(1 - \beta/2)\delta \leq \min_{v \in S} x(v) \leq \max_{v \in S} x(v) \leq (2 + \beta/2)\delta. \quad (15)$$

For the second requirement from an LP separator, consider $u, v \in V$ with $z(u, v) \geq \beta \cdot \min\{x(u), x(v)\}$. Assume without loss of generality that $x(u) \leq x(v)$, and then $d_G(u, v) \geq z(u, v) \geq \beta \cdot \min\{x(u), x(v)\} = \beta \cdot x(u)$. For the event $\{u, v \in S\}$ we must have: (i) $(1 - \beta/2)\delta \leq x(u)$ from Inequality (15); and (ii) $d_G(u, v) \leq \Delta = \beta\delta/2$ by the diameter bound on S . The former implies $\delta \leq \frac{x(u)}{1 - \beta/2} < 2x(u)$ since $\beta < 1$, while the latter implies $\delta \geq \frac{2}{\beta} \cdot d_G(u, v) \geq 2x(u)$. We conclude that $\Pr[u, v \in S] = 0$.

For the third requirement from an LP separator, we will bound $\Pr[u \in S, v \notin S]$ for an edge $(u, v) \in E$, the other (symmetric) case being identical. Using Inequality (15),

$$\Pr[u \in S \mid \delta] \leq \begin{cases} \frac{1}{|V|}, & \text{if } \frac{x(u)}{2 + \beta/2} \leq \delta \leq \frac{x(u)}{1 - \beta/2}; \\ 0, & \text{otherwise.} \end{cases}$$

Using Bayes' rule, then the fact that P_δ is a separating decomposition, and then that $d_G(u, v) \leq \ell(u, v) = z(u, v)$ (because it is an edge, we in fact have equality),

$$\Pr[v \notin S \mid u \in S, \delta] \leq \Pr[P_\delta(u) \neq P_\delta(v) \mid \delta] \leq D_0 \frac{d_G(u, v)}{\beta\delta/2} \leq \frac{2D_0 \cdot z(u, v)}{\beta\delta},$$

where $D_0 = O(r^2)$ is from Theorem 2.4. Combining the last two bounds, and recalling that $\delta \in [0, 1]$ uniformly at random and $\beta < 1$,

$$\Pr[u \in S, v \notin S] \leq \int_{x(u)/3}^{2x(u)} \frac{1}{|V|} \frac{2D_0 \cdot z(u, v)}{\beta\delta} d\delta = \frac{2 \ln 6 \cdot D_0 \cdot z(u, v)}{\beta|V|}.$$

We thus have

$$\Pr[I_S(u) \neq I_S(v)] \leq \frac{4 \ln 6 \cdot D_0 \cdot z(u, v)}{\beta|V|} = \alpha \frac{8 \ln 6 \cdot D_0}{\beta} z(u, v),$$

which proves the third requirement with $D = 8 \ln 6 \cdot D_0/\beta = O(r^2/\beta)$. \square

2.4 From Small-Set Expansion to ρ -Unbalanced Cut

ρ -Unbalanced Cut and Small-Set Expansion are related to each other, from the perspective of bicriteria approximation guarantees, in the same way that Balanced Cut and (uniform-demands) Sparsest Cut are. Our intended application of approximating Min-Max k -Partitioning (in Section 3), requires a weighted version of the ρ -Unbalanced Cut problem.

Definition 2.7 (Weighted ρ -Unbalanced Cut). *The input to this problem is a tuple $\langle G, y, w, \tau, \rho \rangle$, where $G = (V, E)$ is a graph with vertex-weights $y : V \rightarrow \mathbb{R}_+$, edge-weights $w : E \rightarrow \mathbb{R}_{\geq 0}$, and parameters $\tau, \rho \in (0, 1]$. The goal is to find $S \subseteq V$ of minimum weight (cost) $\delta(S)$ satisfying:*

1. $y(S) \geq \tau \cdot y(V)$; and
2. $|S| \leq \rho \cdot |V|$.

The unweighted version of the problem (defined in Section 1.2) has $\tau = \rho$ and unit vertex-weights, i.e. $y(v) = 1$ for all $v \in V$. We show that Weighted ρ -Unbalanced Cut can be reduced to Weighted Small-Set Expansion, which is needed for our intended application. Formally, we have the following corollary of Theorem 2.1, which immediately proves the unweighted version stated in Theorem 1.3. We use $\text{OPT}_{\langle G, y, w, \tau, \rho \rangle}$ to denote the optimal value (cost) of the corresponding weighted ρ -Unbalanced Cut instance.

Corollary 2.6 (Approximating Weighted ρ -Unbalanced Cut). *For every $\varepsilon > 0$, there exists a polynomial-time algorithm that given an instance $\langle G, y, w, \tau, \rho \rangle$ of Weighted ρ -Unbalanced Cut, finds a set $S \subseteq V$ satisfying:*

1. $|S| \leq \beta \rho n$, where $\beta = 1 + \varepsilon$ and $n = |V|$.
2. $y(S) \geq \tau \cdot y(V) / \gamma$, where $\gamma = O(1)$.
3. $\delta(S) \leq \alpha \cdot \text{OPT}_{\langle G, y, w, \tau, \rho \rangle}$ for $\alpha = O_\varepsilon(\sqrt{\log n \log(\max(1/\rho, 1/\tau))})$.

Proof. Let S^* be an optimal solution to $\langle G, y, w, \tau, \rho \rangle$, thus $|S^*| \leq \rho n$, $y(S^*) \geq \tau \cdot y(V)$ and $\delta(S^*) = \text{OPT}_{\langle G, y, w, \tau, \rho \rangle}$. Define two measures μ and η on V as follows. For every $S \subseteq V$, set $\mu(S) := |S|/n$ and $\eta(S) := y(S)/y(V)$.

The algorithm guesses $H \geq \tau$ such that $H \leq \eta(S^*) \leq 2H$ (see Algorithm I above for an argument why we can guess H). Then it invokes Theorem 2.1 part II on G with the measures μ and η defined above, and parameters ρ, H . The obtained solution S satisfies

$$|S| = \mu(S) \cdot n \leq (1 + \varepsilon) \rho n,$$

and

$$y(S) = \eta(S) \cdot y(V) \geq \Omega(H) \cdot y(V) \geq \Omega(\tau) \cdot y(V).$$

Furthermore,

$$\delta(S) \leq \eta(S) \cdot [\alpha \cdot \delta(S^*) / \eta(S^*)] \leq \alpha \cdot 2(1 + \varepsilon) \cdot \delta(S^*),$$

where $\alpha = O_\varepsilon(\sqrt{\log n \log(\max(1/\rho, 1/\tau))})$. □

3 Min-max Balanced Partitioning

In this section, we present our algorithm for Min-Max k -Partitioning, assuming a subroutine that approximates Weighted ρ -Unbalanced Cut. Our algorithm for Min-Max k -Partitioning follows by a straightforward composition of Theorem 3.1 and Theorem 3.3 below. Plugging in for (α, β, γ) the values obtained in Section 2 would complete the proof of Theorem 1.1.

3.1 Uniform Coverings

We first consider a covering relaxation of Min-Max k -Partitioning and solve it using multiplicative updates. This covering relaxation can alternatively be viewed as a fractional solution to a configuration LP of exponential size, as discussed further below. We include the analysis details for the multiplicative updates for completeness.

Let $\mathcal{C} = \{S \subseteq V : |S| \leq n/k, \delta(S) \leq \text{OPT}\}$ denote all the vertex-sets that are feasible for a single part. Note that a feasible solution in Min-Max k -Partitioning corresponds to a partition of V into k parts, where each part belongs to \mathcal{C} . Algorithm 1, *uniformly covers* V , where the exact meaning of uniformly covers is stated in Theorem 3.1, using sets in \mathcal{C} (actually a slightly larger family than \mathcal{C} since our approximation for Weighted ρ -Unbalanced Cut is bicriteria). It is important to note that its output \mathcal{S} is a multiset.

Algorithm 1: Covering Procedure for Min-Max k -Partitioning

```

Set  $t = 1$ ,  $\mathcal{S} = \emptyset$ , and  $y^1(v) = 1$  for all  $v \in V$ .
while  $\sum_{v \in V} y^t(v) > 1/n$  do
    // Solve the following using the algorithm from Corollary 2.6.
    Let  $S^t \subseteq V$  be the solution for the Weighted  $\rho$ -Unbalanced Cut instance  $\langle G, y^t, w, \frac{1}{k}, \frac{1}{k} \rangle$ .
    Add  $S^t$  to  $\mathcal{S}$ .
    // Update the weights of the covered vertices.
    for every  $v \in V$  do
         $\lfloor$  Set  $y^{t+1}(v) = \frac{1}{2} \cdot y^t(v)$  if  $v \in S^t$ , and  $y^{t+1}(v) = y^t(v)$  otherwise.
     $\rfloor$ 
    Set  $t = t + 1$ .
return  $\mathcal{S}$ .

