Nonuniform Graph Partitioning with Unrelated Weights*

Konstantin Makarychev
Northwestern University

Yury Makarychev†
Toyota Technological Institute at Chicago

Abstract

We give a bi-criteria approximation algorithm for the Minimum Nonuniform Graph Partitioning problem, recently introduced by Krauthgamer, Naor, Schwartz, and Talwar (2014). In this problem, we are given a graph $G = (V, E)$ on n vertices and k numbers $\rho_1, \dots, \rho_k$. The goal is to partition V into k disjoint sets $P_1, \dots, P_k$ satisfying $|P_i| \leq \rho_i n$ so as to minimize the number of edges cut by the partition.

Our bi-criteria algorithm gives an $O(\sqrt{\log |V| \log k})$ approximation for the objective function in general graphs and an $O(1)$ approximation in graphs excluding a fixed minor. The approximate solution satisfies relaxed capacity constraints $|P_i| \leq (5 + \varepsilon)\rho_i n$. This algorithm is an improvement upon the $O(\log n)$-approximation algorithm by Krauthgamer, Naor, Schwartz, and Talwar (2014). Our approximation ratio matches the best known ratio for the Minimum (Uniform) k-Partitioning problem.

We extend our results to the case of "unrelated weights" and to the case of "unrelated d-dimensional weights". In the former case, different vertices may have different weights, and the weight of a vertex may depend on the set P_i the vertex is assigned to. In the latter case, each vertex u has a d-dimensional weight $r(u, i) = (r_1(u, i), \dots, r_d(u, i))$ if u is assigned to P_i. Each set P_i has a d-dimensional capacity $c(i) = (c_1(i), \dots, c_d(i))$. The goal is to find a partition such that $\sum_{u \in P_i} r(u, i) \leq c(i)$ coordinate-wise.

1 Introduction

We study the Minimum Nonuniform Partitioning problem, which was recently proposed by Krauthgamer, Naor, Schwartz, and Talwar (2014). We are given a graph $G = (V, E)$, parameter k, and k numbers (capacities) $\rho_1, \dots, \rho_k$. The goal is to partition V into k pieces (bins) $P_1, \dots, P_k$ satisfying the capacity constraints $|P_i| \leq \rho_i |V|$ so as to minimize the number of cut edges. The problem is a generalization of the Minimum k-Partitioning problem studied by Krauthgamer, Naor, and Schwartz (2009), in which all bins have equal capacity $\rho_i = 1/k$. Denote $n = |V|$.

The problem has many applications (see Krauthgamer et al. 2014). Consider an example in cloud computing: Imagine that we need to distribute n computational tasks – vertices of the graph – among k machines, each with capacity $\rho_i n$. Different tasks communicate with each other. The amount of communication between tasks u and v equals the weight of the edges between the corresponding vertices u and v. The goal is to distribute tasks among k machines subject to the capacity constraints so as to minimize the total amount of communication between machines.[1]

The problem is quite challenging. Krauthgamer et al. (2014) note that many existing techniques do not work for this problem. Particularly, it is not clear how to solve this problem on tree graphs[2] and consequently

*The conference version of this paper appeared at ICALP 2014.
†Supported by NSF CAREER award CCF-1150062 and NSF grant IIS-1302662.
[1] In this example, we need to solve a variant of the problem with edge weights.
[2] Our algorithm gives a constant factor bi-criteria approximation for trees.

DRAFT

how to use Räcke's (2008) tree decomposition technique. Krauthgamer et al. (2014) give an $O(\log n)$ bi-criteria approximation algorithm for the problem: the algorithm finds a partition $P_1, \dots, P_k$ such that $|P_i| \leq O(\rho_i n)$ for every i and the number of cut edges is $O(\log n \,\mathrm{OPT})$, where OPT is the cost of the optimal solution. The algorithm first solves a configuration linear program and then uses a new sophisticated method to round the fractional solution.

In this paper, we present a rather simple SDP based $O(\sqrt{\log n \log k})$ bi-criteria approximation algorithm for the problem. We note that our approximation guarantee matches that of the algorithm of Krauthgamer, Naor, and Schwartz (2009) for the the Minimum k-Partitioning problem (which is a special case of Minimum Nonuniform Partitioning, see above). Our algorithm uses a technique of "orthogonal separators" developed by Chlamtac, Makarychev, and Makarychev (2006) and later used by Bansal, Feige, Krauthgamer, Makarychev, Nagarajan, Naor, and Schwartz (2014) for the Small Set Expansion problem. Using orthogonal separators, it is relatively easy to get a distribution over partitions $\{P_1, \dots, P_k\}$ such that $\mathbb{E}[|P_i|] \leq O(\rho_i n)$ for all i and the expected number of cut edges is $O(\sqrt{\log n \log(1/\rho_{min})}\,\mathrm{OPT})$ where $\rho_{min} = \min_i \rho_i$. The problem is that for some i, P_i may be much larger than its expected size. The algorithm by Krauthgamer et al. (2014) solves a similar problem by first simplifying the instance and then grouping parts P_i into "mega-buckets". We propose a simpler fix: Roughly speaking, if a set P_i contains too many vertices, we remove some of these vertices from P_i and re-partition the removed vertices into k pieces again. Thus we ensure that all capacity constraints are (approximately) satisfied. It turns out that every vertex gets removed a constant number of times in expectation. Hence, the re-partitioning step increases the number of cut edges only by a constant factor. Another problem is that $1/\rho_{min}$ may be much larger than k. To deal with this problem, we transform the SDP solution (eliminating "short" vectors) and redefine thresholds ρ_i so that $1/\rho_{min}$ becomes $O(k)$.

Our technique is quite robust and allows us to solve more general versions of the problem, *Nonuniform Graph Partitioning with unrelated weights* and *Nonuniform Graph Partitioning with unrelated d-dimensional weights*, which we discuss next.

Minimum Nonuniform Graph Partitioning with unrelated weights captures the variant of the problem where we assign vertices (tasks/jobs) to unrelated machines and the weight of a vertex (the size of the task/job) depends on the machine it is assigned to. Namely, in this variant of the problem, we assume that a vertex u has weight $\mu_i(u) \geq 0$ if it is assigned to bin i. We require that the total weight of all vertices in bin i is at most ρ_i. We denote the total weight of the vertices in a set S by $\mu_i(S)$; i.e., we think of μ_i as a measure on V. By rescaling the weights $\mu_i(u)$ and capacities ρ_i, we may assume without loss of generality that $\mu_i(V) = 1$ for every i.

Definition 1.1 (Minimum Nonuniform Graph Partitioning with unrelated weights)**.** *We are given a graph $G = (V, E)$ on n vertices and a natural number $k \geq 2$. Additionally, we are given k normalized measures $\mu_1, \dots, \mu_k$ on V (satisfying $\mu_i(V) = 1$) and k numbers $\rho_1, \dots, \rho_k \in (0, 1)$. The goal is to partition the graph into k pieces (bins) $P_1, \dots, P_k$ such that $\mu_i(P_i) \leq \rho_i$ so as to minimize the number of cut edges. Some pieces P_i may be empty.*

We will only consider instances of Minimum Nonuniform Graph Partitioning that have a feasible solution. We give an $O_\varepsilon(\sqrt{\log n \log \min(1/\rho_{min}, k)})$ bi-criteria approximation algorithm for the problem (to indicate that the hidden constant in the big-O notation may depend on ε, we use notation $O_\varepsilon()$). If the instance does not have a feasible solution, the algorithm either finds a solution satisfying relaxed capacity constraints $\mu_i(P_i) \leq 5(1+\varepsilon)\rho_i$ or reports that the instance does not have a feasible solution.

Theorem 1.2. *For every $\varepsilon > 0$, there exists a randomized polynomial-time algorithm that given a feasible instance of Minimum Nonuniform Graph Partitioning with unrelated weights finds a partition $P_1, \dots, P_k$ of*

DRAFT

V satisfying $\mu_i(P_i) \leq 5(1+\varepsilon)\rho_i$. The expected cost of the solution is at most $D \cdot \mathrm{OPT}$, where OPT is the optimal value, $D = O_\varepsilon(\sqrt{\log n \log \min(1/\rho_{min}, k)})$ and $\rho_{min} = \min_i \rho_i$. For graphs excluding a fixed minor, $D = O_\varepsilon(1)$.

Nonuniform Graph Partitioning with unrelated d-dimensional weights further generalizes the problem. In this variant of the problem, we assume that we have d resources (e.g. CPU speed, random access memory, disk space, network). Each piece P_i has $c_j(i)$ units of resource $j \in \{1, \ldots, d\}$, and each vertex u requires $r_j(u,i)$ units of resource $j \in \{1, \ldots, d\}$ when it is assigned to piece P_i. We need to partition V so that the capacity constraints for all resources are satisfied. The d-dimensional version of Minimum (uniform) k-Partitioning was previously studied by Amir et al. (2014). In their problem, all $\rho_i = 1/k$ are the same, and r_j's do not depend on i.

