Partitioning Graphs into Balanced Components

Robert Krauthgamer[*] Joseph (Seffi) Naor[†] Roy Schwartz[‡]

Abstract

We consider the *k-balanced partitioning* problem, where the goal is to partition the vertices of an input graph G into k equally sized components, while minimizing the total weight of the edges connecting different components. We allow k to be part of the input and denote the cardinality of the vertex set by n. This problem is a natural and important generalization of well-known graph partitioning problems, including *minimum bisection* and *minimum balanced cut*.

We present a (bi-criteria) approximation algorithm achieving an approximation of $O(\sqrt{\log n \log k})$, which matches or improves over previous algorithms for all relevant values of k. Our algorithm uses a semidefinite relaxation which combines ℓ_2^2 metrics with spreading metrics. Surprisingly, we show that the integrality gap of the semidefinite relaxation is $\Omega(\log k)$ even for large values of k (e.g., $k = n^{\Omega(1)}$), implying that the dependence on k of the approximation factor is necessary. This is in contrast to previous approximation algorithms for k-balanced partitioning, which are based on linear programming relaxations and their approximation factor is independent of k.

1 Introduction

Graph partitioning is a family of optimization problems, where we wish to break an input graph into pieces (satisfying certain constraints), while minimizing the total weight of the edges (or vertices) connecting different parts. A prototypical problem in this family is *k-balanced partitioning*: given an input graph $G = (V, E)$ with edge weights $w : E \to R^+$ and an integer k, partition the graph vertices into k parts of equal size, so as to minimize the total weight of the edges connecting different parts. Here and throughout, we denote $|V| = n$, and note that k may be a function of n. The special case $k = 2$ is the famous *minimum bisection* problem, which is already known to be NP-hard [14]. Consequently, a long line of research has been devoted to polynomial-time approximation of various graph partitioning problems.

Work on approximation algorithms for the minimum bisection problem has proved to be extremely fruitful, exhibiting strong and intrinsic connections to mathematical disciplines such as metric geometry and functional analysis, as well as to computational complexity. It is best illustrated by briefly mentioning a few key results regarding minimum bisection: The $O(\log n)$ bi-criteria approximation by Leighton and Rao [22], as well as the improvement to $O(\sqrt{\log n})$ by Arora, Rao and Vazirani [3], are seminal papers. They approximate minimum bisection via the closely related problem of sparsest cuts. Very recently, Räcke [23] obtained a remarkable $O(\log n)$ (true) approximation, which improves over the previous $O(\log^{1.5} n)$ by Feige and Krauthgamer [12]. Finally, different inapproximability results shown in the sequence of papers [6, 18, 19, 8] and in [11], relate minimum bisection to the unique games conjecture [17] and to refuting random 3SAT formulas, and have transformed the area.

The k-balanced partitioning problem is motivated by a wide range of applications in areas including scientific computing, VLSI design, parallel computing, data mining (clustering), social network analysis (community discovery), and pattern recognition. For example, in VLSI design it is often required to split designs too large to fit in a single chip into *many* smaller parts of (roughly) equal size, where each part is mapped onto a separate chip. Compared to connections within a chip, connections between chips are costly, slow, and power consuming. Hence, it is desirable to reduce the capacity of the cut defined by a partition. In parallel computation applications, a computation is modeled by a graph (i.e., dataflow graph); a vertex models a computation of an expression and an edge models communication of a result of one expression to an operand of another expression. Partitioning determines the mapping of computations to processors. As opposed to cut edges, edges between vertices mapped to the same processor do not require communication between processors. Since communication between processors is slow, it is again desir-

[*] Weizmann Institute of Science, Rehovot, Israel. Part of this work was done while at IBM Almaden, San Jose, CA, USA. This research was supported in part by The Israel Science Foundation (grant #452/08). E-mail: robert.krauthgamer@weizmann.ac.il.

[†] Computer Science Department, Technion, Haifa 32000, Israel. Research supported in part by ISF Grant 1366/07 and US-Israel BSF Grant 2002276. E-mail: naor@cs.technion.ac.il.

[‡] Computer Science Department, Technion, Haifa 32000, Israel. E-mail: schwartz@cs.technion.ac.il.

able to reduce the cut.

Three types of approximation algorithms are known for k-balanced partitioning, and they all provide bi-criteria approximation. Formally, let a (k, ν)-*balanced partition* be a partition of the vertices into k parts, each of size at most $\nu n/k$. Then a bi-criteria algorithm produces a (k, ν)-balanced partition for a fixed $\nu > 1$, but its value is compared against the optimal $(k, 1)$-balanced partition. In fact, Andreev and Räcke [2] proved that bi-criteria approximation is inevitable in the sense that, unless P=NP, k-balanced partitioning has no polynomial time (true) approximation within any finite factor.

The first approximation algorithms for the k-balanced partitioning problem were designed by Leighton, Makedon, and Tragoudas [21] and by Simon and Teng [24], based on recursively finding a minimum bisection. This algorithm achieves an $O(\log k\sqrt{\log n})$ approximation with $\nu = 2$, when using the minimum bisection algorithm due to [3]. (The original paper obtained an $O(\log k \log n)$ approximation with $\nu = 2$ based on Leighton and Rao [22], but the approximation factor improves to $O(\log k\sqrt{\log n})$ by replacing [22] by [3].)