```

The algorithm above is a multiplicative update type algorithm. Initially each vertex is uncovered and is assigned a weight of 1. Whenever a vertex is covered by some set S , its weight is halved, and the algorithm continues as long as the total weight exceeds $1/n$. At each time step, these weights are used as input to the Weighted ρ -Unbalanced Cut instance, which guides the set S to cover those vertices that have not been covered enough times thus far.

Theorem 3.1. *Running Algorithm 1 on an instance of Min-Max k -Partitioning outputs a multiset \mathcal{S} that satisfies the following conditions:*

1. For all $S \in \mathcal{S}$:

$$\delta(S) \leq \alpha \cdot \text{OPT} \quad \text{and} \quad |S| \leq \beta \cdot n/k,$$

where OPT denotes the optimal value of the Min-Max k -Partitioning instance.

2. For all $v \in V$:

$$\frac{|\{S \in \mathcal{S} : v \in S\}|}{|\mathcal{S}|} \geq \frac{1}{(5\gamma k)}.$$

Above, $\alpha = O_\varepsilon(\sqrt{\log n \log k})$, $\beta = 1 + \varepsilon$ and $\gamma = O_\varepsilon(1)$, for any fixed $\varepsilon > 0$.

Proof. For an iteration t , let us denote $Y^t := \sum_{v \in V} y^t(v)$. The first assertion of the theorem is immediate from the following claim.

Claim 3.2. *Every iteration t of Algorithm 1 satisfies: $\delta(S^t) \leq \alpha \cdot \text{OPT}$ and $|S^t| \leq \beta \cdot n/k$.*

Proof. We claim that no matter what the values of y^t 's are, the optimal solution to the Weighted ρ -Unbalanced Cut instance $\langle G, y^t, w, \frac{1}{k}, \frac{1}{k} \rangle$ is at most OPT . To see this, consider the optimal solution $\{S_i^*\}_{i=1}^k$ of the original Min-Max k -Partitioning instance. We have $|S_i^*| \leq n/k$ and $w(\delta(S_i^*)) \leq \text{OPT}$ for all $i \in [k]$. Since $\{S_i^*\}_{i=1}^k$ partitions V , there is some $j \in [k]$ with $y^t(S_j^*) \geq Y^t/k$. It now follows that S_j^* is a feasible solution to the Weighted ρ -Unbalanced Cut instance $\langle G, y^t, w, \frac{1}{k}, \frac{1}{k} \rangle$, with objective value at most OPT , which proves the claim. \square

We proceed to prove the second assertion of Theorem 3.1. Let ℓ denote the number of iterations of the while loop, for the given Min-Max k -Partitioning instance. For any $v \in V$, let N_v denote the number of iterations t in which $v \in S^t$. Then, by the updating rule for the y weights we have that $y^{\ell+1}(v) = 1/2^{N_v}$. Moreover, the termination condition implies that $y^{\ell+1}(v) \leq 1/n$ (since $Y^{\ell+1} \leq 1/n$). Thus we, obtain that $N_v \geq \log_2 n$ for all $v \in V$. From the approximation guarantee of the Weighted ρ -Unbalanced Cut algorithm (Corollary 2.6), it follows that $y^t(S^t) \geq \frac{1}{\gamma k} \cdot Y^t$ in every iteration t . Thus,

$$Y^{t+1} = Y^t - \frac{1}{2} \cdot y^t(S^t) \leq \left(1 - \frac{1}{2\gamma k}\right) \cdot Y^t.$$

This, in turn, implies that:

$$Y^\ell \leq \left(1 - \frac{1}{2\gamma k}\right)^{\ell-1} \cdot Y^1 = \left(1 - \frac{1}{2\gamma k}\right)^{\ell-1} \cdot n.$$

However, $Y^\ell > 1/n$ since the algorithm performs ℓ iterations. Thus, $\ell \leq 1 + 4\gamma k \cdot \ln n \leq 5\gamma k \cdot \log_2 n$. This finished the proof since:

$$|\{S \in \mathcal{S} : S \ni v\}|/|\mathcal{S}| = N_v/\ell \geq (5\gamma)^{-1} k^{-1}.$$

\square

Alternative View: A configuration LP. We now describe an alternate approach to finding a cover \mathcal{S} . Given a bound λ on the cost of any single cut, define the set of feasible cuts as follows:

$$\mathcal{F}_\lambda = \left\{ S \subseteq V : |S| \leq \frac{n}{k}, \delta(S) \leq \lambda \right\}.$$

We define a configuration LP for Min-Max k -Partitioning as follows. There is a variable x_S for each $S \in \mathcal{F}_\lambda$ indicating whether the cut S is chosen or not. The only constraint we have is that each vertex v belongs to at least one feasible cut.

$$\boxed{\begin{array}{ll} \mathcal{P}(\lambda) = \min & \sum_{S \in \mathcal{F}_\lambda} x_S \\ \text{s.t.} & \sum_{S \in \mathcal{F}_\lambda : v \in S} x_S \geq 1 \quad \forall v \in V \\ & x_S \geq 0 \quad \forall S \in \mathcal{F}_\lambda \end{array}} \quad (16)$$

The goal is determine the smallest $\lambda > 0$ such that $\mathcal{P}(\lambda) \leq k$. Since the configuration LP (16) has an exponential number of variables, we formulate its dual and present a separation oracle for it.

$$\boxed{\begin{array}{ll} \mathcal{D}(\lambda) = \max & \sum_{v \in V} y_v \\ \text{s.t.} & \sum_{v \in S} y_v \leq 1 \quad \forall S \in \mathcal{F}_\lambda \\ & y_v \geq 0 \quad \forall v \in V \end{array}} \quad (17)$$

One can (approximately) solve the dual formulation since a separation oracle which uses Weighted Small-Set Expansion exists. The details for approximating the configuration LP are rather technical (since we only have a bicriteria approximation for Weighted Small-Set Expansion, as opposed to a usual multiplicative approximation) and do not offer any advantage over the multiplicative updates method except for an alternative intuition regarding the issue. Thus, they are omitted.

3.2 Aggregation

The aggregation process, which might be of independent interest, transforms a cover of G into a partition, without violating the min-max objective by much. In particular, we show the following.

Theorem 3.3. *There is a randomized polynomial-time algorithm (Algorithm 2 below) that when given a graph $G = (V, E)$, an $\varepsilon \in (0, 1)$, and a cover \mathcal{S} consisting of subsets of V such that:*

1. *every vertex in V is covered by at least a fraction c/k of sets $S \in \mathcal{S}$ for some $c \in (0, 1]$, and*
2. *each $S \in \mathcal{S}$ satisfies $|S| \leq 2n/k$ and $\delta(S) \leq B$,*

outputs a partition \mathcal{P} of V into at most k sets such that for all $P \in \mathcal{P}$ we have $|P| \leq 2(1 + \varepsilon)n/k$ and $\mathbb{E}[\max_{P \in \mathcal{P}} \delta(P)] \leq 8B/(c\varepsilon)$.

We remark that this process is comprised of three different steps. Intuitively, we first let the sets randomly compete with each other over the vertices so as to form a partition; then, to make sure no set has large cost, we repeatedly fix the partition locally, and use a potential function to track progress. Finally, the sets need to be aggregated while ensuring that their cost and sizes do not increase much, such that we are left with at most k cuts.

Note that in step 2, whenever $\delta(P_i)$ exceeds $2B$, we replace P_i with S_i , and adjust the other sets P_j accordingly to ensure that we maintain a partition. In step 3, we aggregate the pieces P_i to reduce the number of parts to at most k .

Analysis: We now prove that this algorithm satisfies the properties stated in theorem 3.3.

Proof. (Theorem 3.3) We start by analyzing step 1. Observe that after step 1 the collection of sets $\{P_i\}$ is a partition of V and $P_i \subseteq S_i$ for every i . Particularly, $|P_i| \leq |S_i| \leq 2n/k$. Note, however, that the bound $\delta(P_i) \leq B$ may be violated for some i since P_i might be a strict subset of S_i .