Definition 1.3 (Minimum Nonuniform Graph Partitioning with unrelated d-dimensional weights)**.** *We are given a graph $G = (V, E)$ on n vertices. Additionally, we are given non-negative numbers $c_j(i)$ and $r_j(u,i)$ for $i \in \{1, \ldots, k\}$, $j \in \{1, \ldots, d\}$, $u \in V$. The goal is to find a partition of V into $P_1, \ldots, P_k$ subject to the capacity constraints $\sum_{u \in V} r_j(u,i) \leq c_j(i)$ for every i and j so as to minimize the number of cut edges.*

We present a bi-criteria approximation algorithm for this problem.

Theorem 1.4. *For every $\varepsilon > 0$, there exists a randomized polynomial-time algorithm that given a feasible instance of Minimum Nonuniform Graph Partitioning with unrelated d-dimensional weights finds a partition $P_1, \ldots, P_k$ of V satisfying*

$$\sum_{v \in V} r_j(v,i) \leq 5d(1+\varepsilon)c_j(i) \quad \text{for every } i \text{ and } j.$$

The expected cost of the solution is at most $D \cdot \mathrm{OPT}$, where OPT is the optimal value, $D = O_\varepsilon(\sqrt{\log n \log k})$. For graphs excluding a fixed minor $D = O_\varepsilon(1)$.

This result is a simple corollary of Theorem 1.2: we let $\mu'_i(u) = \max_j(r_j(u,i)/c_j(i))$ and then apply our result to measures $\mu_i(u) = \mu'_i(u)/\mu'_i(V)$ (we describe the details in Section 4).

We remark that our algorithms work if edges in the graph have arbitrary positive weights. However, for simplicity of exposition, we describe the algorithms for the setting where all edge weights are equal to one. To deal with arbitrary edge weights, we only need to change the SDP objective function.

Our paper strengthens the result by Krauthgamer et al. (2014) in two ways. First, it improves the approximation factor from $O(\log n)$ to $O(\sqrt{\log n \log k})$. Second, it studies considerably more general variants of the problem, Minimum Nonuniform Partitioning with unrelated weights and Minimum Nonuniform Partitioning with unrelated d-dimensional weights. We believe that these variants are very natural. Indeed, one of the main motivations for the Minimum Nonuniform Partitioning problem is its applications to scheduling and load balancing: in these applications, the goal is to assign tasks to machines so as to minimize the total amount of communication between different machines, subject to the capacity constraints. The constraints that we study in the paper are very general and analogous to those that are often considered in the scheduling literature. We note that the method developed by Krauthgamer et al. (2014) does not handle these more general variants of the problem.

2 Algorithm

SDP Relaxation. Our relaxation for the problem is based on the SDP relaxation for the Small Set Expansion (SSE) problem of Bansal et al. (2014). We write the SSE relaxation for every cluster P_i and then add

DRAFT

consistency constraints similar to constraints used in Unique Games. For every vertex u and index $i \in \{1, \dots, k\}$, we introduce a vector $\bar{u}_i$. In the integral solution, this vector is simply the indicator variable for the event "$u \in P_i$". It is easy to see that in the integral case, the number of cut edges equals (1) (see below). Indeed, if u and v lie in the same P_j, then $\bar{u}_i = \bar{v}_i$ for all i; if u lies in $P_{j'}$ and v lies in $P_{j''}$ (for $j' \neq j''$) then $\|\bar{u}_i - \bar{v}_i\|^2 = 1$ for $i \in \{j', j''\}$ and $\|\bar{u}_i - \bar{v}_i\|^2 = 0$ for $i \notin \{j', j''\}$. The SDP objective is to minimize (1).

We add constraint (2) saying that $\mu_i(P_i) \leq \rho_i$. We further add spreading constraints (3) from Bansal et al. (2014) (see also Louis and Makarychev (2014)). The spreading constraints above are satisfied in the integral solution: If $u \notin P_i$, then $\bar{u}_i = 0$ and both sides of the inequality equal 0. If $u \in P_i$, then the left-hand side equals $\mu_i(P_i)$, and the right hand side equals ρ_i.

We write standard ℓ_2^2-triangle inequalities (4) and (5). Finally, we add consistency constraints. Every vertex u must be assigned to one and only one P_i, hence constraint (6) is satisfied. We obtain the following SDP relaxation.

SDP Relaxation

$$\min \frac{1}{2} \sum_{i=1}^{k} \sum_{(u,v) \in E} \|\bar{u}_i - \bar{v}_i\|^2 \tag{1}$$

subject to

$$\sum_{u \in V} \|\bar{u}_i\|^2 \mu_i(u) \leq \rho_i \qquad \text{for all } i \in [k] \tag{2}$$

$$\sum_{v \in V} \langle \bar{u}_i, \bar{v}_i \rangle \mu_i(v) \leq \|\bar{u}_i\|^2 \rho_i \qquad \text{for all } u \in V,\ i \in [k] \tag{3}$$

$$\|\bar{u}_i - \bar{v}_i\|^2 + \|\bar{v}_i - \bar{w}_i\|^2 \geq \|\bar{u}_i - \bar{w}_i\|^2 \qquad \text{for all } u, v, w \in V,\ i \in [k] \tag{4}$$

$$0 \leq \langle \bar{u}_i, \bar{v}_i \rangle \leq \|\bar{u}_i\|^2 \qquad \text{for all } u, v \in V,\ i \in [k] \tag{5}$$

$$\sum_{i=1}^{k} \|\bar{u}_i\|^2 = 1 \qquad \text{for all } u \in V \tag{6}$$

Small Set Expansion and Orthogonal Separators. Our algorithm uses a technique called "orthogonal separators". The notion of orthogonal separators was introduced in Chlamtac, Makarychev, and Makarychev (2006), where it was used in an algorithm for Unique Games. Later, Bansal et al. (2014) showed that the following holds. If an SDP solution satisfies constraints (2), (3), (4), and (5), then for every $\varepsilon \in (0, 1)$, $\delta \in (0, 1)$, and $i \in [k]$, there exist a distortion $D_i = O_\varepsilon(\sqrt{\log n \log(1/(\delta\rho_i))})$, and a probability distribution over subsets of V such that for a random set $S_i \subset V$ ("orthogonal separator") drawn from this distribution, we have for $\alpha = 1/n$,

- $\mu_i(S_i) \leq (1 + \varepsilon)\rho_i$ (always);
- For all u, $\Pr(u \in S_i) \in [(1 - \delta)\alpha\|\bar{u}_i\|^2, \alpha\|\bar{u}_i\|^2]$;
- For all $(u, v) \in E$, $\Pr(u \in S_i, v \notin S_i) \leq \alpha D_i \cdot \|\bar{u}_i - \bar{v}_i\|^2$.

This statement was proved in Bansal et al. (2014) implicitly, so for completeness we prove it in Section 5 — see Theorem 5.1. We let $D = \max_i D_i$. For graphs excluding a fixed minor and bounded genus graphs, $D = O_\varepsilon(1)$.

DRAFT

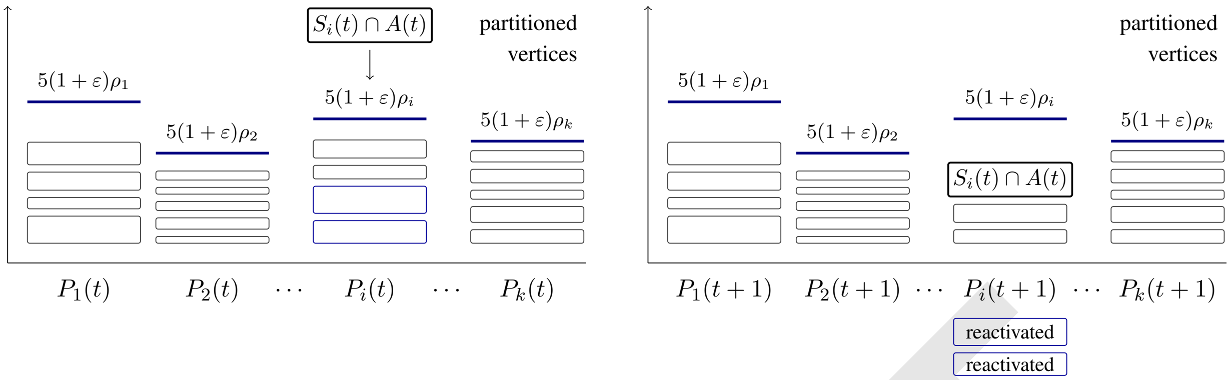

Figure 1: The figure shows how we update sets $P_i(t)$ in iteration t. In this figure, rectangles represent layers of vertices in sets $P_i(t)$ (on the left) and $P_i(t+1)$ (on the right). All vertices in these layers are inactive (they are already partitioned). Blue horizontal lines show capacity constraints. In the example shown in the figure, we add set $S_i(t) \cap A(t)$ to $P_i(t)$. The measure of the obtained set is greater than $5(1+\varepsilon)\rho_i$, and so we remove the two bottom layers from $P_i(t) \cup (S_i(t) \cap A(t))$ (the removed layers are shown in blue). We get a set of measure at most $5(1+\varepsilon)\rho_i$. Vertices in the removed layers are reactivated after the iteration is over.