A second algorithm, which was designed by Even, Naor, Rao and Schieber [9], uses a more direct linear programming (LP) relaxation of partitioning problems called *spreading metrics*. A spreading metric is used to repeatedly cut out of the graph small parts, achieving an approximation factor of $O(\log n)$, again with $\nu = 2$. Prior to our work, this was the best approximation ratio when k is at least $k^* = 2^{\sqrt{\log n}}$, but it is inferior to the first approach when $k \ll k^*$.

The third algorithm, given by Andreev and Räcke [2], uses a more sophisticated recursive partitioning together with dynamic programming to obtain an $O(\varepsilon^{-2}\log^{1.5} n)$ approximation with $\nu = 1 + \varepsilon$ for any desired (but fixed) $\varepsilon > 0$. This is the first result for k-balanced partitioning for which $\nu < 2$.

We note that due to the practical importance of the balanced k-partitioning problem, many fast heuristics have been designed for it. In particular, a multi-level graph partitioning heuristic called Metis [16] has been very successful in quickly finding good partitions of small graphs. However, these heuristics do not provide any guarantee on the quality of the solutions computed.

1.1 Results

Theorem 1.1. (Main) *The k-balanced partitioning problem admits polynomial time (bi-criteria) approximation with factor $O(\sqrt{\log k \log n})$ and $\nu = 2$.*

Our approximation factor improves over or matches previous algorithms for all relevant values of k. In particular, our bound degrades gracefully with k, thus offering a smooth transition between the performance previously known for small k (the first algorithm mentioned above) and for large k (the second algorithm above).

It is natural to ask whether the approximation factor in Theorem 1.1 is optimal. Recalling that the case $k = 2$ is minimum bisection, we observe that improving the $\sqrt{\log n}$ term in the approximation factor is simply an established open problem of improving the approximation factor of [3], and would be a breakthrough result. Thus, the main question is whether the dependence on k in our approximation factor is indeed necessary. Past experience has suggested the answer is negative, since Even et al. [9] managed to obtain for general k the same approximation factor as was known at the time for $k = 2$ from [22]. Somewhat surprisingly, we show the converse; we prove that the semidefinite programming (SDP) relaxation of spreading metrics used by our algorithm has an integrality gap that exceeds $O(\sqrt{\log n})$, when k is sufficiently large.

Theorem 1.2. *For every $k \le n^{1-\Omega(1)}$, the SDP relaxation in Theorem 1.1 has integrality gap $\Omega(\log k)$.*

We remark that the remaining gap between the two theorems above falls into a typically hard gray area, and the correct answer might differ from both bounds, e.g. $\Theta(\log k + \sqrt{\log n})$.

1.2 Techniques We formulate a semidefinite programming relaxation for k-balanced partitioning using two well-known ideas. The first one is ℓ_2^2 (a.k.a. negative-type) metrics, which were first used in approximation algorithms by [3], and the second one is spreading metrics (a.k.a spreading constraints), which were introduced by Even, Naor, Rao, and Schieber [10] in the context of approximation algorithms that employ a divide-and-conquer approach. The main challenge we face is designing a rounding procedure that exploits these two types of constraints simultaneously.

A natural approach is to repeatedly cut from the graph small pieces by relying on the spreading constraints, similarly to [9]. They implemented this approach using region growing in the LP relaxation, which can be guaranteed to cut out a piece whose size is not too large, while paying proportionally to the part that was actually cut out; hence, their overall approximation factor matches that of a single cut [22]. In contrast, the basic tool for ℓ_2^2 metrics, the random projection procedure of [3], operates globally and (apparently) cannot cut out only a small piece while paying proportionally to that piece. Hence, using this tool directly requires recursion which introduces a factor of $O(\log k)$ on top of the factor of $O(\sqrt{\log n})$ from [3]. We mitigate this problem using an idea of Chlamtac, Makarychev and Makarychev [7]; we first preprocess the entire SDP solution using [3] to exploit the ℓ_2^2 metric, and only then we repeatedly cut out pieces from the graph, using the spreading constraints to control the size of the pieces.

As it turns out, our first step can use the specific preprocessing step of [7] which, roughly speaking, maps the ℓ_2^2 metric space into ℓ_2 (with some distortion). Our second step repeatedly cuts out pieces from the graph using a random projection procedure that is normalized so that each vertex is cut with probability of at most $1/k$. The probability of separating two vertices (an edge) is proportional to their ℓ_2 distance after the preprocessing, and in turn to their contribution to the SDP value. Since the preprocessing is guaranteed not to shrink too much one scale of distances, we can use the spreading constraints to bound the expected size of the piece cut out from the graph. However, the piece size might not be concentrated around its expectation, and we thus face the problem of dealing with deviations from the expectation. Our solution is to "reject" cuts that separate too large a piece. This might adversely affect the probability of cutting an edge (since we must condition on additional information), but fortunately this probability can still be bounded by a careful calculation of all the dependencies.

Our algorithm is inspired by the notion of *m-orthogonal separators* which was introduced by [7] for the purpose of approximating the unique game problem. An m-orthogonal separator of a given ℓ_2^2 metric is a distribution over its subsets that has the following three properties. First, the probability that a vector is chosen is proportional to its ℓ_2^2 norm. Second, the probability that two vectors are separated by the random subset is bounded by their ℓ_2^2 distance times a factor dependent on m and the total number of vectors. Third, the probability that two *orthogonal* vectors are chosen simultaneously is relatively small – the precise bound also depends on m.