We finish the analysis of step 1 of Algorithm 2 by proving that $\mathbb{E}[\sum_i \delta(P_i)] \leq 2kB/c$. Fix an $i \leq |\mathcal{S}|$ and estimate the expected weight of edges $E(P_i, \cup_{j>i} P_j)$ given that $S_i = S$. If an edge (u, v) belongs to $E(P_i, \cup_{j>i} P_j)$ then $(u, v) \in E(S_i, V \setminus S_i) = E(S, V \setminus S)$ and both $u, v \notin \cup_{j<i} S_j$. For any edge $(u, v) \in \delta(S)$ (with $u \in S, v \notin S$), $\Pr((u, v) \in E(P_i, \cup_{j>i} P_j) \mid S_i = S) \leq \Pr(v \notin$

Algorithm 2: Aggregation Procedure for Min-Max k -Partitioning

1 // Step 1: Sampling
 Sort sets in \mathcal{S} in a random order: $S_1, S_2, \dots, S_{|\mathcal{S}|}$. Let $P_i = S_i \setminus (\cup_{j < i} S_j)$.
2 // Step 2: Replacing Expanding Sets with Sets from \mathcal{S}
 while there is a set P_i such that $\delta(P_i) > 2B$ **do**
 Set $P_i = S_i$, and for all $j \neq i$, set $P_j = P_j \setminus S_i$.
3 // Step 3: Aggregating
 Let $B' = \max\{\frac{1}{k} \sum_i \delta(P_i), 2B\}$.
 while there are $P_i \neq \emptyset, P_j \neq \emptyset$ ($i \neq j$) such that $|P_i| + |P_j| \leq 2(1 + \varepsilon)n/k$ and
 $\delta(P_i) + \delta(P_j) \leq 2B'\varepsilon^{-1}$ **do**
 Set $P_i = P_i \cup P_j$ and set $P_j = \emptyset$.
4 return all non-empty sets P_i .

$\cup_{j < i} S_j \mid S_i = S) \leq (1 - c/k)^{i-1}$, since v is covered by at least c/k fraction of sets in \mathcal{S} and is not covered by $S_i = S$. Hence,

$$\mathbb{E}[w(E(P_i, \cup_{j > i} P_j)) \mid S_i = S] \leq (1 - c/k)^{i-1} \delta(S) \leq (1 - c/k)^{i-1} B,$$

and $\mathbb{E}[w(E(P_i, \cup_{j > i} P_j))] \leq (1 - c/k)^{i-1} B$. Therefore, the total expected weight of edges crossing the boundary of the P_i 's is at most $\sum_{i=0}^{\infty} (1 - c/k)^i B = kB/c$, and $\mathbb{E}[\sum_i \delta(P_i)] \leq 2kB/c$. This finishes the analysis of step 1.

Let us analyze now step 2. Our potential function is $\sum_i \delta(P_i)$, whose initial value was bounded in step 1. We prove that this potential function reduces quickly over the iterations of step 2, thus, step 2 terminates after a ‘‘small’’ number of steps. After each iteration of step 2, the following invariant holds: the collection of sets $\{P_i\}$ is a partition of V and $P_i \subseteq S_i$ for all i . Particularly, $|P_i| \leq |S_i| \leq 2n/k$. The key observation is that at every iteration of the ‘‘while’’ loop, the sum $\sum_j \delta(P_j)$ decreases by at least $2B$. This is due to the following uncrossing argument:

$$\begin{aligned}
 \delta(S_i) + \sum_{j \neq i} \delta(P_j \setminus S_i) &\leq \delta(S_i) + \sum_{j \neq i} \left(\delta(P_j) + w(E(P_j \setminus S_i, S_i)) - w(E(S_i \setminus P_j, P_j)) \right) \\
 &\leq \delta(S_i) + \left(\sum_{j \neq i} \delta(P_j) \right) + \underbrace{w(E(V \setminus S_i, S_i))}_{\delta(S_i)} - \underbrace{w(E(P_i, V \setminus P_i))}_{\delta(P_i)} \\
 &= \left(\sum_j \delta(P_j) \right) + 2\delta(S_i) - 2\delta(P_i) \\
 &\leq \left(\sum_j \delta(P_j) \right) - 2B.
 \end{aligned}$$

The above inequalities use the facts that $P_i \subseteq S_i$ for all i and that all the P_j 's are disjoint. Hence, the number of iterations of the loop in step 2 is always polynomially bounded and after the last iteration $\mathbb{E}[\sum_i \delta(P_i)] \leq 2kB/c$ (the expectation is over random choices at step 1; step 2 does not use random bits). Hence, $\mathbb{E}[B'] \leq 4B/c$; recall that $B' = \max\{\frac{1}{k} \sum_i \delta(P_i), 2B\}$.

Let us now analyze step 3. The following analysis holds conditional on any value of B' . After each iteration of step 3, the following invariant holds: the collection of sets $\{P_i\}$ is a partition of

V , and for all i , $|P_i| \leq 2(1 + \varepsilon)n/k$ and $\delta(P_i) \leq 2B'\varepsilon^{-1}$. Clearly, after step 2, $\delta(P_i) \leq 2B \leq B'$ and $|P_i| \leq 2n/k$ for all i . Moreover, the sum $\sum_i \delta(P_i)$ can only decrease in step 3.

When the loop terminates, we obtain a partition of V into sets P_i satisfying $|P_i| \leq 2(1 + \varepsilon)n/k$, $\sum_i |P_i| = n$, $\delta(P_i) \leq 2B'\varepsilon^{-1}$, $\sum_i \delta(P_i) \leq kB'$, such that no two sets can be merged without violating above constraints. Hence, by Lemma 3.4 below (with $a_i = |P_i|$ and $b_i = \delta(P_i)$), the number of non-empty sets is at most

$$2 \frac{n}{2(1 + \varepsilon)n/k} + \frac{kB'}{2B'\varepsilon^{-1}} = (1 + \varepsilon)^{-1}k + (\varepsilon/2)k \leq k.$$

This finishes the proof. \square

Lemma 3.4 (Greedy Aggregation). *Let a_1, \dots, a_t and b_1, \dots, b_t be two sequences of nonnegative numbers satisfying the following constraints $a_i < A$, $b_i < B$, $\sum_{i=1}^t a_i \leq S$ and $\sum_{i=1}^t b_i \leq T$ (for some positive real numbers A , B , S , and T). Moreover, assume that for every i and j ($i \neq j$) either $a_i + a_j > A$ or $b_i + b_j > B$. Then,*

$$t < \frac{S}{A} + \frac{T}{B} + \max \left\{ \frac{S}{A}, \frac{T}{B}, 1 \right\}.$$

Proof. By rescaling we assume that $A = 1$ and $B = 1$. Moreover, we may assume that $\sum_{i=1}^t a_i < S$ and $\sum_{i=1}^t b_i < T$ by slightly decreasing the values of all a_i and b_i so that all inequalities still hold.

We write two linear programs:

$$\begin{aligned} (LP_I) \quad & \min && \sum_i x_i \\ & \text{s.t.} && x_i + x_j \geq 1 && \forall (i, j) \in \{(i, j) : a_i + a_j \geq 1\} \\ & && x_i \geq 0 && \forall i \end{aligned}$$

$$\begin{aligned} (LP_{II}) \quad & \min && \sum_i y_i \\ & \text{s.t.} && y_i + y_j \geq 1 && \forall (i, j) \in \{(i, j) : b_i + b_j \geq 1\} \\ & && y_i \geq 0 && \forall i \end{aligned}$$

The first LP (LP_I) has variables x_i and constraints $x_i + x_j \geq 1$ for all i, j such that $a_i + a_j \geq 1$. The second LP (LP_{II}) has variables y_i and constraints $y_i + y_j \geq 1$ for all i, j such that $b_i + b_j \geq 1$. The LP objectives are to minimize $\sum_i x_i$ and to minimize $\sum_i y_i$. Note, that $\{a_i\}$ is a feasible point for LP_I and $\{b_i\}$ is a feasible point for LP_{II} . Thus, the optimum values of LP_I and LP_{II} are strictly less than S and T respectively.