Algorithm. Let us examine a somewhat naïve algorithm for the problem inspired by the algorithm of Bansal et al. (2014) for Small Set Expansion. We shall maintain the set of active (yet unassigned) vertices $A(t)$. Initially, all vertices are active, i.e. $A(0) = V$. At every step t, we pick a random index $i \in \{1, \dots, k\}$ and sample an orthogonal separator $S_i(t)$ as described above. We assign all active vertices from $S_i(t)$ to bin i:

$$P_i(t+1) = P_i(t) \cup (S_i(t) \cap A(t)),$$

and mark all newly assigned vertices as inactive, i.e., we let $A(t+1) = A(t) \setminus S_i(t)$. Other bins remain the same: $P_j(t+1) = P_j(t)$ for $j \neq i$. We stop when the set of active vertices $A(t)$ is empty. We output the partition $\mathcal{P} = \{P_1(T), \dots, P_k(T)\}$, where T is the index of the last iteration.

We can show that the number of edges cut by the algorithm is at most $O(D \cdot \mathrm{OPT})$, where D is the distortion of orthogonal separators. Furthermore, the expected weight of each P_i is $O(\rho_i)$. However, weights of some pieces may significantly deviate from the expectation and may be much larger than ρ_i. So we need to alter the algorithm to guarantee that all sizes are bounded by $O(\rho_i)$ simultaneously. We face a problem similar to the one Krauthgamer, Naor, Schwartz, and Talwar (2014) had to solve in their paper. Their solution is rather complex and does not seem to work in the weighted case. Here, we propose a very simple fix for the naïve algorithm we presented above. We shall store vertices in every bin in layers. When we add new vertices to a bin at some iteration, we put them in a new layer on top of already stored vertices. Now, if the weight of bin i is greater than $5(1+\varepsilon)\rho_i$, we remove bottom layers from this bin so that its weight is at most $5(1+\varepsilon)\rho_i$. Then we mark the removed vertices as active and proceed to the next iteration. It is clear that this algorithm always returns a solution satisfying $\mu_i(P_i) \leq 5(1+\varepsilon)\rho_i$ for all i. But now we need to prove that the algorithm terminates, and that the expected number of cut edges is still bounded by $O(D \cdot \mathrm{OPT})$.

Before proceeding to the analysis, we describe the algorithm in detail.

DRAFT

Algorithm for Nonuniform Partitioning with Unrelated Weights

Input: a graph $G = (V, E)$ on n vertices; a positive integer $k \leq n$; a sequence of numbers $\rho_1, \dots, \rho_k \in (0, 1)$ (with $\rho_1 + \cdots + \rho_k \geq 1$); weights $\mu_i : V \to \mathbb{R}^+$ (with $\mu_i(V) = 1$).

Output: a partition of vertices into disjoint sets $P_1, \dots, P_k$ such that $\mu_i(P_i) \leq 5(1+\varepsilon)\rho_i$.

- The algorithm maintains a partition of V into a set of active vertices $A(t)$ and k sets $P_1(t), \dots P_k(t)$, which we call bins. For every inactive vertex $u \notin A(t)$, we remember its depth in the bin it belongs to. We denote the depth by $\mathrm{depth}_u(t)$. If $u \in A(t)$, then we let $\mathrm{depth}_u(t) = \perp$.
- Initially, set $A(0) = V$; and $P_i(0) = \varnothing$, $\mathrm{depth}_u(t) = \perp$ for all i; $t = 0$.
- Solve the SDP relaxation. If it is not feasible, return that the problem does not have a feasible solution and terminate.
- **while** $A(t) \neq \varnothing$
  1. Pick an index $i \in \{1, \dots, k\}$ uniformly at random.
  2. Sample an orthogonal separator $S_i(t) \subset V$ with $\delta = \varepsilon/4$ as described in Section 2.
  3. Add all active vertices from $S_i(t)$ to bin i as follows. If $\mu_i(P_i(t) \cup (S_i(t) \cap A(t))) \leq 5(1+\varepsilon)\rho_i$, then simply let:
     $$P_i(t+1) = P_i(t) \cup (S_i(t) \cap A(t)).$$
     Otherwise, find the largest depth h such that the set
     $$P_i^h(t+1) = \{u \in P_i(t) : \mathrm{depth}_u(t) \leq h\} \cup (S_i(t) \cap A(t))$$
     has size at most $5(1+\varepsilon)\rho_i$ and let $P_i(t+1) = P_i^h(t+1)$. (In other words, put the vertices from $S_i(t) \cap A(t)$ into bin i and then remove vertices from the bottom layers of the bin so that the weight of the bin is at most $5(1+\varepsilon)\rho_i$.)
  4. For all $j \neq i$, let $P_j(t+1) = P_j(t)$.
  5. If $A(t) \cap S_i(t) \neq \varnothing$ (and, consequently, we put at least one new vertex into bin i at the current iteration), then set the depth of all newly stored vertices to 1; increase the depth of all other vertices in bin i by 1.
  6. Update the set of active vertices: let $A(t+1) = V \setminus \bigcup_j P_j(t+1)$ and $\mathrm{depth}_u(t+1) = \perp$ for $u \in A(t+1)$. Let $t = t+1$.
- Set $T = t$ and return the partition $P_1(T), \dots, P_k(T)$.

Note that Step 3 is well defined. We can always find an index h such that $\mu_i(P_i(t+1)) \leq 5(1+\varepsilon)\rho_i$, because for $h = 0$, we have $P_i(t+1) = S_i(t) \cap A(t)$ and thus

$$\mu(P_i(t+1)) = \mu_i(S_i(t) \cap A(t)) \leq \mu_i(S_i(t)) \leq (1+\varepsilon)\rho_i < 5(1+\varepsilon)\rho_i,$$

by the first property of orthogonal separators.

DRAFT

Analysis. We will first prove Theorem 2.1, which states that the algorithm has approximation factor $D = O_\varepsilon(\sqrt{\log n \log(1/\rho_{min})})$ on arbitrary graphs, and $D = O_\varepsilon(1)$ on graphs excluding a minor. Then we will show how to obtain $D = O_\varepsilon(\sqrt{\log n \log k})$ approximation on arbitrary graphs (see Section 3). To this end, we will transform the SDP solution and redefine measures μ_i and capacities ρ_i so that $\rho_{min} \geq \delta/k$, then apply Theorem 2.1. The new SDP solution will satisfy all SDP constraints except possibly for constraint (6); it will however satisfy a relaxed constraint (where as above $\delta = \varepsilon/4$):

$$\sum_{i=1}^{k} \|\bar{u}_i\|^2 \in [1-\delta, 1] \qquad \text{for all } u \in V. \tag{6$'$}$$

Thus in Theorem 2.1, we will assume only that the solution satisfies the SDP relaxation with constraint (6) replaced by constraint (6$'$).

Theorem 2.1. *Given a feasible solution to the SDP with with constraint (6) replaced by constraint (6$'$), the algorithm returns a partition $P_1, \dots, P_k$ satisfying $\mu_i(P_i) \leq 5(1+\varepsilon)\rho_i$. The expected number of iterations of the algorithm is at most $\mathbb{E}[T] \leq 4n^2 k$, and the expected number of cut edges is at most $O(D \cdot \mathrm{SDP})$, where $D = O_\varepsilon(\sqrt{\log n \log(1/\rho_{min})})$ is the distortion of orthogonal separators, $\rho_{min} = \min_i \rho_i$, and SDP is the cost of the SDP solution. If the SDP solution is optimal, then the expected number of cut edges is at most $O(D \cdot \mathrm{OPT})$.*

Further, there is a variant of this algorithm for graphs excluding a fixed minor. The algorithm finds a partition $P_1, \dots, P_k$ of cost at most $O(D \cdot \mathrm{OPT})$, where $D = O_\varepsilon(1)$ (the constant depends on the excluded minor); the partition satisfies $\mu_i(P_i) \leq 5(1+\varepsilon)\rho_i$.