However, our SDP relaxation does not guarantee any orthogonal vectors, thus the third property is irrelevant. Additionally, we need some control over the size of the pieces cut out from the graph, yet m-orthogonal separators do not provide such a guarantee on the random subsets. Thus, using m-orthogonal separators as a black box does not seem a possibility. Nevertheless, a simplified version of the mechanics of m-orthogonal separators, with the additional analysis of the dependencies generated by random projections, suffice to prove our result.

Remark. Our algorithm can be easily seen to solve also the ρ-separator problem, which was introduced in [9]. The goal in this problem is to partition the vertices of G into parts of size at most ρn, for some $0 < \rho < 1$ which is part of the input, so as to minimize the total weight of edges connecting different parts. For any desired $\varepsilon > 0$, the algorithm of [9] finds a $(1+\varepsilon)\rho$-separator of G whose value is at most $O(\frac{1+\varepsilon}{\varepsilon}\log n)$ times the optimal ρ-separator. Our algorithm similarly obtains a bi-criteria approximation factor of $O(\frac{1+\varepsilon}{\varepsilon}\sqrt{\log(1/\varepsilon\rho)\log n})$, which matches or improves over [9] for all relevant values of ε and ρ.

1.3 Related Work

ℓ_2^2 metrics play a key role in a sequence of approximation algorithms, starting with the minimum bisection algorithm of [3]. The main technical contribution of [3] is an ingenious analysis of the projection of respective vectors on a random line, and subsequent work refined and built upon that technical tool. At an abstract level, our work may be viewed as continuing this direction of examining the power of ℓ_2^2 metrics and developing appropriate algorithmic tools.

Spreading metrics were introduced by Even, Naor, Rao, and Schieber [10] in the context of divide-and-conquer approximation algorithms. They were subsequently used for several problems, including k-balanced partitioning [9], but they were computed via an LP relaxation. Spreading metrics were also used in conjunction with semi-definite programming and ℓ_2^2 metrics for minimum linear arrangement [4, 13], although the use of the spreading constraint therein was very different from ours.

2 Preliminaries

2.1 SDP Relaxation

We formulate the problem as a semidefinite program. Each vertex u is associated with a vector in $\mathbb{R}^d$, denoted also by u (the dimension d is finite but otherwise unrestricted), and the program finds an ℓ_2^2 semi-metric on V, as follows.

$$\min \sum_{(u,v)\in E} w(u,v)\cdot \tfrac{1}{2}||u-v||_2^2 \qquad \text{(SDP-SM)}$$

s.t.

$$||u-v||_2^2 + ||v-w||_2^2 \geq ||u-w||_2^2 \quad \forall u,v,w \in V \qquad (2.1)$$

$$\sum_{v\in S} \tfrac{1}{2}||u-v||_2^2 \geq |S| - \tfrac{n}{k} \qquad \forall S \subseteq V, u \in S \quad (2.2)$$

Constraints of type (2.1) are the triangle inequality for ℓ_2^2. They imply that $\big(V, ||u-v||_2^2\big)$ is a semi-metric. Constraints of type (2.2) are the *spreading* constraints. Such constraints forbid too many vertices to be close to each other. The following lemmas prove that (SDP-SM) is indeed a relaxation solvable in polynomial time.

LEMMA 2.1. *The value of an optimal solution to (SDP-SM) is a lower bound on the cost of an optimal solution to the k-partitioning problem.*

Proof. Let $C_1^*, C_2^*, \dots C_k^*$ be an optimal partitioning of G. For each vertex $u \in V$, let the corresponding vector $u \in R^k$ be all 0's except for one coordinate j, where C_j is the component to which u belongs, and that coordinate is set to 1. It can be seen that $\frac{1}{2}||u-v||_2^2 = 0$ if u and v belong to the same component, and that $\frac{1}{2}||u-v||_2^2 = 1$ if u and v belong to different components. Thus, it can be verified that constraints of types (2.1) and (2.2) are not violated. It can be easily seen that the value of the objective function of the relaxation for the given solution, equals the value of the optimal partitioning $C_1^*, C_2^*, \dots C_k^*$. ■

LEMMA 2.2. *An optimal solution to (SDP-SM) is computable in polynomial time.*

Proof. There is a polynomial number of constraints of type (2.1). We focus on constraints of type (2.2) (the spreading constraints). Fix a vertex u and a cardinality m of subset S. Arrange all vertices v according to increasing distance from u, with respect to the ℓ_2^2 semimetric of the relaxation. The subset S defined by the first m vertices in this ordering minimizes $\sum_{v \in S} \frac{1}{2}||u-v||_2^2$ (over all subsets of cardinality m). Therefore, for every size m and vertex u, we only need to check that the subset S above satisfies the constraint. This gives a separation oracle, which finishes the proof. ■

2.2 Spreading Constraints We now show that every feasible solution to (SDP-SM) has the property that not too many vertices are close to each other.