Observe that both LPs are half-integral. Consider optimal solutions x_i^*, y_j^* where $x_i^*, y_j^* \in \{0, 1/2, 1\}$. Note that for every i, j either $x_i^* + x_j^* \geq 1$ or $y_i^* + y_j^* \geq 1$. Consider several cases. If for all i , $x_i^* + y_i^* \geq 1$, then $t < S + T$, since $\sum_{i=1}^t (x_i^* + y_i^*) < S + T$. If for some j , $x_j^* + y_j^* = 0$ (and hence $x_j^* = y_j^* = 0$), then $x_i^* + y_i^* \geq 1$ for $i \neq j$ and, thus, $t < S + T + 1$. Finally, assume that for some j , $x_j^* + y_j^* = 1/2$, and w.l.o.g. $x_j^* = 1/2$ and $y_j^* = 0$. The number of i 's with $x_i^* \neq 0$ is (strictly) bounded by $2S$. For the remaining i 's, $x_i^* = 0$ and hence $y_i^* = 1$ (because $y_i^* = x_i^* + y_j^* \geq 1$), and thus the number of such i 's is (strictly) bounded by T . This finishes the proof of the lemma. \square

4 Further Extensions

Both Theorems 1.1 and 1.4 follow from a more general result for a problem that we call Min–Max Cut. This problem models the aforementioned cloud computing scenario, where in addition, certain processes are preassigned to machines (each set T_i maps to machine $i \in [k]$). The goal is to assign the processes V to machines $[k]$ while respecting the pre-assignment and machine load constraints, and minimizing both bandwidth per machine and total volume of communication. It is clear that in fact Theorem 1.6 generalizes both Theorems 1.1 and 1.4. Let us now describe modifications to the Min–Max k -Partitioning algorithm used to obtain Theorem 1.6.

Uniform Coverings. First, by the introduction of vertex weights, we can shrink each preassigned set T_i to a single terminal t_i (for $i \in [k]$). Then, feasible vertex-sets \mathcal{C} in the covering procedure (Section 3.1) consist of those $S \subseteq V$ where $\text{weight}(S) \leq \rho n$ (balance constraint) and $|S \cap \{t_i\}_{i=1}^k| \leq 1$ (preassignment constraint). The subproblem Weighted ρ -Unbalanced Cut also has the additional $|S \cap \{t_i\}_{i=1}^k| \leq 1$ constraint; this can be handled in the algorithm from Section 2 by guessing which terminal belongs to S (see Remark 2.4). Using Corollary 2.6 we assume an (α, β, γ) approximation algorithm for this (modified) Weighted ρ -Unbalanced Cut problem; where for any $\varepsilon > 0$, $\alpha = O_\varepsilon(\sqrt{\log n \log(\max\{1/\rho, 1/\tau\})})$, $\beta = 1 + \varepsilon$ and $\gamma = O_\varepsilon(1)$.

Algorithm 3 below gives the procedure to obtain a uniform covering \mathcal{S} bounding total edge-cost in addition to the conditions in Theorem 3.1.

Algorithm 3: Covering Procedure for Min–Max Cut:

```

Set  $t \leftarrow 1$ ,  $\mathcal{S} \leftarrow \emptyset$ ,  $y^1(v) \leftarrow 1$  for all  $v \in V$ , and  $Y^1 \leftarrow \sum_{v \in V} y^1(v)$ .
while  $Y^t > \frac{1}{n}$  do
  for  $i = 0, \dots, \log_2 k + 1$  do
    Solve the Weighted  $\rho$ -Unbalanced Cut instance  $\langle G, y^t, w, \frac{1}{2^i}, \rho \rangle$  using the algorithm
    from Corollary 2.6, to obtain  $S^t(i) \subseteq V$ .
    If  $(\delta(S^t(i)) \leq \alpha \cdot \min\{C, 4D/2^i\})$  then  $S^t \leftarrow S^t(i)$  and quit this loop.
  Add  $S^t$  to  $\mathcal{S}$ .
  for  $v \in V$  do
    Set  $y^{t+1}(v) \leftarrow \frac{1}{2} \cdot y^t(v)$  if  $v \in S^t$ , and  $y^{t+1}(v) \leftarrow y^t(v)$  otherwise.
  Set  $Y^{t+1} \leftarrow \sum_{v \in V} y^{t+1}(v)$ .
  Set  $t \leftarrow t + 1$ .
return  $\mathcal{S}$ .

```

Theorem 4.1. *For any instance of Min–Max Cut, output \mathcal{S} of Algorithm 3 satisfies:*

1. $\delta(S) \leq \alpha \cdot C$ and $|S| \leq \beta \cdot \frac{n}{k}$ for all $S \in \mathcal{S}$.
2. $|\{S \in \mathcal{S} : v \in S\}| \geq \log_2 n$ for all $v \in V$.
3. $|\mathcal{S}| \leq 5\gamma k \cdot \log_2 n$.
4. $\sum_{S \in \mathcal{S}} \delta(S) \leq 17\alpha\gamma \log_2 n \cdot D$.

Above, for any $\varepsilon > 0$, $\alpha = O_\varepsilon(\sqrt{\log n \log k})$, $\beta = 1 + \varepsilon$ and $\gamma = O_\varepsilon(1)$.

Proof. We start with the following key claim.

Claim 4.2. *In any iteration t of Algorithm 3, there exists an $i \in \{0, 1, \dots, \log_2 k + 1\}$ such that $\delta(S^t(i)) \leq \alpha \cdot \min\{C, 4D/2^i\}$, $|S^t(i)| \leq \beta \cdot \frac{n}{k}$, and $y^t(S^t(i)) \geq \frac{Y^t}{\gamma 2^i}$.*

Proof. Consider the optimal solution $\{S_j^*\}_{j=1}^k$ of the original Min–Max Cut instance. For all $j \in [k]$ we have that $|S_j^*| \leq \rho n$, $\delta(S_j^*) \leq C$ and S_j^* contains at most one terminal. Moreover, $\sum_{j=1}^k \delta(S_j^*) \leq D$. Since $\{S_j^*\}_{j=1}^k$ partitions V , we also have $\sum_{j=1}^k y^t(S_j^*) = Y^t$. Let $L \subseteq [k]$ denote the indices j having $\delta(S_j^*) \leq \frac{2D}{Y^t} \cdot y^t(S_j^*)$.

We claim that $\sum_{j \in L} y^t(S_j^*) \geq Y^t/2$. This is true since:

$$\sum_{j \notin L} y^t(S_j^*) \leq \frac{Y^t}{2D} \cdot \sum_{j \notin L} \delta(S_j^*) \leq \frac{Y^t}{2D} \cdot \sum_{j=1}^k \delta(S_j^*) \leq Y^t/2.$$

Since $|L| \leq k$, there is some $q \in L$ with $y^t(S_q^*) \geq \frac{Y^t}{2k}$. Let $i \in \{1, \dots, \log_2 k + 1\}$ be the value such that $y^t(S_q^*)/Y^t \in [\frac{1}{2^i}, \frac{1}{2^{i-1}}]$; note that such an i exists because $y^t(S_q^*)/Y^t \in [\frac{1}{2k}, 1]$. For this i , consider the Weighted ρ -Unbalanced Cut instance $\langle G, y^t, w, 2^{-i}, \rho \rangle$. Observe that S_q^* is a feasible solution here since $y^t(S_q^*) \geq Y^t/2^i$, $|S_q^*| \leq \rho n$ and S_q^* contains at most one terminal. Hence, the optimal value of this instance is at most:

$$\delta(S_q^*) \leq \min \left\{ C, \frac{2D}{Y^t} \cdot y^t(S_q^*) \right\} \leq \min \left\{ C, \frac{4D}{2^i} \right\}.$$

The first inequality uses the definition of L and that $\delta(S_q^*) \leq C$, and the second inequality is by the choice of i . It now follows from Corollary 2.6 that the solution $S^t(i)$ satisfies the claimed properties. We note that $\alpha = O_\varepsilon(\sqrt{\log n \log k})$ because each instance of Weighted ρ -Unbalanced Cut has parameters $\tau = 2^{-i} \geq 1/k$ and $\rho \geq 1/k$. \square

For each iteration t , let $i_t \in \{1, \dots, \log_2 k + 1\}$ be the index such that $S^t = S^t(i_t)$. Claim 4.2 implies that such an index always exists, and so the algorithm is indeed well-defined. **Condition 1** of Theorem 4.1 also follows directly. Let ℓ denote the number of iterations of the while loop, for the given Min–Max Cut instance. For any $v \in V$, let N_v denote the number of iterations t in which $v \in S^t$. Then, by the update rule of the weights y we have $y^{\ell+1}(v) = 1/2^{N_v}$. Moreover, the termination condition implies that $y^{\ell+1}(v) \leq 1/n$ (since $Y^{\ell+1} \leq 1/n$). Thus we obtain $N_v \geq \log_2 n$ for all $v \in V$, proving **condition 2** of Theorem 4.1.