Remark 2.1. *The algorithm $\mathcal{A}$ from Theorem 2.1 is a randomized algorithm: it always finds a feasible solution (a solution with $\mu_i(P_i) \leq 5(1+\varepsilon)\rho_i$), the expected cost of the solution is $\alpha_{\mathcal{A}}\mathrm{SDP} = O(D \cdot \mathrm{OPT})$ (where $\alpha_{\mathcal{A}} = O(D)$), and the expected number of iterations the algorithm performs is upper bounded by $4n^2k$. The algorithm can be easily converted to an algorithm $\mathcal{A}'$ that always runs in polynomial-time and that succeeds with high probability. If it succeeds, it outputs a feasible solution of cost $O(D \cdot \mathrm{OPT})$; if it fails, it outputs $\perp$ ($\perp$ is a special symbol that indicates that the algorithm failed). The algorithm $\mathcal{A}'$ works as follows. It executes $\mathcal{A}$. If $\mathcal{A}$ does not stop after $4n^4k$ iterations, $\mathcal{A}'$ terminates and outputs $\perp$. Otherwise, it compares the value of the solution that $\mathcal{A}$ found with $3\alpha_{\mathcal{A}}\,\mathrm{SDP}$: If the cost is less than $3\alpha_{\mathcal{A}}\,\mathrm{SDP}$, the algorithm outputs the solution; otherwise it outputs $\perp$. Clearly the algorithm always runs in polynomial time, and if it succeeds it finds a solution of cost at most $3\alpha_{\mathcal{A}}\,\mathrm{OPT} = O(D \cdot \mathrm{OPT})$. By Markov's inequality, the probability that the algorithm fails is at most $1/n^2 + 1/3 < 1/2$. By running the algorithm n times, we can make the failure probability exponentially small (note that we need the algorithm to succeed at least once).*

As we mentioned earlier, the algorithm always returns a valid partition. We need to verify that the algorithm terminates in expected polynomial time, and that it produces cuts of expected cost at most $O(D \cdot \mathrm{SDP})$.

The state of the algorithm at iteration t is determined by the sets $A(t)$, $P_1(t), \dots, P_k(t)$ and the depths of the elements. We denote the state by $\mathcal{C}(t) = \{A(t), P_1(t), \dots, P_k(t), \mathrm{depth}(t)\}$. Observe that the probability distribution of the index i and set $S_i(t)$ the algorithm picks at iteration t does not depend on t. Thus, the probability that the algorithm is in the state $\mathcal{C}^*$ at iteration $(t+1)$ depends only on the state of the algorithm at iteration t. That is, for every two states $\mathcal{C}^*$ and $\mathcal{C}^{**}$ and every two iterations t_1 and t_2, we have

$$\Pr(\mathcal{C}(t_1+1) = \mathcal{C}^* \mid \mathcal{C}(t_1) = \mathcal{C}^{**}) = \Pr(\mathcal{C}(t_2+1) = \mathcal{C}^* \mid \mathcal{C}(t_2) = \mathcal{C}^{**}).$$

DRAFT

Thus, the states of the algorithm form a Markov chain. The number of possible states is finite (since the depth of every vertex is bounded by n). To simplify the notation, we assume that for $t \geq T$, $\mathcal{C}(t) = \mathcal{C}(T)$. This is consistent with the definition of the algorithm — if we did not stop the algorithm at time T, it would simply idle, since $A(t) = \varnothing$, and thus $S_i(t) \cap A(t) = \varnothing$ for $t \geq T$.

We are interested in the probability that an inactive vertex u which lies *in the top layer* of one of the bins (i.e., $u \notin A(t)$ and $\operatorname{depth}_u(t) = 1$) is removed from that bin within m iterations. Consider a state $\mathcal{C}^*$ in which a vertex u lies in the top layer of bin i. We let

$$f(m, u, \mathcal{C}^*) = \Pr(\exists t \in [t_0, t_0 + m] \text{ s.t. } u \in A(t) \mid \mathcal{C}(t_0) = \mathcal{C}^*).$$

That is, $f(m, u, \mathcal{C}^*)$ is the probability that u is removed from bin i at one of the iterations $t \in [t_0, t_0 + m]$ given that at iteration t_0 the state of the algorithm is $\mathcal{C}^*$. Note that the probability above does not depend on t_0 and thus $f(m, u, \mathcal{C}^*)$ is well defined. Let $\mathcal{U}(u, i)$ be the set of all states $\mathcal{C} = \{A, P_1, \dots, P_k, \operatorname{depth}\}$ such that $u \in P_i$ and $\operatorname{depth}_u = 1$. Let

$$f(m) = \max_{\substack{u \in V \\ i \in \{1,\dots,k\}}} \max_{\mathcal{C}^* \in \mathcal{U}(u,i)} f(m, u, \mathcal{C}^*).$$

Our first lemma gives a bound, in terms of $f(m)$, on the expected number of iterations at which a vertex u is active.

Lemma 2.2. *For every possible state of the algorithm $\mathcal{C}^*$, every vertex u, natural number t_0, and $m \geq 1$,*

$$\sum_{t=t_0}^{t_0+m} \Pr(u \in A(t) \mid \mathcal{C}(t_0) = \mathcal{C}^*) \leq \frac{k}{(1 - 2\delta)\alpha(1 - f(m - 1))}. \tag{7}$$

Proof. The left-hand side of inequality (7) equals the expected number (conditioned on $\mathcal{C}(t_0) = \mathcal{C}^*$) of iterations t in the interval $[t_0, t_0 + m]$ at which u is active, i.e., $u \in A(t)$. The goal is to upper bound this quantity.

We split the interval $[t_0, t_0 + m]$ into alternating intervals at which u is active and at which u is inactive. We call former intervals *active intervals*. We denote the length of the j-th active interval by a_j. If there are fewer than j active intervals in $[t, t_0 + m]$, then we let $a_j = 0$. We need to bound $\sum_{j=1}^{\infty} \mathbb{E}[a_j \mid \mathcal{C}(t_0) = \mathcal{C}^*]$.

$$\sum_{j=1}^{\infty} \mathbb{E}[a_j \mid \mathcal{C}(t_0) = \mathcal{C}^*] = \sum_{j=1}^{\infty} \mathbb{E}[a_j \mid a_j > 0, \mathcal{C}(t_0) = \mathcal{C}^*] \Pr(a_j > 0 \mid \mathcal{C}(t_0) = \mathcal{C}^*).$$

We first estimate $\mathbb{E}[a_j \mid a_j > 0, \mathcal{C}(t_0) = \mathcal{C}^*]$, i.e. the expected length of the j-th active interval given that the j-th active interval exists. At every iteration t when u is active, u is thrown into bin P_i with probability at least $(1 - \delta)\alpha\|\bar{u}_i\|^2$ by property 2 of orthogonal separators. Hence, it is thrown into one of the bins with probability at least (here, we use that the SDP solution satisfies constraint ($6'$))

$$\frac{1}{k} \sum_{i=1}^{k} (1 - \delta)\alpha\|\bar{u}_i\|^2 \geq \frac{(1 - 2\delta)\alpha}{k}.$$

So the expected number of iterations passed since u becomes active till u is put into one of the bins and thus becomes inactive is at most $k/((1 - 2\delta)\alpha)$. We get the bound $\mathbb{E}[a_j \mid a_j > 0, \mathcal{C}(t_0) = \mathcal{C}^*] \leq k/((1 - 2\delta)\alpha)$.

DRAFT

We estimate $\Pr(a_j > 0 \mid \mathcal{C}(t_0) = \mathcal{C}^*)$. For $j = 1$, we use the trivial bound $\Pr(a_1 > 0 \mid \mathcal{C}(t_0) = \mathcal{C}^*) \leq 1$. To get a bound for $j > 1$, we consider $\Pr(a_j > 0 \mid a_{j-1} > 0, \mathcal{C}(t_0) = \mathcal{C}^*)$ – the conditional probability that the interval $[t_0, t_0 + m]$ contains at least j active intervals given that it contains at least $j-1$ active intervals. Assume $a_{j-1} > 0$. Let τ be the right end of the $(j-1)$-st active interval. If $\tau < t_0 + m$, then u is active at the iteration τ and inactive at the iteration $\tau + 1$. Therefore, if $\tau < t_0 + m$, then for some bin i, we have $u \in P_i(\tau + 1)$ and $\mathrm{depth}_u(\tau + 1) = 1$. The probability that u is reactivated till iteration $t_0 + m$, i.e., the probability that for some $\tau' \in [(\tau + 1), t_0 + m]$, $u \in A(\tau')$ is at most $f(m-1)$, since the length of the interval $[(\tau + 1), t_0 + m]$ is at most $(m-1)$. Consequently, $\Pr(a_j > 0 \mid a_{j-1} > 0, \mathcal{C}(t_0) = \mathcal{C}^*) \leq f(m-1)\Pr(\tau < m \mid \mathcal{C}(t_0) = \mathcal{C}^*) \leq f(m-1)$. Hence, for $j \geq 1$, we have

$$\Pr(a_j > 0 \ \mid \ \mathcal{C}(t_0) = \mathcal{C}^*) \leq \prod_{j'=1}^{j-1} \Pr(a_{j'+1} > 0 \mid a_{j'} > 0) \leq f(m-1)^{j-1}.$$

Combining the bounds on $\mathbb{E}[a_j \mid a_j > 0, \mathcal{C}(t_0) = \mathcal{C}^*]$ and $\Pr(a_j > 0, \mathcal{C}(t_0) = \mathcal{C}^*)$ we get the following inequality

$$\sum_{j=1}^{\infty} \mathbb{E}[a_j \ \mid \ \mathcal{C}(t_0) = \mathcal{C}^*] \leq \sum_{j=1}^{\infty} \frac{k}{(1-2\delta)\alpha} f(m-1)^{j-1} = \frac{k}{(1-2\delta)\alpha(1 - f(m-1))}.$$

□

We now show that $f(m) \leq 1/2$ for all m.