LEMMA 2.3. *Assume that $V \subset R^d$ satisfies constraints of type (2.2). Given $\epsilon > 0$, a subset $S \subseteq V$ and a vertex $u \in S$, if $\frac{1}{2}||u-v||_2^2 \leq \frac{\epsilon}{1+\epsilon}$ for every $v \in S$, then $|S| \leq (1+\epsilon)\frac{n}{k}$.*

Proof. Given $\epsilon > 0$, $S \subseteq V$ and $u \in S$, the spreading constraints (2.2) imply that there exists a vertex $v \in S$ such that $\frac{1}{2}||u-v||_2^2 \geq 1 - \frac{n}{k|S|}$. Assume to the contrary that $|S| > (1+\epsilon)\frac{n}{k}$. Thus, we get that $\frac{1}{2}||u-v||_2^2 > \frac{\epsilon}{1+\epsilon}$, which is a contradiction ■

2.3 Transforming ℓ_2^2 into ℓ_2 Our algorithm uses a transformation that changes ℓ_2^2 distances to ℓ_2 (Euclidean) distances, while incurring a loss of $O(\sqrt{\log n})$ in the distances. This single scale embedding of ℓ_2^2 metrics into ℓ_2 comes from [3] and the followup works of [5, 20]. The transformation is stated in the next lemma from [7].

LEMMA 2.4. *Assume that $V \subset R^d$, $|V| = n$, satisfies constraints of type (2.1). There are absolute constants $\delta > 0$ and $A > 0$, such that for every $\Delta > 0$, there exists a transformation $g_\Delta : V \to R^n$ (which can be computed efficiently) with the following properties for all $u, v \in V$:*

1. *$||g_\Delta(u) - g_\Delta(v)||_2 \leq \frac{A\sqrt{\log n}}{\Delta}||u-v||_2^2$.*
2. *$||g_\Delta(u) - g_\Delta(v)||_2 \geq \delta$ whenever $||u-v||_2^2 \geq \Delta$.*
3. *$||g_\Delta(u)||_2 = 1$.*

Notice that the transformation g_Δ does not preserve the spreading constraints completely, since distances might decrease. However, *large* ℓ_2^2 distances, specifically ℓ_2^2 distances of at least Δ, are guaranteed to be at least some absolute constant δ in ℓ_2. The analysis of the algorithm shows that this property is all that is needed, since we are only interested in the number of vertices that can be close to any specific vertex.

2.4 Thresholds of Correlated Gaussians Our analysis of the approximation algorithm uses two events involving correlated Gaussians. Set α_m such that $\Pr[X \geq \alpha_m] = \frac{1}{m}$, for a normally distributed random variable $X \sim N(0,1)$. Denote $\Phi(t) = \int_{x=-\infty}^{t} \varphi(x)dx$, where $\varphi(x) = (2\pi)^{-1/2} e^{-x^2/2}$ (these are the cumulative density function and probability density function, respectively, of the standard normal distribution). We wish to estimate the probability of the following two events. First, the event that two (correlated) standard normal variables both exceed the threshold α_m. Second, the event that two (correlated) standard normal variables are separated by the threshold α_m in the sense that one exceeds α_m and the other does not. The estimates are stated in the following lemma, whose part 1 is proved in [15], and part 2 proved in [7]).

LEMMA 2.5. *Let (X, Y) be drawn from a bivariate normal distribution where the expectation of each coordinate is 0, the standard deviation of each coordinate is 1, and the correlation of the two coordinates is ρ. Then:*

1. *$\Pr[X \geq \alpha_m, Y \geq \alpha_m] \leq \frac{2}{m} \cdot \left(1 - \Phi\left(\alpha_m \sqrt{\frac{1-\rho}{1+\rho}}\right)\right)$.*
2. *If $m > 3$, then $\Pr[X \geq \alpha_m, Y < \alpha_m] \leq \frac{B\sqrt{\log m}}{m} \cdot \sqrt{1-\rho}$, for some absolute constant B.*

3 The Rounding Procedure

3.1 Description of the Procedure We set $\varepsilon > 0$ and $\Delta = \frac{2\varepsilon}{1+\varepsilon}$. Let $\delta > 0$ be the absolute constant guaranteed in Lemma 2.4. Also let $C > 0$ be an absolute constant that will be determined later.

The rounding procedure, which we shall call k-**Partition**, iteratively chooses clusters and removes them from the graph, until the current remaining subgraph is small enough to be a cluster. More specifically, the procedure works as follows. First, it applies g_Δ to transform the metric from ℓ_2^2 to ℓ_2. Second, it chooses a random vector and assigns all vertices that are close to this vector to a new potential cluster. If the potential cluster is empty, or contains too many vertices, it is discarded and a new random cluster is chosen by the same method. This is repeated until a *good* cluster is found. It is then removed from the graph. A detailed description follows.

Procedure k-Partition.

```
$S \leftarrow V$.
While $|S| > (1+2\varepsilon)\frac{n}{k}$:
    Choose $r \in \mathbb{R}^n$ where $r_i \sim N(0,1)$ and i.i.d.
    Let $S_r \leftarrow \{u :\ u \in S, g_\Delta(u) \cdot r \geq \alpha_{Ck/\varepsilon}\}$.
    If $0 < |S_r| \leq (1+2\varepsilon)\frac{n}{k}$ then:
        Output $S_r$ as a cluster and let $S \leftarrow S \setminus S_r$.
```

3.2 Analysis of the Procedure The analysis is composed of two stages. The first stage consists shows that given that a vertex u belongs to S_r, hence with constant probability not more than $\varepsilon\frac{n}{k}$ vertices that are *far* from u also belong to S_r. Thus, with constant probability, given that u is in S_r, no more than $(1+2\varepsilon)\frac{n}{k}$ vertices belong to S_r. The second stage consists of bounding the total cost of the procedure over all iterations. Finally, in order to prove Theorem 1.1, we show that the procedure runs in expected polynomial time.