Claim 4.2 implies that for each iteration t , we have $y^t(S^t) \geq \frac{Y^t}{\gamma 2^{i_t}}$ and $\delta(S^t) \leq 4\alpha D/2^{i_t}$. Since $i_t \leq \log_2 k + 1$, we also obtain that:

$$y^t(S^t) \geq \max \left\{ \frac{Y^t}{2\gamma k}, \frac{Y^t}{4\alpha\gamma D} \cdot \delta(S^t) \right\}.$$

We need to prove that **condition 3** and **condition 4** also hold. First, using the fact that $y^t(S^t) \geq Y^t/(2\gamma k)$ in each iteration, we can deduce that:

$$Y^{t+1} = Y^t - \frac{1}{2} \cdot y^t(S^t) \leq \left(1 - \frac{1}{4\gamma k}\right) \cdot Y^t.$$

This, in turn, implies that

$$Y^\ell \leq \left(1 - \frac{1}{4\gamma k}\right)^{\ell-1} \cdot Y^1 = \left(1 - \frac{1}{4\gamma k}\right)^{\ell-1} \cdot n.$$

However, $Y^\ell > 1/n$ since the algorithm performs ℓ iterations. Thus, $\ell \leq 1 + 8\gamma k \cdot \ln n \leq 9\gamma k \cdot \log_2 n$. This proves **condition 3** in Theorem 4.1.

Second, using $y^t(S^t) \geq \frac{Y^t}{4\alpha\gamma D} \cdot \delta(S^t)$ in each iteration, $Y^{t+1} = Y^t - \frac{1}{2} \cdot y^t(S^t) \leq \left(1 - \frac{\delta(S^t)}{8\alpha\gamma D}\right) \cdot Y^t$. So,

$$\frac{1}{n} < Y^\ell \leq \prod_{t=1}^{\ell-1} \left(1 - \frac{\delta(S^t)}{8\alpha\gamma D}\right) \cdot Y^1 \leq \exp\left(-\frac{\sum_{t=1}^{\ell-1} \delta(S^t)}{8\alpha\gamma D}\right) \cdot n$$

This implies:

$$\sum_{t=1}^{\ell-1} \delta(S^t) \leq (16\alpha\gamma \ln n) \cdot D.$$

Adding in $\delta(S^\ell) \leq \alpha \cdot C \leq \alpha D$, we obtain **condition 4** of Theorem 4.1. This completes the proof of Theorem 4.1. \square

Aggregation. This step remains essentially the same as in Section 3.2, namely Algorithm 2 (with parameter $B := \alpha \cdot C$). The only difference is that in Step 3 we do not merge parts containing terminals. We first show that this yields a slightly weaker version of Theorem 1.6: in condition (ii) we obtain a bound of $(3 + \varepsilon)\rho n$ on the cardinality of each part. (Later we show how to achieve the cardinality bound of $(2 + \varepsilon)\rho n$ as claimed in Theorem 1.6.)

Note that each of the final sets $\{P_i\}$ is a subset of some set in \mathcal{S} , and hence contains at most one terminal. It also follows that the final sets $\{P_i\}$ are at most $2k$ in number: at most k of them contain no terminals (just as in Theorem 3.3), and at most k contain a terminal (since there are at most k terminals). Each of these sets $\{P_i\}$ has size at most $(2 + \varepsilon)\rho n$ and cut value at most $8B/(c\varepsilon)$, by the analysis in Theorem 1.6. Moreover, if a set P_i contains a terminal then $|P_i| \leq \beta \cdot \rho n = (1 + \varepsilon)\rho n$ (since it does not participate in any merge). Finally in order to reduce the number of parts to k , we merge arbitrarily each part containing a terminal with one non-terminal part; and output this as the final solution. It is clear that each part has at most one terminal, has size $\leq (3 + \varepsilon)\rho n$, and cut value at most $O_\varepsilon(\sqrt{\log n \log k}) \cdot C$. The bound on total cost (condition (iv) in Theorem 1.6) is by the following claim. This proves a weaker version of Theorem 1.6, with size bound $(3 + \varepsilon)\rho n$.

Claim 4.3. *Algorithm 2 applied on collection \mathcal{S} from Theorem 4.1 outputs partition $\{P_i\}_{i=1}^k$ satisfying $\mathbb{E}\left[\sum_{i=1}^k w(\delta(P_i))\right] = O_\varepsilon(\sqrt{\log n \log k}) D$.*

Proof. We will show that the random partition $\{P_i\}$ at the end of Step 1 in Algorithm 2 satisfies $\mathbb{E}\left[\sum_i w(\delta(P_i))\right] \leq O_\varepsilon(\sqrt{\log n \log k}) D$. This would suffice, since $\sum_i w(\delta(P_i))$ does not increase in Steps 2 and 3. For notational convenience, we assume (by adding empty sets) that $|\mathcal{S}| = 5\gamma k \cdot \log_2 n$ in Theorem 4.1; note that this does not affect any of the other conditions.

To bound the cost of the partition $\{P_i\}$ in Step 1, consider any index $i \leq |\mathcal{S}|$. From the proof of Theorem 3.3, we have:

$$\mathbb{E}[w(E(P_i, \cup_{j>i} P_j)) \mid S_i = S] \leq (1 - c/k)^{i-1} \delta(S),$$

where $c = 1/(5\gamma)$ is such that each vertex lies in at least a fraction c/k of sets \mathcal{S} . We remove the conditioning,

$$\mathbb{E}[w(E(P_i, \cup_{j>i} P_j))] \leq (1 - c/k)^{i-1} \cdot \mathbb{E}[\delta(S_i)] = (1 - c/k)^{i-1} \cdot \left(\sum_{S \in \mathcal{S}} \delta(S)/|\mathcal{S}| \right),$$

where we used that S_i is a uniformly random set from \mathcal{S} . Therefore, the total edge-cost can be upper bounded as follows:

$$\begin{aligned} \mathbb{E} \left[\sum_i \delta(P_i) \right] &= 2 \cdot \sum_i \mathbb{E}[w(E(P_i, \cup_{j>i} P_j))] \\ &\leq \left(\sum_{i \geq 0} (1 - c/k)^i \right) \cdot \left(\sum_{S \in \mathcal{S}} \delta(S)/|\mathcal{S}| \right) \\ &= \frac{k}{c} \cdot \sum_{S \in \mathcal{S}} \delta(S)/|\mathcal{S}|. \end{aligned}$$

Using $\sum_{S \in \mathcal{S}} \delta(S) \leq 17\alpha\gamma \log_2 n \cdot D$ and $|\mathcal{S}| = 5\gamma k \cdot \log_2 n$ from Theorem 4.1, we can conclude that

$$\mathbb{E} \left[\sum_i \delta(P_i) \right] \leq \frac{17\alpha}{5c} D = O_\varepsilon(\sqrt{\log n \log k}) D,$$

since $\alpha = O_\varepsilon(\sqrt{\log n \log k})$ and $1/c = O_\varepsilon(1)$. □

Obtaining size bound of $(2 + \varepsilon)\rho n$. We now describe a modified aggregating step (replacing Step 3 of Algorithm 2) that yields the promised guarantee of Theorem 1.6. Given the uniform cover \mathcal{S} from Algorithm 3, run Steps 1 and 2 of Algorithm 2 (use $B = \alpha C$) to obtain parts $P_1, \dots, P_{|\mathcal{S}|}$. Then perform the following:

1. Set $B' := \max \{ \frac{1}{k} \sum_i \delta(P_i), 2B \}$.
2. While there are $P_i, P_j \neq \emptyset$ ($i \neq j$) such that $|P_i| + |P_j| \leq (1 + \varepsilon)\rho n$, $\delta(P_i) + \delta(P_j) \leq 2B'$ and $P_i \cup P_j$ does not contain a terminal: replace $P_i \leftarrow P_i \cup P_j$ and $P_j \leftarrow \emptyset$.
3. Sort the resulting non-empty sets P_1, \dots, P_t according to a non-increasing order of size.
4. Form $\lceil t/k \rceil$ groups where the j^{th} group consists of parts indexed between $(j-1)k+1$ and jk .
5. For each $i \in [k]$ define Q_i as the union of one part from each group such that it contains terminal i but no other terminal. Additionally, ensure that each part is assigned to one of $\{Q_i\}_{i=1}^k$.