Lemma 2.3. *For all natural* m, $f(m) \leq 1/2$.

Proof. We prove this lemma by induction on m. For $m = 0$, the statement is trivial as $f(0) = 0$.

Consider an arbitrary state $\mathcal{C}^*$, bin i^*, vertex u, and iteration t_0. Suppose that $\mathcal{C}(t_0) = \mathcal{C}^*$, $u \in P_{i^*}(t_0)$ and $\mathrm{depth}_u(t_0) = 1$, i.e., u lies in the top layer in bin i^*. We need to estimate the probability that u is removed from bin i^* till iteration $t_0 + m$. The vertex u is removed from bin i^* if and only if at some iteration $t \in \{t_0, \dots, t_0 + m - 1\}$, u is "pushed away" from the bin by new vertices (see Step 2 of the algorithm). This happens only if the weight of vertices added to bin i^* at iterations $\{t_0, \dots, t_0 + m - 1\}$ plus the weight of vertices in the first layer of the bin at iteration t_0 exceeds $5(1+\varepsilon)\rho_i$. Since the weight of vertices in the first layer is at most $(1+\varepsilon)\rho_i$, the weight of vertices added to bin i^* at iterations $\{t_0, \dots, t_0 + m - 1\}$ must be greater than $4(1+\varepsilon)\rho_{i^*}$.

We compute the expected weight of vertices thrown in bin i^* at iterations $t \in \{t_0, \dots, t_0 + m - 1\}$. Let us introduce some notation: $M = \{t_0, \dots, t_0 + m - 1\}$; $i(t)$ is the index i chosen by the algorithm at the iteration t. Let X_{M,i^*} be the weight of the vertices thrown in bin i^* at iterations $t \in M$. Then,

$$\begin{aligned} \mathbb{E}\big[X_{M,i^*} \mid \mathcal{C}(t_0) = \mathcal{C}^*\big] &= \mathbb{E}\Big[\sum_{\substack{t \in M \\ s.t.\ i(t) = i^*}} \mu_{i^*}\big(S_{i^*}(t) \cap A(t)\big) \mid \mathcal{C}(t_0) = \mathcal{C}^*\Big] \\ &= \sum_{t \in M} \sum_{v \in V} \Pr\big(i(t) = i^* \text{ and } v \in S_{i^*}(t) \cap A(t) \ \mid \ \mathcal{C}(t_0) = \mathcal{C}^*\big)\mu_{i^*}(v). \end{aligned} \tag{8}$$

The event $\{i(t) = i^* \text{ and } v \in S_{i^*}(t)\}$ is independent from the event $\{v \in A(t) \text{ and } \mathcal{C}(t_0) = \mathcal{C}^*\}$. Thus,

$$\begin{aligned} &\Pr\big(i(t) = i^* \text{ and } v \in S_{i^*}(t) \cap A(t) \ \mid \ \mathcal{C}(t_0) = \mathcal{C}^*\big) \\ &\qquad = \Pr\big(i(t) = i^* \text{ and } v \in S_{i^*}(t)\big) \cdot \Pr\big(v \in A(t) \ \mid \ \mathcal{C}(t_0) = \mathcal{C}^*\big). \end{aligned}$$

DRAFT

Since $i(t)$ is chosen uniformly at random in $\{1, \dots, k\}$, we have $\Pr(i(t) = i^*) = 1/k$. Then, by property 2 of orthogonal separators, $\Pr(v \in S_{i^*}(t) \mid i(t) = i^*) \le \alpha \|\bar{v}_{i^*}\|^2$. We get

$$\Pr\big(i(t) = i^* \text{ and } v \in S_{i^*}(t) \cap A(t) \mid \mathcal{C}(t_0) = \mathcal{C}^*\big) \le \frac{\alpha \|\bar{v}_{i^*}\|^2}{k} \cdot \Pr\big(v \in A(t) \mid \mathcal{C}(t_0) = \mathcal{C}^*\big).$$

We now plug this expression in (8) and use Lemma 2.2,

$$\begin{aligned}
\mathbb{E}[X_{M,i^*} \mid \mathcal{C}(t_0) = \mathcal{C}^*] &\le \sum_{v \in V} \frac{\alpha \|\bar{v}_{i^*}\|^2 \mu_{i^*}(v)}{k} \cdot \sum_{t \in M} \Pr\big(v \in A(t) \mid \mathcal{C}(t_0) = \mathcal{C}^*\big) \\
&\le \sum_{v \in V} \frac{\alpha \|\bar{v}_{i^*}\|^2 \mu_{i^*}(v)}{k} \cdot \frac{k}{(1-2\delta)\alpha(1 - f(m-1))} \\
&= \sum_{v \in V} \frac{\|\bar{v}_{i^*}\|^2 \mu_{i^*}(v)}{(1-2\delta)(1 - f(m-1))}.
\end{aligned}$$

Finally, observe that $1 - f(m-1) \ge 1/2$ by the inductive hypothesis, and $\sum_{v \in V} \|\bar{v}_{i^*}\|^2 \mu_{i^*}(v) \le \rho_{i^*}$ by the SDP constraint (2). Hence, $\mathbb{E}[X_{M,i^*} \mid \mathcal{C}(t_0) = \mathcal{C}^*] \le 2\rho_{i^*}/(1-2\delta)$. By Markov's inequality,

$$\Pr\big(X_{M,i^*} \ge 4(1+\varepsilon)\rho_{i^*}\big) \le \frac{2\rho_{i^*}}{4(1-2\delta)(1+\varepsilon)\rho_{i^*}} \le \frac{1}{2},$$

since $\delta = \varepsilon/4$. This concludes the proof of Lemma 2.3. □

As an immediate corollary of Lemmas 2.2 and 2.3, we get that for all $u \in V$,

$$\sum_{t=0}^{\infty} \Pr(u \in A(t)) = \lim_{m \to \infty} \sum_{t=0}^{m} \Pr(u \in A(t)) \le \frac{2k}{(1-2\delta)\alpha} \le \frac{4k}{\alpha}. \tag{9}$$

Proof of Theorem 2.1. We now prove Theorem 2.1. We first bound the expected running time. At every iteration of the algorithm $t < T$, the set $A(t)$ is not empty. Hence, using (9), we get

$$\mathbb{E}[T] \le \mathbb{E}\Big[\sum_{t=0}^{\infty} |A(t)|\Big] = \sum_{v \in V} \sum_{t=0}^{\infty} \Pr(v \in A(t)) \le n \cdot \frac{4k}{\alpha} = 4kn^2.$$

We now upper bound the expected size of the cut. For every edge $(u, v) \in E$, we estimate the probability that (u, v) is cut. Suppose that (u, v) is cut. Then, u and v belong to distinct sets $P_i(T)$. Consider the iteration t at which u and v are separated the first time. A priori, there are two possible cases:

1. At iteration t, u and v are active, but only one of the vertices u or v is added to some set $P_i(t+1)$; the other vertex remains in the set $A(t+1)$.

2. At iteration t, u and v are in some set $P_i(t)$, but only one of the vertices u or v is removed from the set $P_i(t+1)$.

It is easy to see that, in fact, the second case is not possible, since if u and v were never separated before iteration t, then u and v must have the same depth (i.e., $\mathrm{depth}_u(t) = \mathrm{depth}_v(t)$) and thus u and v may be removed from bin i only together.