Define the following random variables:

- Y - the current subgraph.
- Y_r - the random subset of Y chosen by the procedure: $\{u :\ u \in Y, g_\Delta(u) \cdot r \geq \alpha_{Ck/\varepsilon}\}$.

A component Y_r is *good* if it is not empty and does not contain too many vertices. The following lemma shows that conditioning on $u \in Y_r$ (for any $u \in S$), then with constant probability Y_r is good. In order to prove the lemma we take into account the dependencies between u and all other vertices in S. These dependencies are related to the distances. Recall that the ℓ_2 distances do not satisfy the spreading constraints, but only a weakened version of them.

Lemma 3.1. *For every $S \subseteq V$, $u \in S$, and $\varepsilon > 0$,*

$$\Pr\left[|Y_r| \leq (1+2\varepsilon)\frac{n}{k} \;\middle|\; u \in Y_r, Y = S\right] \geq \frac{1}{2}.$$

Proof. Let I_v be an indicator for the event that $v \in Y_r$. Therefore, $|Y_r| = \sum_{v \in S} I_v$. The random variables $g_\Delta(u) \cdot r$ and $g_\Delta(v) \cdot r$, for any $u, v \in S$, are correlated normal variables. Specifically, using Property (3) of Lemma 2.4 and the choice of r, we get that $(g_\Delta(u) \cdot r, g_\Delta(v) \cdot r)$ is a bivariate normal random variables where the expectation of each coordinate is 0, the standard deviation of each coordinate is 1, and the correlation of the two coordinates is $g_\Delta(u) \cdot g_\Delta(v)$.

Let us choose a vertex $v \in S$ such that $||g_\Delta(u) - g_\Delta(v)||_2 \geq \delta$. This implies that $g_\Delta(u) \cdot g_\Delta(v) \leq 1 - \frac{1}{2}\delta^2$. By part (1) of Lemma 2.5, we get that:

$$\begin{aligned}
&\Pr\left[v \in Y_r \middle|\ u \in Y_r, Y = S\right] = \\
&= \Pr\left[g_\Delta(v) \cdot r \geq \alpha_{Ck/\varepsilon} \middle|\ g_\Delta(u) \cdot r \geq \alpha_{Ck/\varepsilon}, Y = S\right] \\
&\leq 2 \cdot \left(1 - \Phi\left(\alpha_{Ck/\varepsilon}\sqrt{\frac{1 - g_\Delta(u) \cdot g_\Delta(v)}{1 + g_\Delta(u) \cdot g_\Delta(v)}}\right)\right) \\
&\leq 2 \cdot \left(1 - \Phi\left(\alpha_{Ck/\varepsilon}\sqrt{\frac{\delta^2}{4 - \delta^2}}\right)\right),
\end{aligned}$$

where the last inequality is due to the fact that the function $1 - \Phi\left(\alpha\sqrt{\frac{1-x}{1+x}}\right)$ is an increasing function in x, for any $\alpha > 0$. By choosing a sufficiently large constant C (recall that by Lemma 2.4, δ is an absolute constant), we get that:

$$2 \cdot \left(1 - \Phi\left(\alpha_{Ck/\varepsilon}\sqrt{\frac{\delta^2}{4 - \delta^2}}\right)\right) \leq \frac{\varepsilon}{2k}.$$

Hence, we can conclude that:

$$\mathbf{E}\Big[\sum_{v \in S:\ \delta \leq ||g_\Delta(u) - g_\Delta(v)||_2} I_v \Big|\ u \in Y_r, Y = S\Big] \leq$$

$$\leq \frac{\varepsilon|S|}{2k} \leq \frac{\varepsilon n}{2k}.$$

By Markov's inequality, we get that:

$$\Pr\left[\sum_{v \in S:\ \delta \leq ||g_\Delta(u) - g_\Delta(v)||_2} I_v \leq \frac{\varepsilon n}{k} \;\middle|\; u \in Y_r, Y = S\right]$$

is at least $\frac{1}{2}$.

By Lemma 2.3, there are at most $(1+\varepsilon)\frac{n}{k}$ vertices at ℓ_2^2 distance of $\Delta = \frac{2\varepsilon}{1+\varepsilon}$ from u. By Property (2) of Lemma 2.4, there are at most $(1+\varepsilon)\frac{n}{k}$ vertices at ℓ_2 distance of δ from $g_\Delta(u)$. Thus,

$$\sum_{v \in S:\ \delta > ||g_\Delta(u) - g_\Delta(v)||_2} I_v \leq (1+\varepsilon)\frac{n}{k}.$$

Taking into account both cases, we can conclude that:

$$\Pr\left[\sum_{v\in S} I_v \le (1+2\varepsilon)\frac{n}{k} \;\Big|\; u\in Y_r, Y=S\right] \ge \frac{1}{2}.$$

■

Define the following events:

- $\mathcal{A}_{u,v}$ - u and v are separated by k-**Partition**.
- $\mathcal{B}_{u,v}$ - u and v are separated in the current iteration: $\{u \in Y_r, v \notin Y_r\}$ or $\{u \notin Y_r, v \in Y_r\}$.
- $\mathcal{C}_{u,v}$ - both u and v remain: $\{u, v \notin Y_r\}$.
- $\mathcal{D}$ - the cluster is *good*: $\{Y_r \neq \phi, |Y_r| \le (1+2\varepsilon)\frac{n}{k}\}$.