We first show that the number of parts after Step 2 above is $t \leq 4k$. Note that each part contains at most one terminal, and the number of parts containing a terminal is at most k . For the non-terminal parts, using Lemma 3.4 (with $a_i = |P_i|$, $b_i = \delta(P_i)$, $a = (1 + \varepsilon)\rho n$, $b = 2B'$, $S = n$ and $T = kB'$) we obtain a bound of $5k/2$, which implies $t \leq 4k$.

Next observe that sets $\{Q_i\}_{i=1}^k$ in Step 5 above are well-defined, and can be found using a simple greedy rule: this is because each group contains at most k parts. This gives Theorem 1.6 (i). Since

$t \leq 4k$ there are at most 4 groups and hence $\max_{i=1}^k \delta(Q_i) \leq 4 \cdot \max_{i=1}^t \delta(P_i) \leq 8B'$. This proves Theorem 1.6 (iii). Also, the proof of Claim 4.3 implies Theorem 1.6 (iv); this is due to the fact that in the final partition $\sum_{i=1}^k \delta(Q_i) \leq \sum_{\ell=1}^{|S|} \delta(P_\ell)$.

We now show Theorem 1.6 (ii), i.e. $\max_{i=1}^k |Q_i| \leq (2 + \varepsilon)\rho n$. Consider any $i \in [k]$ and let P'_i denote the part assigned to Q_i from the first group. By the non-increasing size ordering P_1, \dots, P_t and the round-robin assignment (Step 5) into Q_i s, we obtain that $|Q_i| - |P'_i| \leq |Q_j|$ for all $j \in [k]$. Taking an average, $|Q_i| - |P'_i| \leq \frac{1}{k} \sum_{j=1}^k |Q_j| = \frac{n}{k}$. Finally, since each part $\{P_\ell\}_{\ell=1}^t$ has size at most $(1 + \varepsilon)\rho n$ we have $|Q_i| \leq (2 + \varepsilon)\rho n$. This completes the proof of Theorem 1.6.

5 Hardness of Min-Max-Multiway-Cut

In this section we prove Theorem 1.5, which shows that obtaining a $k^{1-\varepsilon}$ -approximation algorithm for Min-Max-Multiway-Cut should be very challenging, and suggests that some dependence on n might be necessary, unless we are satisfied with approximation factor that is linear in k , which is trivial. This situation is in contrast to several other cut problems with sum-objective (Multiway-Cut, Multicut, Requirement-Cut etc.), where $\text{poly}(\log k)$ approximation guarantees are known when k denotes the size of the terminal set [Moi09, LM10, EGK⁺10, CLLM10, MM10]. Throughout this section, we assume k is independent of n , which simplifies the statements; strictly speaking, our reductions relate solving one problem with parameter $k(n)$ to solving another problem with parameter $k(n')$.

We will refer to the min-sum version of Min-Max k -Partitioning, called Min-Sum k -Partitioning, in which the input is an edge-weighted graph $G = (V, E)$ and a parameter k , and the goal is to partition the vertices into k equal-sized parts while minimizing the total edge-weight of all edges cut. An algorithm for Min-Sum k -Partitioning is an (α, β) bicriteria approximation if for every instance, it partitions the vertices into k pieces, each of size at most $\beta|V|/k$, and the total edge-weight of all edges cut is at most α times the least possible among all partitions into k equal-size sets.

The basic idea in proving Theorem 1.5 is as follows. Although there is no vertex-balance requirement in Min-Max-Multiway-Cut, an edge-balance is implicit in the objective. By introducing a complete bipartite graph (having suitable edge weight) between the terminals and the rest of the graph, this edge-balance can be used to enforce the vertex-balance required in Min-Sum k -Partitioning. After this first step, which obtains a bicriteria approximation for Min-Sum k -Partitioning, we do a second step which improves the size-violation in the algorithm for Min-Sum k -Partitioning. These two steps are formalized in the two foregoing Lemmas 5.1 and 5.3. Putting them together immediately proves Theorem 1.5.

Lemma 5.1. *If there is a ρ -approximation algorithm for Min-Max-Multiway-Cut then there is a $(5\rho k, 10\rho)$ -bicriteria approximation algorithm for Min-Sum k -Partitioning.*

Proof. Let \mathcal{A} denote the ρ -approximation algorithm for Min-Max-Multiway-Cut. Consider any instance \mathcal{I} of Min-Sum k -Partitioning: graph $G = (V, E)$, edge costs $c : E \rightarrow \mathbb{R}_+$ and parameter k . For every $B \geq 0$, consider an instance $\mathcal{J}(B)$ of Min-Max-Multiway-Cut on graph G_B defined as:

- The vertex set is $V \cup \{t_i\}_{i=1}^k$ where $\{t_i\}_{i=1}^k$ are the terminals.
- The edges are $E \cup \{(t_i, u) : i \in [k], u \in V\}$.
- Extend cost function c by setting $c(t_i, u) = \frac{B}{n}$ for all $i \in [k], u \in V$.

Note that every solution to $\mathcal{J}(B)$ corresponds to a k -partition in G (though possibly unbalanced). We say that a solution to $\mathcal{J}(B)$ is β -balanced if each piece in the partition has size at most $\beta \cdot \frac{n}{k}$.

The algorithm for Min-Sum k -Partitioning on \mathcal{I} runs algorithm \mathcal{A} on all the Min-Max-Multiway-Cut instances $\{\mathcal{J}(2^i) : 0 \leq i \leq \log_2(\sum_{e \in E} c_e)\}$, and returns the cheapest partition that is (10ρ) -balanced. We now show that this results in a $(5\rho k, 10\rho)$ bicriteria approximation ratio.

Note that algorithm \mathcal{A} must be invoked on $\mathcal{J}(B)$ for some value B with $B < \text{OPT}(\mathcal{I}) \leq 2 \cdot B$. We will show that the partition resulting from this call is the desired bicriteria approximation.

Claim 5.2. $\text{OPT}(\mathcal{J}(B)) \leq \text{OPT}(\mathcal{I}) + 2B \leq 5\text{OPT}(\mathcal{I})$.

Proof. Let P^* denote the optimal k -partition to \mathcal{I} . Consider the solution to $\mathcal{J}(B)$ obtained by including each terminal into a distinct piece of P^* . The boundary of the piece containing t_i (any $i \in [k]$) in G_B costs at most:

$$\text{OPT}(\mathcal{I}) + n \cdot \frac{B}{n} + \frac{n}{k} \cdot (k-1) \cdot \frac{B}{n} \leq \text{OPT}(\mathcal{I}) + 2B,$$

the term $\text{OPT}(\mathcal{I})$ is due to edges in E , the second term is due to edges at t_i and the third is due to edges at all other $\{t_j : j \neq i\}$. The claim now follows since $B \leq 2\text{OPT}(\mathcal{I})$. \square

Let P denote \mathcal{A} 's solution to $\mathcal{J}(B)$, and $\{P_i\}_{i=1}^k$ the partition of V induced by P . From Claim 5.2, P has objective value (for Min-Max-Multiway-Cut) at most $5\rho \text{OPT}(\mathcal{I})$. Note that the boundary (in graph G_B) of t_i 's piece in P costs at least $c(\delta_G(P_i)) + (k-1)\frac{B}{n} \cdot |P_i|$. Thus we obtain:

$$c(\delta_G(P_i)) + (k-1)\frac{B}{n} \cdot |P_i| \leq 5\rho \text{OPT}(\mathcal{I}), \quad \text{for all } i \in [k].$$

It follows that for every $i \in [k]$, we have:

1. $|P_i| \leq \frac{5\rho \text{OPT}(\mathcal{I})}{B} \cdot \frac{n}{k-1} \leq 10\rho \cdot \frac{n}{k}$ since $B > \text{OPT}(\mathcal{I})$, and
2. $c(\delta_G(P_i)) \leq 5\rho \cdot \text{OPT}(\mathcal{I})$.