DRAFT

Consider the first case, and assume that $u \in P_{i(t)}(t+1)$ and $v \in A(t+1)$. Here, as in the proof of Lemma 2.3, we denote the index i chosen at iteration t by $i(t)$. Since $u \in P_{i(t)}(t+1)$ and $v \in A(t+1)$, we have $u \in S_{i(t)}(t)$ and $v \notin S_{i(t)}(t)$. Write

$$\begin{aligned}
\Pr(u, v \in A(t);\, u \in S_{i(t)}(t);\; v \notin S_{i(t)}(t)) = \\
&= \Pr(u, v \in A(t)) \cdot \Pr(u \in S_{i(t)}(t);\; v \notin S_{i(t)}(t)) \\
&= \Pr(u, v \in A(t)) \cdot \sum_{i=1}^{k} \Pr(u \in S_i(t);\; v \notin S_i(t) \mid i(t) = i) \cdot \Pr(i(t) = i) \\
&= \Pr(u, v \in A(t)) \cdot \sum_{i=1}^{k} \frac{\Pr(u \in S_i(t);\; v \notin S_i(t) \mid i(t) = i)}{k}.
\end{aligned}$$

We bound $\Pr(u, v \in A(t)) \leq \Pr(u \in A(t))$ and use the third property of orthogonal separators, $\Pr(u \in S_i(t);\; v \notin S_i(t)) \leq \alpha D \,\|\bar{u}_i - \bar{v}_i\|^2$, to get

$$\Pr(u, v \in A(t);\; u \in S_{i(t)}(t);\; v \notin S_{i(t)}(t)) \leq \Pr(u \in A(t)) \cdot \Big(\frac{1}{k} \sum_{i=1}^{k} \alpha D \,\|\bar{u}_i - \bar{v}_i\|^2\Big).$$

By swapping the roles of u and v, we get a bound on $\Pr(u, v \in A(t),\; v \in S_{i(t)}(t),\; u \notin S_{i(t)}(t))$. Combining the bounds, we get that the probability that u and v are separated at iteration t is upper bounded by $\Big(\Pr(u \in A(t)) + \Pr(v \in A(t))\Big) \cdot \Big(\frac{1}{k}\sum_{i=1}^{k} \alpha D \;\|\bar{u}_i - \bar{v}_i\|^2\Big)$. The probability that the edge (u, v) is ever cut is at most

$$\begin{aligned}
\Big(\sum_{t=0}^{\infty} \Pr(u \in A(t)) + \Pr(v \in A(t))\Big) \cdot \Big(\frac{1}{k} \sum_{i=1}^{k} \alpha D \,\|\bar{u}_i - \bar{v}_i\|^2\Big) \leq \\
\leq \frac{8k}{\alpha} \Big(\frac{1}{k} \sum_{i=1}^{k} \alpha D \,\|\bar{u}_i - \bar{v}_i\|^2\Big) = 8 \sum_{i=1}^{k} D \,\|\bar{u}_i - \bar{v}_i\|^2.
\end{aligned}$$

Here, we used inequality (9) to bound the first term on the left-hand side. The desired bound on the expected number of cut edges follows:

$$\mathbb{E}[\text{number of cut edges}] = \sum_{(u,v)\in E} \Pr((u, v) \text{ is cut}) \leq 8 \sum_{(u,v)\in E} \sum_{i=1}^{k} D \,\|\bar{u}_i - \bar{v}_i\|^2 = 16D \cdot \mathrm{SDP}.$$

□

3 $O(\sqrt{\log n \log k})$ approximation

Informal Discussion. In this section, we describe a bi-citeria $O_\varepsilon(\sqrt{\log n \log k})$ approximation algorithm for Minimum Nonuniform Graph Partitioning. In Theorem 2.1, we showed how to get an $O_\varepsilon(\sqrt{\log n \log 1/\rho_{min}})$-approximation. Thus, to get an $O_\varepsilon(\sqrt{\log n \log k})$ approximation, it suffices to modify our instance so that $\rho'_{min} \geq 1/poly(k)$. Let $P_1^*, \dots, P_k^*$ be the unknown optimal partition. Consider i for which $\rho_i < \delta/k$. Suppose for a moment that we could guess a set A_i containing P_i^* such that $\mu_i(P_i^*)/\mu_i(A_i) \geq \delta/k$. Then, we

DRAFT

would restrict P_i to be a subset of A_i: We would define a new "conditional" measure $\mu_i'(u) = \mu_i(u)/\mu_i(A_i)$ for $u \in A_i$ and $\mu_i'(u) = 0$ for $u \notin A_i$ and let $\rho_i' = \rho_i/\mu_i(A_i)$. We would also set $\bar{u}_i = 0$ for all $u \notin A_i$ thus forcing the algorithm to pick vertices for P_i from the set A_i. Note that in the integral solution $\bar{u}_i = 0$ for $u \notin A_i$ so $\bar{u}_i = 0$ is a valid constraint. Since $\mu_i(P_i^*)/\mu_i(A_i) \geq \delta/k$, we get

$$\rho_i' = \rho_i/\mu_i(A_i) \geq \mu(P_i^*)/\mu_i(A_i) \geq \delta/k.$$

By applying the above transformation to all i with $\rho_i < \delta/k$, we would get an instance with $\rho_{min} \geq \delta/k$. It is easy to check that any feasible solution to the new problem is a feasible solution to the original problem as well.

Of course, the challenge is that we cannot guess such set A_i. Instead, we solve the SDP and define A_i using the SDP solution. Roughly speaking, we let $A_i = \{u : \|\bar{u}_i\|^2 \geq \delta/k\}$. Then, as before, we redefine the measure μ_i and zero out vectors $\bar{u}_i$ with $u \notin A_i$. The analogue of the measure of the optimal set P_i^* will be now the measure of the fractional solution defined as $\sum_{u \in A_i} \frac{\mu_i(u)\|\bar{u}_i\|^2}{\mu(A_i)}$. We have

$$\sum_{u \in A_i} \mu_i(u)\|\bar{u}_i\|^2 \geq \sum_{u \in A_i} \mu_i(u) \cdot \frac{\delta}{k} = \frac{\delta}{k} \cdot \mu(A_i).$$

That is, the measure of the fractional solution is at most δ/k times the measure of A_i. This is a sufficient condition for the algorithm from Theorem 2.1 (see details below). The transformed SDP solution does not satisfy constraint (6), but it satisfies constraint (6′) which also suffices for our algorithm. The only remaining problem is that by zeroing out some vectors $\bar{u}_i$ we can increase the SDP value. We take care of this issue by picking a random threshold $\theta \approx \delta$ and letting $A_i = \{u : \|\bar{u}_i\|^2 \geq \theta/k\}$. We will now proceed to a formal proof.

Theorem 3.1. *There is a randomized polynomial-time algorithm that given a feasible instance of Minimum Nonuniform Graph Partitioning with unrelated weights returns a partition $P_1, \dots, P_k$ of V satisfying $\mu_i(P_i) \leq 5(1+\varepsilon)\rho_i$ such that the expected number of cut edges is at most $O(D \cdot \mathrm{OPT})$, where $D = O_\varepsilon(\sqrt{\log n \log k})$.*

Proof. We perform three steps. First we solve the SDP relaxation, then transform its solution and change measures μ_i, and finally apply Theorem 2.1 to the obtained SDP solution.

We start with describing how we transform the solution. We set $\delta = \varepsilon/4$ as before. Then we choose a threshold θ uniformly at random from $[\delta/2, \delta]$. We let $\tilde{u}_i = \bar{u}_i$ if $\|\bar{u}_i\|^2 \geq \theta/k$ and $\tilde{u}_i = 0$, otherwise. It is immediate that the solution $\tilde{u}_i$ satisfies all SDP constraints except possibly constraint (6). Note, however, that it satisfies constraint (6′):

$$\sum_{i=1}^{k} \|\tilde{u}_i\|^2 = \sum_{i=1}^{k} \|\bar{u}_i\|^2 - \sum_{i:\|\tilde{u}_i\|^2 < \theta/k} \|\tilde{u}_i\|^2 = 1 - \sum_{i:\|\tilde{u}_i\|^2 < \theta/k} \|\tilde{u}_i\|^2 \in [1-\delta, 1].$$

Consider two vertices u and v. Assume without loss of generality that $\|\bar{u}_i\|^2 \leq \|\bar{v}_i\|^2$. By SDP constraint (5), $\|\bar{v}_i\|^2 - \|\bar{u}_i\|^2 \leq \|\bar{u}_i - \bar{v}_i\|^2$. If either $\|\bar{u}_i\|^2 \leq \|\bar{v}_i\|^2 < \theta/k$ or $\theta/k \leq \|\bar{u}_i\|^2 \leq \|\bar{v}_i\|^2$, then we have $\|\tilde{u}_i - \tilde{v}_i\| \leq \|\bar{u}_i - \bar{v}_i\|$. Otherwise, $\|\bar{u}_i\|^2 < \theta/k \leq \|\bar{v}_i\|^2$ and

$$\|\tilde{u}_i - \tilde{v}_i\|^2 = \|\bar{v}_i\|^2 \leq \|\bar{u}_i - \bar{v}_i\|^2 + \|\bar{u}_i\|^2 \leq \|\bar{u}_i - \bar{v}_i\|^2 + \delta/k.$$

Therefore,

$$\mathbb{E}[\|\tilde{u}_i - \tilde{v}_i\|^2] \leq \|\bar{u}_i - \bar{v}_i\|^2 + (\delta/k)\Pr\left(\|\bar{u}_i\|^2 < \theta/k \leq \|\bar{v}_i\|^2\right).$$

DRAFT

To upper bound $\Pr\left(\|\bar{u}_i\|^2 < \theta/k \leq \|\bar{v}_i\|^2\right)$, note that the random variable θ is distributed uniformly on $(\delta/2, \delta)$, so its probability density is bounded from above by $2/\delta$. Thus,

$$\Pr\left(\|\bar{u}_i\|^2 < \theta/k \leq \|\bar{v}_i\|^2\right) \leq (2k/\delta) \cdot (\|\bar{v}_i\|^2 - \|\bar{u}_i\|^2) \leq (2k/\delta) \cdot \|\bar{u}_i - \bar{v}_i\|^2,$$

We have,

$$\mathbb{E}[\|\tilde{u}_i - \tilde{v}_i\|^2] \leq \|\bar{u}_i - \bar{v}_i\|^2 + (\delta/k) \cdot (2k/\delta) \cdot \|\bar{u}_i - \bar{v}_i\|^2 = 3\|\bar{u}_i - \bar{v}_i\|^2.$$

We conclude that the expected SDP value of solution $\tilde{u}_i$ is at most $3\,\mathrm{SDP} \leq 3\,\mathrm{OPT}$ (where SDP is the cost of the original SDP solution).