The following theorem proves the required approximation factor of k-**Partition**, as it bounds the probability, over *all* iterations, that u and v are in different components.

THEOREM 3.1. *For every subset $S \subseteq V$ where $u, v \in S$,*

$$\begin{aligned}&\Pr\left[\mathcal{A}_{u,v}\middle|Y=S\right] \le \\ &\le \quad \tfrac{(1+\varepsilon)AB}{\varepsilon}\sqrt{2\log\left(Ck/\varepsilon\right)\log n}\cdot||u-v||_2^2.\end{aligned}$$

Proof. The proof is by induction on $|S|$.
Base case: Given a subset $S \subseteq V$, where $|S| \le (1+2\varepsilon)\frac{n}{k}$ and $u, v \in S$, the desired probability is:

$$\Pr\left[\mathcal{A}_{u,v}\middle|Y=S\right] = 0.$$

This is true since the procedure stops when the remaining subgraph contains at most $(1+2\varepsilon)\frac{n}{k}$ vertices, hence u and v are not separated by k-**Partition**.
Inductive step: Let $S \subseteq V$, where $u, v \in S$, and assume correctness for all subsets of S that contain u and v. We use the following recursive formula:

$$\begin{aligned}&\Pr\left[\mathcal{A}_{u,v}\middle|Y=S\right] = \\ &= \quad \Pr\left[\mathcal{B}_{u,v}\middle|\mathcal{D}, Y=S\right] + \\ &\qquad + \Pr\left[\mathcal{C}_{u,v}\middle|\mathcal{D}, Y=S\right]\cdot \Pr\left[\mathcal{A}_{u,v}\middle|\mathcal{C}_{u,v}, \mathcal{D}, Y=S\right].\end{aligned}$$

Using Property (3) of Lemma 2.4 and the choice of r, we get that $(g_\Delta(u)\cdot r, g_\Delta(v)\cdot r)$ is a bivariate normal random variable, where the expectation of each coordinate is 0, the standard deviation of each coordinate is 1, and the correlation of the two coordinates is $g_\Delta(u)\cdot g_\Delta(v)$. Therefore, notice that:

$$\begin{aligned}\Pr\left[\mathcal{B}_{u,v}\middle|\mathcal{D}, Y=S\right] &\le \frac{\Pr\left[\mathcal{B}_{u,v}\middle|Y=S\right]}{\Pr\left[\mathcal{D}\middle|Y=S\right]} \le \\ &\le \quad \frac{2B\sqrt{\log\left(Ck/\varepsilon\right)}}{Ck/\varepsilon}\cdot\frac{\sqrt{1-g_\Delta(u)\cdot g_\Delta(v)}}{\Pr\left[\mathcal{D}\middle|Y=S\right]} \\ &\le \quad \frac{B\sqrt{2\log\left(Ck/\varepsilon\right)}}{Ck/\varepsilon}\cdot\frac{A\sqrt{\log n}}{\Delta}\cdot\frac{||u-v||_2^2}{\Pr\left[\mathcal{D}\middle|Y=S\right]}.\end{aligned}$$

The second inequality is by part (2) of Lemma 2.5. The last inequality is derived from Property (1) of Lemma 2.4 and the fact that $\sqrt{1-g_\Delta(u)\cdot g_\Delta(v)} = \frac{1}{\sqrt{2}}||g_\Delta(u)-g_\Delta(v)||_2$. Additionally,

$$\begin{aligned}&\Pr\left[\mathcal{C}_{u,v}\middle|\mathcal{D}, Y=S\right] \le \\ &\le \quad 1-\Pr\left[u\in Y_r\middle|\mathcal{D}, Y=S\right] \\ &= \quad 1-\frac{\Pr\left[\mathcal{D}\middle|u\in Y_r, Y=S\right]\cdot\Pr\left[u\in Y_r\middle|Y=S\right]}{\Pr\left[\mathcal{D}\middle|Y=S\right]} \\ &\le \quad 1-\frac{1}{2Ck/\varepsilon}\cdot\frac{1}{\Pr\left[\mathcal{D}\middle|Y=S\right]}.\end{aligned}$$

The first inequality is derived from the definition of the event $\mathcal{C}_{u,v}$. The last inequality is derived from Lemma 3.1 and the fact that: $\Pr\left[u\in Y_r\middle|Y=S\right] = \frac{1}{Ck/\varepsilon}$.

By the inductive hypothesis we get that:

$$\begin{aligned}&\Pr\left[\mathcal{A}_{u,v}\middle|\mathcal{C}_{u,v}, \mathcal{D}, Y=S\right] \le \\ &\le \quad \frac{(1+\varepsilon)AB}{\varepsilon}\sqrt{2\log\left(Ck/\varepsilon\right)\log n}\cdot||u-v||_2^2.\end{aligned}$$

Using the recursive formula and plugging in all of the above (recall that $\Delta = \frac{2\varepsilon}{1+\varepsilon}$) completes the proof. ■

Proof. [Proof of Theorem 1.1.] Procedure k-**Partition** outputs components of size at most $(1+2\epsilon)\frac{n}{k}$. The expected cost of separating these components, by Theorem 3.1 and Lemma 2.1, is at most

$$\frac{(1+\epsilon)AB\sqrt{2\log\left(Ck/\epsilon\right)\log n}}{\epsilon}$$

times the value of an optimal solution to (SDP-SM).