Thus the solution $\{P_i\}_{i=1}^k$ to \mathcal{I} is (10ρ) -balanced and costs at most $5\rho k \cdot \text{OPT}(\mathcal{I})$. This completes the proof of Lemma 5.1. \square

Lemma 5.3. *If there is an $(\alpha, k^{1-\varepsilon})$ -bicriteria approximation algorithm for Min-Sum k -Partitioning for some constant $\varepsilon > 0$, then there is also an $(\alpha \cdot \log \log k, \gamma)$ -bicriteria approximation algorithm with $\gamma \leq 3^{2/\varepsilon}$.*

Proof. Let \mathcal{A} denote the $(\alpha, k^{1-\varepsilon})$ -bicriteria approximation algorithm. The idea is to use \mathcal{A} recursively to obtain the claimed $(\alpha \cdot \log \log k, \gamma)$ approximation; details are below. Let G denote the input graph with n vertices, and OPT the optimal balanced k -partition. The algorithm deals with several sub-instances of G , each of which is assigned a *level* from $\{0, 1, \dots, t\}$ where $t := \Theta(\log \log k)$ is fixed below. Every level i instance will contain $k_i = k^{(1-\varepsilon/2)^i}$ parts and $n_i = \frac{n}{k} \cdot k_i$ vertices. Choose t to be the smallest integer such that $k_t \leq 3^{2/\varepsilon}$. While generating the sub-instances we also add dummy singleton vertices (to keep the instances balanced). For notational simplicity we use the same identifier for a sub-instance and the graph corresponding to it. Note that G is the unique level 0 instance. For each $i \in \{0, 1, \dots, t-1\}$, every level i instance \mathcal{I} generates k_i level $i+1$ instances as follows:

- Run algorithm \mathcal{A} on \mathcal{I} to obtain a $k_i^{1-\varepsilon}$ -balanced partition $\{P_j : 1 \leq j \leq k_i\}$ of \mathcal{I} .
- For each $j \in [k_i]$, add $n_{i+1} - |P_j|$ singleton vertices to $\mathcal{I}[P_j]$ to obtain a new level $i+1$ instance.

Note that this process is indeed well-defined since $|P_j| \leq \frac{n_i}{k_i} \cdot k_i^{1-\varepsilon} \leq \frac{n_i}{k_i} \cdot k_i^{1-\varepsilon/2} = \frac{n_i}{k_i} \cdot k_{i+1} = n_{i+1}$. Moreover the total number of level $i+1$ instances is $\prod_{\ell=0}^i k_\ell = k^{\sum_{\ell=0}^i (1-\varepsilon/2)^\ell} \leq k^{2/\varepsilon}$. Since each instance has size at most n , the total size of all instances is polynomial.

The algorithm finally returns the partition \mathcal{P} corresponding to the set of all level t instances. Note that there are at most $n_t = \frac{n}{k} \cdot k_t \leq 3^{2/\varepsilon} \cdot \frac{n}{k}$ vertices in each level t instance. The algorithm ignores all dummy vertices in each piece of \mathcal{P} and greedily merges pieces until every piece has at least n/k vertices (all from V); let \mathcal{P}' denote the resulting partition of V . Clearly there are at most k pieces in \mathcal{P}' , and each has size at most $3^{2/\varepsilon} \cdot \frac{n}{k}$. Thus \mathcal{P}' is a $3^{2/\varepsilon}$ -balanced k -partition.

We now upper bound the total cost of all edges removed by the algorithm (over all instances); this also bounds the cost of \mathcal{P}' . This is immediate from Claim 5.4 below: since there are t levels and \mathcal{A} achieves an α -approximation to the cost, the total cost is bounded by $\alpha t \cdot \text{OPT}$.

Claim 5.4. *For each $i \in \{1, \dots, t\}$, the sum of optimal values of level i instances is at most OPT .*

Proof. Consider any fixed i , and an instance \mathcal{I} of level i . We will show that the edges of OPT induced on \mathcal{I} form a balanced k_i -partition for \mathcal{I} . This suffices to prove the claim since the level i instances partition V (the original vertex-set). Let V' denote the vertices from V in \mathcal{I} ; note that $|V'| \leq \frac{n_{i-1}}{k_{i-1}} \cdot k_i^{1-\varepsilon}$. Consider the partition \mathcal{Q} of V' induced by OPT ; note that each piece in \mathcal{Q} has size at most n/k . Greedily merge pieces in \mathcal{Q} as long as the size of each piece is at most n/k , to obtain partition \mathcal{Q}' ; so the number of pieces is:

$$|\mathcal{Q}'| \leq 1 + 2|V'| \cdot \frac{k}{n} \leq 1 + 2 \frac{n_{i-1}}{k_{i-1}} k_i^{1-\varepsilon} \cdot \frac{k}{n} = 1 + 2 k_{i-1}^{1-\varepsilon} \leq 3 k_{i-1}^{1-\varepsilon} \leq k_{i-1}^{1-\varepsilon/2} = k_i.$$

The second-last inequality uses $k_{i-1} \geq 3^{2/\varepsilon}$ which is true by the choice of t . Now we can fill pieces of \mathcal{Q}' with dummy singleton vertices of instance \mathcal{I} to obtain a balanced k_i -partition of \mathcal{I} . \square

This completes the proof of Lemma 5.3. \square

Acknowledgements. We thank the anonymous reviewers for their careful reading and instructive comments.

References

- [AK09] N. Alon and B. Klartag. Economical Toric Spines via Cheeger's Inequality. *Journal of Topology and Analysis*, 1:101–111, 2009.
- [AR98] Y. Aumann and Y. Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM J. Comput.*, 27(1), 1998.
- [AR06] K. Andreev and H. Racke. Balanced graph partitioning. *Theor. Comp. Sys.*, 39(6):929–939, 2006.
- [ARV08] S. Arora, S. Rao, and U. V. Vazirani. Geometry, flows, and graph-partitioning algorithms. *Commun. ACM*, 51(10):96–105, 2008.
- [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.

- [BLNZ11] N. Bansal, K.-W. Lee, V. Nagarajan, and M. Zafer. Minimum congestion mapping in a cloud. In *PODC*, 2011.
- [CKN09] J. Cheeger, B. Kleiner, and A. Naor. A $(\log n)^{\Omega(1)}$ integrality gap for the sparsest cut SDP. In *FOCS*, pages 555–564. IEEE Computer Society, 2009.
- [CKR00] G. Calinescu, H. J. Karloff, and Y. Rabani. An improved approximation algorithm for multiway cut. *J. Comput. Syst. Sci.*, 60(3):564–574, 2000.
- [CLLM10] M. Charikar, T. Leighton, S. Li, and A. Moitra. Vertex sparsifiers and abstract rounding algorithms. In *FOCS*, pages 265–274. IEEE Computer Society, 2010.
- [CMM06] E. Chlamtac, K. Makarychev, and Y. Makarychev. How to play unique games using embeddings. In *FOCS*, pages 687–696, 2006.
- [CRB09] N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba. Virtual network embedding with coordinated node and link mapping. In *INFOCOM*, pages 783–791. IEEE, 2009.
- [EGK⁺10] M. Englert, A. Gupta, R. Krauthgamer, H. Raecke, I. Talgam-Cohen, and K. Talwar. Vertex sparsifiers: New results from old techniques. In *APPROX*, 2010.
- [ENRS99] G. Even, J. Naor, S. Rao, and B. Schieber. Fast approximate graph partitioning algorithms. *SIAM J. Comput.*, 28(6):2187–2214, 1999.
- [FKN03] U. Feige, R. Krauthgamer, and K. Nissim. On cutting a few vertices from a graph. *Discrete Appl. Math.*, 127:643–649, May 2003.
- [FT03] J. Fakcharoenphol and K. Talwar. Improved decompositions of graphs with forbidden minors. In *6th International workshop on Approximation algorithms for combinatorial optimization*, pages 871–882, 2003.
- [GKL03] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *44th Annual IEEE Symposium on Foundations of Computer Science*, pages 534–543, October 2003.
- [KKS⁺04] D. R. Karger, P. Klein, C. Stein, M. Thorup, and N. E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. *Math. Oper. Res.*, 29(3):436–461, 2004.
- [KNS09] R. Krauthgamer, J. Naor, and R. Schwartz. Partitioning graphs into balanced components. In *SODA*, pages 942–949, 2009.
- [KPR93] P. Klein, S. A. Plotkin, and S. Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, STOC, pages 682–690, New York, NY, USA, 1993. ACM.
- [KR11] R. Krauthgamer and T. Roughgarden. Metric clustering via consistent labeling. *Theory of Computing*, 7(1):49–74, 2011.
- [KST98] M. Kiwi, D. A. Spielman, and S.-H. Teng. Min-max-boundary domain decomposition. *Theor. Comput. Sci.*, 261, 1998.
- [LLR95] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2), 1995.
- [LM10] T. Leighton and A. Moitra. Extensions and limits to vertex sparsification. In *STOC*, pages 47–56, 2010.
- [LN06] J. R. Lee and A. Naor. L_p metrics on the Heisenberg group and the Goemans-Linial conjecture. In *FOCS*, pages 99–108. IEEE Computer Society, 2006.
- [LR99] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999.