Now we modify measures μ_i and capacities c_i. For every $i \in [k]$, let $A_i = \{u : \tilde{u}_i \neq 0\}$. Define

$$\mu'_i(Z) = \mu_i(Z \cap A_i)/\mu_i(A_i) \text{ for } Z \subseteq V,$$
$$\tilde{\rho}_i = \rho_i/\mu_i(A_i)$$

(if $\mu_i(A_i) = 0$ we let $\tilde{\mu}_i = \mu_i$ and $\tilde{\rho}_i = 1$, essentially removing the capacity constraint for P_i). We have $\tilde{\mu}_i(V) = \mu_i(A_i)/\mu_i(A_i) = 1$. By (2), we get

$$\rho_i \geq \sum_{u \in V} \|\bar{u}_i\|^2 \mu_i(u) \geq \sum_{u \in V} \|\tilde{u}_i\|^2 \mu_i(u) = \sum_{u \in A_i} \|\tilde{u}_i\|^2 \mu_i(u) \geq \sum_{u \in A_i} \frac{\delta}{2k} \cdot \mu_i(u) = \frac{\delta \mu_i(A_i)}{2k}.$$

Therefore, $\tilde{\rho}_i = \rho_i/\mu_i(A_i) \geq \delta/(2k)$, and $\tilde{\rho}_{min} = \min \tilde{\rho}_i \geq \delta/(2k) \geq \varepsilon/(8k)$ (if $\mu_i(A_i) = 0$ then $\tilde{\rho}_i = 1 > \delta/(2k)$).

Note that since each ρ_i increases by a factor of $1/\mu_i(A_i)$ and each $\mu_i(u)$ increases by a factor at most $1/\mu_i(A_i)$, vectors $\tilde{u}_i$ satisfy SDP constraints (2) and (3), in which μ_i and ρ_i are replaced with $\tilde{\mu}_i$ and $\tilde{\rho}_i$, respectively (assuming that $\mu_i(A_i) \neq 0$; if $\mu_i(A_i) = 0$, the constraints clearly hold). We run the algorithm from Theorem 2.1 on vectors $\tilde{u}_i$ with measures $\tilde{\mu}_i$ and capacities $\tilde{\rho}_i$. The algorithm finds a partition $P_1, \dots, P_k$ that cuts at most $D \cdot \mathrm{SDP} \leq D \cdot \mathrm{OPT}$ edges, where $D = O_\varepsilon(\sqrt{\log n \log(1/\tilde{\rho}_{min})}) = O_\varepsilon(\sqrt{\log n \log k})$. We verify that the weight of each set P_i is $O(\rho_i)$. Note that $P_i \subset A_i$ since for $u \notin A_i$, $\|\tilde{u}_i\|^2 = 0$, and thus the algorithm does not add u to P_i. We have,

$$\mu_i(P_i) = \mu'_i(P_i \cap A_i) \cdot \mu_i(A_i) = \mu'_i(P_i) \cdot \mu_i(A_i) \leq 5(1+\varepsilon)\tilde{\rho}_i \cdot \mu_i(A_i) \leq 5(1+\varepsilon)\rho_i.$$

□

4 Partitioning with d-Dimensional Weights

We describe how Minimum Nonuniform Graph Partitioning with unrelated d-dimensional weights reduces to Minimum Nonuniform Graph Partitioning with unrelated weights. Consider an instance $\mathcal{I}$ of Minimum Nonuniform Graph Partitioning with unrelated d-dimensional weights. Let $\mu'_i(u) = \max_j (r_j(u,i)/c_j(i))$. Then define measures $\mu_i(u)$ and capacities $\rho_i(u)$ by

$$\mu_i(u) = \mu'_i(u)/\mu'_i(V) \quad \text{and} \quad \rho_i = d/\mu'_i(V).$$

We obtain an instance $\mathcal{I}'$. Note that the optimal solution $P_1^*, \dots, P_k^*$ for $\mathcal{I}$ is a feasible solution for $\mathcal{I}'$ since

$$\mu_i(P_i^*) = \sum_{u \in P_i^*} \frac{\mu'_i(u)}{\mu'_i(V)} = \frac{1}{\mu'_i(V)} \sum_{u \in P_i^*} \max_j \frac{r_j(u,i)}{c_j(i)} \leq \frac{1}{\mu'_i(V)} \sum_{u \in P_i^*} \sum_{j=1}^{d} \frac{r_j(u,i)}{c_j(i)}$$
$$= \frac{1}{\mu'_i(V)} \sum_{j=1}^{d} \sum_{u \in P_i^*} \frac{r_j(u,i)}{c_j(i)} \leq \frac{d}{\mu'_i(V)} = \rho_i.$$

DRAFT

We solve instance $\mathcal{I}'$ and get a partition $P_1, \ldots, P_k$ that cuts at most $O(\sqrt{\log n \log k}\,\mathrm{OPT})$ edges. The partition satisfies d-dimensional capacity constraints:

$$\sum_{u \in P_i} r_j(u,i) \leq \sum_{u \in P_i} c_j(i)\mu_i'(u) = c_j(i)\mu_i'(V) \sum_{u \in P_i} \mu_i(u) \leq c_j(i)\mu_i'(V)(5(1+\varepsilon)\rho_i) = 5d(1+\varepsilon)\, c_j(i).$$

This concludes the analysis of the reduction.

5 Orthogonal Separators

In this section, we prove Theorem 5.1 for completeness of exposition.

Theorem 5.1 (Bansal et al. (2014))**.** *There exists a polynomial-time algorithm that given a graph $G = (V,E)$, a measure μ on V ($\mu(V) = 1$), parameters $\rho, \varepsilon, \delta \in (0,1)$ and a collection of vectors $\bar{u}$ satisfying the following constraints:*

$$\sum_{u \in V} \|\bar{u}\|^2 \mu(u) \leq \rho \tag{10}$$

$$\sum_{v \in V} \langle \bar{u}, \bar{v} \rangle \mu(v) \leq \|\bar{u}\|^2 \rho \qquad \text{for all } u \in V \tag{11}$$

$$\|\bar{u} - \bar{v}\|^2 + \|\bar{v} - \bar{w}\|^2 \geq \|\bar{u} - \bar{w}\|^2 \qquad \text{for all } u, v, w \in V \tag{12}$$

$$0 \leq \langle \bar{u}, \bar{v} \rangle \leq \|\bar{u}\|^2 \qquad \text{for all } u, v \in V \tag{13}$$

$$\|\bar{u}\|^2 \leq 1 \qquad \text{for all } u \in V \tag{14}$$

outputs a random set $S \subset V$ (“orthogonal separator”) such that

1. *$\mu(S) \leq (1+\varepsilon)\rho$ (always);*
2. *For all u, $\Pr(u \in S) \in [(1-\delta)\alpha\|\bar{u}\|^2, \alpha\|\bar{u}\|^2]$;*
3. *For all $(u,v) \in E$, $\Pr(u \in S, v \notin S) \leq \alpha D \cdot \|\bar{u} - \bar{v}\|^2$.*

Where the probability scale $\alpha = 1/n$, and the distortion $D \leq O_\varepsilon(\sqrt{\log n \log(1/(\rho\delta))})$. For graphs with excluded minors, $D = O_\varepsilon(1)$.

Remark 5.1. *Note that if vectors $\{\bar{u}_i : u \in V, i \in [k]\}$ satisfy constraints (2)–(6), then for every i, vectors $\{\bar{u} \equiv \bar{u}_i : u \in V\}$ satisfy constraints (10)–(14) with $\mu(u) = \mu_i(u)$ and $\rho = \rho_i$. In particular, constraint (14) follows from (6).*

In Chlamtac, Makarychev, and Makarychev (2006), we showed that for every $\beta > 0$ there exists a randomized polynomial-time algorithm that given G, SDP solution $\{\bar{u}\}$ satisfying (12) – (14), and a parameter $m > 0$, outputs a random set S with the following properties (see also Bansal et al. (2014) and Louis and Makarychev (2014)).