We bound the expected number of iterations of k-**Partition**. The number of random choices of r needed to find a single *good* cluster is a geometric random variable with parameter $\Pr\left[\mathcal{D}\middle|Y=S\right]$. Notice that for any subset $S \subseteq V$ and for any $u \in S$, $\Pr\left[u\in Y_r\middle|Y=S\right] = \frac{1}{Ck/\varepsilon}$. Additionally, by Lemma 3.1 $\Pr\left[\mathcal{D}\middle|u\in Y_r, Y=S\right] \ge \frac{1}{2}$. This implies that: $\Pr\left[\mathcal{D}\middle|Y=S\right] \ge \frac{1}{2Ck/\varepsilon}$. Thus, the expected number of choices of r needed to find a single good cluster is upper bounded by $2Ck/\varepsilon$. Therefore, the expected number of iterations made throughout the procedure is upper bounded by $2Cnk/\varepsilon$.

The theorem now follows by setting $\epsilon = \frac{1}{2}$. Notice that k-**Partition** might not output exactly k components. This can be easily resolved by standard methods, see [9] for further details. ■

4 Integrality Gap of $\Omega(\log k)$

We present an integrality gap example for (SDP-SM). Specifically, we show that for the d-dimensional hypercube the integrality gap is $\Omega(\log k)$. Denote by

$H(p) = -p\log(p) - (1-p)\log(1-p)$ the entropy function. We remark that the hypercube is also an $\Omega(\log n)$ integrality gap example for another cut problem, namely minimum multicut [25, 1].

THEOREM 4.1. *For every $d > 1$, $\alpha > 1$ and $k \leq 2^{d\left(1-H\left(\frac{1}{2\alpha^2}\right)\right)}$, the integrality gap of (SDP-SM) for the d-dimensional hypercube graph (on 2^d vertices) is at least $\frac{1}{2\alpha^2}\log k$.*

Proof. First, we lower bound the cost of any integral solution. By the edge isoperimetric inequality for the hypercube, for any subset $S \subseteq \{\pm 1\}^d$ (which is not too large), the number of edges that cross the cut defined by S is at least $|S|(d - \log|S|)$. The size of S is at most $\frac{2^d}{k}$, which implies that the number of edges in the cut defined by S is at least $|S|\log k$. Let $C_1^*, C_2^*, \ldots C_k^*$ be an optimal solution for the problem. Then its cost is at least:

$$\frac{1}{2}\left(|C_1^*| + |C_2^*| + \cdots + |C_k^*|\right)\log k = 2^{d-1}\log k.$$

Second, we present a solution to (SDP-SM) which costs $\alpha^2 2^d$. For any $u \in \{\pm 1\}^d$, define a vector $\bar{u} = \frac{\alpha}{\sqrt{d}}u$. We show that the vectors $\{\bar{u}\}_{u\in\{\pm 1\}^d}$ define a feasible solution. First, notice that

$$d_H(u,v) = \frac{d}{2\alpha^2}\left(\frac{1}{2}||\bar{u} - \bar{v}||_2^2\right),$$

where d_H is the Hamming distance. Hence, the vectors $\{\bar{u}\}_{u\in\{\pm 1\}^d}$ do not violate the triangle inequality, implying that constraints of type (2.1) are satisfied.

We focus our attention on constraints of type (2.2). Notice that we only need to prove that:

$$\frac{2\alpha^2}{d}\sum_{v\in S} d_H(u,v) \geq |S| - \frac{2^d}{k}, \tag{4.3}$$

for any subset $S \subseteq \{\pm 1\}^d$ and vertex $u \in S$. The subset S of size m which minimizes the LHS of (4.3) contains the m closest vertices to u with respect to the Hamming metric.

Assume to the contrary that (4.3) is incorrect for some subset S of size m. Without loss of generality, assume that S minimizes the LHS of (4.3). Choose a vertex v in S. If $d_H(u,v) < \frac{d}{2\alpha^2}$, then v's contribution to the LHS of (4.3) is *strictly* smaller than 1. If $d_H(u,v) = \frac{d}{2\alpha^2}$, then v's contribution to the LHS of (4.3) is 1. If $d_H(u,v) > \frac{d}{2\alpha^2}$, then v's contribution to the LHS of (4.3) is *strictly* greater than 1. However, in all cases, each $v \in S$ contributes an additional 1 to the RHS of (4.3).

Thus, without loss of generality assume that m is such that the maximal Hamming distance from u in S is $\frac{d}{2\alpha^2}$. Arbitrarily choose $m = \sum_{j=0}^{\frac{d}{2\alpha^2}} \frac{d!}{j!(d-j)!}$. Therefore, according to the assumption,

$$\frac{2\alpha^2}{d}\sum_{v\in S} d_H(u,v) < \sum_{j=0}^{d/2\alpha^2} \frac{d!}{j!(d-j)!} - \frac{2^d}{k}.$$

However, the LHS is nonnegative, therefore we get that $k > 2^d / \sum_{j=0}^{d/2\alpha^2} \frac{d!}{j!(d-j)!}$. The bound $\sum_{j=0}^{d/2\alpha^2} \frac{d!}{j!(d-j)!} \leq 2^{dH\left(\frac{1}{2\alpha^2}\right)}$ implies that $k > 2^{d\left(1-H\left(\frac{1}{2\alpha^2}\right)\right)}$, which is a contradiction. Hence, constraints of type (2.2) are satisfied and $\{\bar{u}\}_{u\in\{\pm 1\}^d}$ is a feasible solution.