- [LS10] J. R. Lee and A. Sidiropoulos. Genus and the geometry of the cut graph. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 193–201. SIAM, 2010.
- [MM10] K. Makarychev and Y. Makarychev. Metric extension operators, vertex sparsifiers and lipschitz extendability. In *FOCS*, pages 255–264. IEEE Computer Society, 2010.
- [Moi09] A. Moitra. Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size. In *FOCS*, pages 3–12, 2009.
- [Räc08] H. Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *STOC*, pages 255–264, 2008.
- [Rao99] S. Rao. Small distortion and volume preserving embeddings for planar and Euclidean metrics. In *Proceedings of the 15th Annual Symposium on Computational Geometry*, pages 300–306. ACM, 1999.
- [RS10] P. Raghavendra and D. Steurer. Graph expansion and the unique games conjecture. In *STOC*, pages 755–764, 2010.
- [RST10] P. Raghavendra, D. Steurer, and P. Tetali. Approximations for the isoperimetric and spectral profile of graphs and related parameters. In *STOC*, pages 631–640, 2010.
- [RST12] P. Raghavendra, D. Steurer, and M. Tulsiani. Reductions between expansion problems. In *IEEE Conference on Computational Complexity*, pages 64–73, 2012.
- [ST04] Z. Svitkina and É. Tardos. Min-max multiway cut. In *APPROX-RANDOM*, pages 207–218, 2004.
- [YYRC08] M. Yu, Y. Yi, J. Rexford, and M. Chiang. Rethinking virtual network embedding: substrate support for path splitting and migration. *ACM SIGCOMM CCR*, 38(2):17–29, 2008.
- [ZA06] Y. Zhu and M. H. Ammar. Algorithms for assigning substrate network resources to virtual network components. In *INFOCOM*. IEEE, 2006.

Appendix

A Integrality Gap for SDP Relaxation of Min-Max-Multiway-Cut

Consider the following semi-definite relaxation for the min-max multiway cut problem:

$$\begin{aligned} \min \quad & \lambda \\ \text{s.t.} \quad & \lambda \geq \sum_{(u,v) \in E} \|y_{u,i} - y_{v,i}\|_2^2 \quad \forall i = 1, 2, \dots, k \end{aligned} \quad (18)$$

$$\sum_{i=1}^k \|y_{u,i}\|_2^2 = 1 \quad \forall u \in V \quad (19)$$

$$\|y_{t_i,i}\|_2^2 = 1 \quad \forall i = 1, 2, \dots, k \quad (20)$$

$$y_{u,i} \cdot y_{u,j} = 0 \quad \forall u \in V, \forall i \neq j \quad (21)$$

$$y_{u,i} \cdot \sum_{j=1}^k y_{v,j} = \|y_{u,i}\|_2^2 \quad \forall u, v \in V, \forall i = 1, 2, \dots, k \quad (22)$$

$$\|y_{u,i} - y_{v,j}\|_2^2 + \|y_{v,j} - y_{w,r}\|_2^2 \geq \|y_{u,i} - y_{w,r}\|_2^2 \quad \forall u, v, w \in V, \forall i, j, r = 1, 2, \dots, k \quad (23)$$

$$y_{u,i} \cdot y_{v,j} \geq 0 \quad \forall u, v \in V, \forall i, j = 1, 2, \dots, k \quad (24)$$

$$\|y_{u,i}\|_2^2 \geq y_{u,i} \cdot y_{v,j} \quad \forall u, v \in V, \forall u, v \in V, \forall i, j = 1, 2, \dots, k \quad (25)$$

In the above relaxation, each vertex $u \in V$ is associated with k different vectors: $y_{u,1}, y_{u,2}, \dots, y_{u,k}$. Each of those corresponds to a different terminal. Notice that the last two constraints of the relaxation are the ℓ_2^2 triangle inequality constraints including the origin (the same constraints used by [CMM06]).

As stated, the integrality gap we consider is the star graph which constrains as single vertex u connected by k edges to the k terminals. The value of any integral solution to this instance is exactly $k - 1$ since the only choice is to which terminal should u be assigned. Any choice made results in a min-max objective value of $k - 1$.

Let us construct the fractional solution to the above relaxation. Fix e to be an arbitrary unit vector, and x_1, x_2, \dots, x_k be k unit vectors which are all orthogonal to e and the inner product between any two of them is $-1/(k - 1)$. Set the following fractional solution:

$$\begin{aligned} y_{t_i,i} &= e & \forall i = 1, 2, \dots, k \\ y_{t_i,j} &= 0 & \forall i \neq j \\ y_{u,i} &= \frac{1}{k}e + \frac{\sqrt{k-1}}{k}x_i & \forall i = 1, 2, \dots, k \end{aligned}$$

First, we show that the above fractional solution is feasible. Constraints (19), (20) and (21) are obviously feasible for all terminals. Vertex u also upholds constraint (19) since:

$$\sum_{i=1}^k \left\| \frac{1}{k}e + \frac{\sqrt{k-1}}{k}x_i \right\|_2^2 = \sum_{i=1}^k \left(\frac{1}{k^2} + \frac{k-1}{k^2} \right) = 1.$$

Let us verify that vertex u satisfies also constraint (21):

$$\left(\frac{1}{k}e + \frac{\sqrt{k-1}}{k}x_i\right) \cdot \left(\frac{1}{k}e + \frac{\sqrt{k-1}}{k}x_j\right) = \frac{1}{k^2} + \frac{k-1}{k^2} \cdot \left(-\frac{1}{k-1}\right) = 0.$$

Focus on constraint (22). It is easy to verify that for all terminals, the sum of all k vectors that are associated with the picked vertex is exactly e . Since $\sum_{i=1}^k x_i = 0$, the sum of all vectors associated with u is also e . Therefore, all constraints of type (22) are satisfied.

Regarding the last three constraints, which are the ℓ_2^2 triangle inequality constraints including the origin, one can verify the following:

$$\begin{aligned} \|e - 0\|_2^2 &= 1 \\ \left\|e - \left(\frac{1}{k}e - \frac{\sqrt{k-1}}{k}x_i\right)\right\|_2^2 &= 1 - \frac{1}{k} \\ \left\|0 - \left(\frac{1}{k}e - \frac{\sqrt{k-1}}{k}x_i\right)\right\|_2^2 &= \frac{1}{k} \\ \left\|\left(\frac{1}{k}e - \frac{\sqrt{k-1}}{k}x_i\right) - \left(\frac{1}{k}e - \frac{\sqrt{k-1}}{k}x_j\right)\right\|_2^2 &= \frac{2}{k} \end{aligned}$$

All the last three constraints are derived from the above calculations. Hence, we can conclude that the fractional solution defined above is feasible for the semi-definite relaxation.

The value of this solution is:

$$\max_{1 \leq i \leq k} \left\{ \left(1 - \frac{1}{k}\right) + (k-1) \cdot \frac{1}{k} \right\} = 2\frac{k-1}{k}.$$

This gives an integrality gap of $k/2$.

B Bad Example: Greedy Algorithm for Min-Max k -Partitioning

We show that the naive greedy algorithm that repeatedly uses Small-Set Expansion to remove a part of size n/k performs very poorly. In this example we even assume that there is an exact algorithm for Small-Set Expansion. The graph is a tree on $n = k^2$ vertices $V := \{v\} \cup \bigcup_{i=1}^{k-1} \{u_{i,0}, \dots, u_{i,k}\}$, and edges $E = \bigcup_{i=1}^{k-1} \{(u_{i,j}, u_{i,j-1}) : 1 \leq j \leq k\} \cup \{(u_{i,0}, u_{i-1,0}) : 2 \leq i \leq k-1\} \cup \{(v, u_{1,0})\}$.

The simple greedy algorithm will cut out parts having small boundary for $k-1$ iterations, namely $P_i = \{u_{i,1}, \dots, u_{i,k}\}$ for $i \in [k-1]$; note that $\delta(P_i) = 1$ for all $i \in [k-1]$. However the last part $\{v, u_{1,0}, \dots, u_{k-1,0}\}$ has cut-value $k-1$; so the resulting objective value is $k-1$.

On the other hand, it can be checked directly that the optimal value is at most *four*: Consider the partition obtained by repeatedly taking the first k consecutive vertices from the ordering $v, u_{1,0}, \dots, u_{1,k}, u_{2,0}, \dots, u_{2,k}, \dots, u_{k-1,0}, \dots, u_{k-1,k}$.