O1. For all $u \in V$, $\Pr(u \in S) = \alpha\,\|\bar{u}\|^2$, where $\alpha = 1/n$.

O2. For all $u, v \in V$ with $\|\bar{u} - \bar{v}\|^2 \geq \beta \min(\|\bar{u}\|^2, \|\bar{v}\|^2)$,

$$\Pr(u \in S \text{ and } v \in S) \leq \frac{\alpha \min(\|\bar{u}\|^2, \|\bar{v}\|^2)}{m}.$$

DRAFT

O3. For all $(u,v) \in E$,

$$\Pr(u \in S \text{ and } v \notin S) \le \alpha D \cdot \|\bar{u} - \bar{v}\|^2,$$

where $D \le O_\beta(\sqrt{\log n \log m})$.

Bansal et al. (2014) showed that for graphs excluding a fixed minor, there is an algorithm as described above with $D = O_\beta(1)$. Specifically, they proved (see Theorem 2.5 in Bansal et al. (2014)) that there exists an algorithm that given any collection of numbers $\{z(u,v)\}$ and $\{x(u)\}$ in $[0,1]$ satisfying conditions (a) $z(u,v) + z(v,w) \ge z(u,v)$, (b) $|x(u) - x(v)| \le z(u,v)$, and (c) $x(u) + x(v) \ge z(u,v)$, outputs a random set S such that:

O1′. For all $u \in V$, $\Pr(u \in S) = \alpha\, x(u)$, where $\alpha = \Omega(1/n)$.

O2′. For all $u, v \in V$ with $z(u,v) \ge \beta \min(x(u), x(v))$, $\Pr(u \in S \text{ and } v \in S) = 0$.

O3′. For all $(u,v) \in E$, $\Pr(u \in S \text{ and } v \notin S) \le \alpha D \cdot z(u,v)$, where $D = O_\beta(1)$.

Observe that if $\bar{u}$ is an SDP solution satisfying constraints (12)–(14), then the numbers $x(u) = \|\bar{u}\|^2$ and $z(u,v) = \|\bar{u} - \bar{v}\|^2$ satisfy conditions (a), (b), and (c). Hence, for a feasible SDP solution $\{\bar{u}\}$ and $x(u) = \|\bar{u}\|^2$, $z(u,v) = \|\bar{u} - \bar{v}\|^2$, this algorithm outputs a random set S such that O1, O2, and O3 hold with $D = O_\beta(1)$ and arbitrarily large $m > 0$ (or even that $m = \infty$ so that $1/m = 0$).

Below, we will use the algorithm by Chlamtac, Makarychev, and Makarychev (2006) for arbitrary graphs and the algorithm by Bansal et al. (2014) for graphs excluding a fixed minor.

We now describe the algorithm used in Theorem 5.1. The algorithm samples S as above with $m = 2/(\delta\varepsilon\rho)$, $\beta = \varepsilon/4$ (m may be fractional) and outputs $S' = S$ if $\mu(S) \le (1+\varepsilon)\rho$, and $S' = \varnothing$, otherwise. It is clear that $\mu(S') \le (1+\varepsilon)\rho$ (always), and thus the first property in Theorem 5.1 is satisfied. Then, for $(u,v) \in E$,

$$\Pr(u \in S' \text{ and } v \notin S') \le \Pr(u \in S \text{ and } v \notin S) \le \alpha D \cdot \|\bar{u} - \bar{v}\|^2,$$

where $D = O_\beta(\sqrt{\log n \log m}) = O_\varepsilon(\sqrt{\log n \log(1/(\rho\delta))})$ for arbitrary graphs and $D = O_\beta(1)$ for graphs excluding a fixed minor.

For every $u \in V$,

$$\Pr(u \in S') \le \Pr(u \in S) = \alpha \|\bar{u}\|^2.$$

So we only need to verify that $\Pr(u \in S') \ge \alpha(1-\delta)\|\bar{u}\|^2$. We assume $\|\bar{u}\|^2 \ne 0$. We have

$$\Pr(u \in S') = \Pr(u \in S' \mid u \in S) \cdot \Pr(u \in S) = \Pr\big(\mu(S) \le (1+\varepsilon)\rho \mid u \in S\big) \cdot \alpha \|\bar{u}\|^2.$$

We split V into two sets $A_u = \{v : \|\bar{u} - \bar{v}\|^2 \ge \beta \|\bar{u}\|^2\}$ and $B_u = \{v : \|\bar{u} - \bar{v}\|^2 < \beta \|\bar{u}\|^2\}$. We show below (see Lemma 5.2) that $\mu(B_u) \le (1 + \varepsilon/2)\rho$. Then,

$$\mu(S) = \mu(S \cap A_u) + \mu(S \cap B_u) \le (1+\varepsilon/2)\rho + \mu(S \cap A_u)$$

and

$$\Pr(u \in S') \ge \alpha \|\bar{u}\|^2 \cdot \Pr\big(\mu(S \cap A_u) \le \varepsilon\rho/2 \mid u \in S\big). \tag{15}$$

We estimate $\Pr\big(\mu(S \cap A_u) \ge \varepsilon\rho/2 \mid u \in S\big)$. For every $v \in A_u$, $\|\bar{u} - \bar{v}\|^2 \ge \beta \|\bar{u}\|^2$. Thus, for $v \in A_u$, $\Pr(u \in S; v \in S) \le \alpha \|\bar{u}\|^2/m$, and

$$\Pr(v \in S \mid u \in S) = \frac{\Pr(u \in S, v \in S)}{\Pr(u \in S)} \le \frac{1}{m}.$$

DRAFT

Therefore, $\mathbb{E}[\mu(S \cap A_u) \mid u \in S] \le \mu(A_u)/m \le 1/m$, and, by Markov's inequality,

$$\Pr\big(\mu(S \cap A_u) \ge \varepsilon\rho/2 \;\mid\; u \in S\big) \le \frac{\mathbb{E}[\mu(S \cap A_u) \mid u \in S]}{\varepsilon\rho/2} \le \frac{2}{m\varepsilon\rho} = \delta.$$

The last equality holds since $m = 2/(\delta\varepsilon\rho)$. We plug this bound in (15) and get the desired inequality,

$$\Pr(u \in S') \ge \alpha\|\bar{u}\|^2 \cdot (1 - \delta).$$

We now prove Lemma 5.2.

Lemma 5.2. *For every $u \in V$ such that $\bar{u} \neq 0$, $\mu(B_u) \le (1 + \varepsilon/2)\rho$.*

Proof. To prove the lemma, we first derive a lower bound on $\langle \bar{u}, \bar{v} \rangle$ for points $v \in B_u$ and an upper bound on $\sum_{v \in B_u} \langle \bar{u}, \bar{v} \rangle \mu(v)$; combining the bounds we get an upper bound for $\mu(B_u)$. If $v \in B_u$, then by the definition of B_u and by inequality (13), we have

$$\langle \bar{u}, \bar{v} \rangle = \|\bar{u}\|^2 + (\underbrace{\|\bar{v}\|^2 - \langle \bar{u}, \bar{v} \rangle}_{\ge 0}) - \|\bar{u} - \bar{v}\|^2 \ge (1 - \beta)\|\bar{u}\|^2.$$

From constraints (11) and (13),

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

Thus,

$$\mu(B_u) = \sum_{v \in B_u} \mu(v) \le \sum_{v \in B_u} \mu(v) \cdot \frac{\langle \bar{u}, \bar{v} \rangle}{(1 - \beta)\|\bar{u}\|^2} \le \frac{\rho\|\bar{u}\|^2}{(1 - \beta)\|\bar{u}\|^2}$$
$$\le (1 + 2\beta)\rho = (1 + \varepsilon/2)\rho.$$

□

Acknowledgement

The authors thank an anonymous referee for valuable comments and suggestions.

References

A. Amir, J. Ficler, R. Krauthgamer, L. Roditty, and O. Sar Shalom. Multiply Balanced k-Partitioning. LATIN 2014.

N. Bansal, U. Feige, R. Krauthgamer, K. Makarychev, V. Nagarajan, J. Naor, and R. Schwartz. Min-max Graph Partitioning and Small Set Expansion. SIAM Journal on Computing 43.2 (2014): 872-904.

E. Chlamtac, K. Makarychev, and Y. Makarychev. How to Play Unique Games Using Embeddings. FOCS 2006.

R. Krauthgamer, J. Naor, and R. Schwartz. Partitioning graphs into balanced components. SODA 2009.

R. Krauthgamer, J. Naor, R. Schwartz, and K. Talwar. Non-Uniform Graph Partitioning. SODA 2014.

A. Louis and K. Makarychev. Approximation Algorithm for Sparsest k-Partitioning. SODA 2014.

H. Räcke. Optimal hierarchical decompositions for congestion minimization in networks. STOC 2008.

DRAFT