We calculate the cost of the given solution to (SDP-SM) as follows:

$$\sum_{u,v|d_H(u,v)=1}\left(\frac{1}{2}||\bar{u} - \bar{v}||_2^2\right) = 2^d \cdot \frac{2\alpha^2}{d} \cdot d \cdot \frac{1}{2} = \alpha^2 2^d.$$

This gives the desired integrality gap of $\frac{1}{2\alpha^2}\log k$. ∎

Proof. [Proof of Theorem 1.2.] Let $c = \Omega(1)$ be an absolute constant where $k \leq n^{1-c}$. Set α such that the following equality holds: $H\left(\frac{1}{2\alpha^2}\right) = c$. Since $0 < c < 1$ there is a (unique) solution in which $\alpha > 1$. By Theorem 4.1 the integrality gap of (SDP-SM) is at least $\frac{1}{2\alpha^2}\log k$. Additionally, by the definition of α, $\alpha = O(1)$. Hence, we obtain an integrality gap of $\Omega(\log k)$. ∎

Acknowledgments We wish to thank Yuval Filmus for helpful discussions.

References

[1] A. Agarwal, M. Charikar, K. Makarychev, and Y. Makarychev. $O(\sqrt{\log n})$ approximation algorithms for min uncut, min 2cnf deletion, and directed cut problems. In *37th Annual ACM Symposium on Theory of computing*, pages 573–581, 2005.

[2] K. Andreev and H. Räcke. Balanced graph partitioning. *Theory Comput. Syst.*, 39(6):929–939, 2006.

[3] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings, and graph partitionings. In *36th Annual Symposium on the Theory of Computing*, pages 222–231, May 2004.

[4] M. Charikar, M. T. Hajiaghayi, H. Karloff, and S. Rao. l22 spreading metrics for vertex ordering problems. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1018–1027, 2006.

[5] S. Chawla, A. Gupta, and H. Räcke. Improved approximations to sparsest cut. In *16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 102–111, 2005.

[6] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. In *20th Annual IEEE Conference on Computational Complexity*, pages 144–153, June 2005.

[7] E. Chlamtac, K. Makarychev, and Y. Makarychev. How to play unique games using embeddings. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 687–696, 2006.

[8] N. R. Devanur, S. A. Khot, R. Saket, and N. K. Vishnoi. Integrality gaps for sparsest cut and minimum linear arrangement problems. In *38th Annual ACM Symposium on Theory of Computing*, pages 537–546. ACM Press, 2006.

[9] G. Even, J. Naor, S. Rao, and B. Schieber. Fast approximate graph partitioning algorithms. *SIAM J. Comput.*, 28(6):2187–2214, 1999.

[10] G. Even, J. Naor, S. Rao, and B. Schieber. Divide-and-conquer approximation algorithms via spreading metrics. *J. ACM*, 47(4):585–616, 2000.

[11] U. Feige. Relations between average case complexity and approximation complexity. In *34th Annual ACM Symposium on the Theory of Computing*, pages 534–543, July 2002.

[12] U. Feige and R. Krauthgamer. A polylogarithmic approximation of the minimum bisection. *SIAM Review*, 48(1):99–130, 2006.

[13] U. Feige and J. R. Lee. An improved approximation ratio for the minimum linear arrangement problem. *Inf. Process. Lett.*, 101(1):26–29, 2007.

[14] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoret. Comput. Sci.*, 1(3):237–267, 1976.

[15] L. Holst and S. Janson. Poisson approximation using the stein-chen method and coupling: number of exceedances of gaussian random variables. *Ann. Probab.*, 18:713–723, 1990.

[16] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 1998.

[17] S. Khot. On the power of unique 2-prover 1-round games. In *34th Annual ACM Symposium on the Theory of Computing*, pages 767–775, July 2002.

[18] S. Khot and N. K. Vishnoi. The unique games conjecture, integrality gap for cut problems and the embeddability of negative type metrics into ℓ_1. In *46th IEEE Annual Symposium on Foundations of Computer Science"*, pages 53–62, 2005.

[19] R. Krauthgamer and Y. Rabani. Improved lower bounds for embeddings into l1. In *17th annual ACM-SIAM symposium on Discrete algorithm*, pages 1010–1017. ACM, 2006.

[20] J. R. Lee. On distance scales, embeddings, and efficient relaxations of the cut cone. In *16th annual ACM-SIAM symposium on Discrete algorithms*, pages 92–101. SIAM, 2005.

[21] F. Leighton, F. Makedon, and S. Tragoudas. Approximation algorithms for VLSI partition problems. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 2865–2868. IEEE Computer Society Press, 1990.

[22] 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.

[23] H. Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of 40th Annual ACM Symposium on Theory of Computing*, 2008. To appear.

[24] H. D. Simon and S. Teng. How good is recursive bisection? *SIAM J. Sci. Comput.*, 18(5):1436–1445, 1997.

[25] B. Yu, J. Cheriyan, and P. E. Haxell. Hypercubes and multicommodity flows. *SIAM J. Discret. Math.*, 10(2):190–200, 1997.