Approximate Hierarchical Clustering via Sparsest Cut and Spreading Metrics

Moses Charikar* moses@cs.stanford.edu

Vaggos Chatziafratis* vaggos@stanford.edu

Abstract

Dasgupta recently introduced a cost function for the hierarchical clustering of a set of points given pairwise similarities between them. He showed that this function is NP-hard to optimize, but a top-down recursive partitioning heuristic based on an α_n-approximation algorithm for uniform sparsest cut gives an approximation of $O(\alpha_n \log n)$ (the current best algorithm has $\alpha_n = O(\sqrt{\log n})$). We show that the aforementioned sparsest cut heuristic in fact obtains an $O(\alpha_n)$-approximation. The algorithm also applies to a generalized cost function studied by Dasgupta. Moreover, we obtain a strong inapproximability result, showing that the Hierarchical Clustering objective is hard to approximate to within any constant factor assuming the *Small-Set Expansion (SSE) Hypothesis*. Finally, we discuss approximation algorithms based on convex relaxations. We present a spreading metric SDP relaxation for the problem and show that it has integrality gap at most $O(\sqrt{\log n})$. The advantage of the SDP relative to the sparsest cut heuristic is that it provides an explicit lower bound on the optimal solution and could potentially yield an even better approximation for hierarchical clustering. In fact our analysis of this SDP served as the inspiration for our improved analysis of the sparsest cut heuristic. We also show that a spreading metric LP relaxation gives an $O(\log n)$-approximation.

1 Introduction

Hierarchical Clustering (HC) of a data set is a recursive partitioning of the data into clusters. Such methods are widely used in data analysis. To be more formal, in the Hierarchical Clustering (HC) problem, the input is a weighted undirected graph $G = (V, E, w)$. Each data point corresponds to a node in the graph and edges connect similar points. The heavier the edge weight the stronger the similarity between the data points. The goal is to produce a partitioning of the data into successively smaller clusters, starting from the original graph G as the initial cluster and ending with n singleton clusters. The HC is represented as a tree with leaves corresponding to data points and internal nodes corresponding to clusters in the hierarchy.

Such a hierarchical decomposition of data has several advantages over *flat clustering* (k-means, k-center etc): firstly, there is no need to fix the number k of clusters we want to create; secondly, large datasets are understood simultaneously at many levels of granularity and thirdly, many greedy heuristics with provable approximation guarantees can be used to construct it.

Despite its important applications for many scientific areas such as biology (e.g. gene expression), data analysis, phylogenetics, social sciences and statistics, HC and the algorithms we use to solve it in practice are not yet well-understood. Many heuristics have been proposed, some of which are based on a natural "bottom-up" approach by recursively merging data that are similar: at the beginning each data point is a separate cluster and we start merging them based on their similarity as we go up the hierarchy. These are the so-called *agglomerative* methods that are provided by standard data analysis packages and include for example single-linkage, average linkage etc. ([20, 21, 4, 19]). Another way for dealing with HC is to follow a "top down" approach. These are the so called *divisive* methods (including k-means etc.), where all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy. One disadvantage about these methods is that they are specified procedurally rather than in terms of an objective function for HC and that it is not clear what they are optimizing for.

This lack of a concrete objective function for hierarchical clustering (based on the similarities between points) was addressed by the recent work of Dasgupta ([10]). He introduced a simple cost function that, given pairwise similarities between data points, assigns a score to any possible tree on those points. The tree corresponds to the hierarchical decomposition of the data and its score reflects the quality of the solution.

Let T be any rooted (not necessarily binary) tree that has a leaf for each point in our dataset. For a node u in T, we denote with $T[u]$ the subtree rooted at u,

*Computer Science Department, Stanford University.

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited

and with $\mathbf{leaves}(T[u]) \subseteq V$ we denote the leaves of this subtree. For leaves $i, j \in V$, the expression $i \vee j$ denotes their lowest common ancestor in T, i.e. $T[i \vee j]$ is the smallest subtree whose leaves include both i and j. The following cost function is the HC cost function:

$$cost_G(T) = \sum_{ij \in E} w_{ij} |\mathbf{leaves}(T[i \vee j])|. \tag{1.1}$$

We observe that a heavy edge should not be cut at the top of the tree because it would cause a high cost due to the term $|\mathbf{leaves}(T[i \vee j])|$ that would be large. For example, if an edge $\{i, j\}$ of unit weight is cut at the first split of the data, then we pay n. If it is cut further down, in a subtree that contains a δ fraction of the data, then we pay δn (see Figure 1).

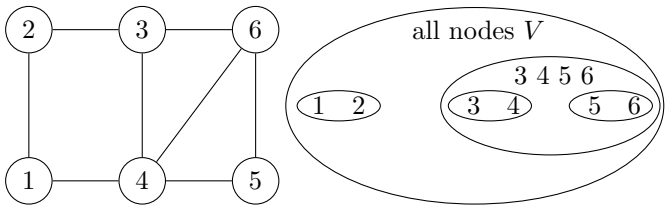

Figure 1: Here is a simple unweighted graph G and a candidate hierarchical clustering of G. It is easy to compute the cost of the implied tree decomposition: we pay 2×6 for the edges (2,3), (1,4), 3×4 for the edges (3,6), (4,6), (4,5), 2 for (1,2), 2 for (3,4) and 2 for (5,6). The total cost is 30.

We would like to find a tree T^* that minimizes the above cost. It is not difficult to see that there must always exist an optimal tree that is binary, since by converting any split that creates more than two subtrees to a sequence of binary splits, we can never increase the cost. A generalized version for the cost function is also considered in [10]:

$$cost_G(T) = \sum_{ij \in E} w_{ij} f(|\mathbf{leaves}(T[i \vee j])|). \tag{1.2}$$

1.1 Related Work. Dasgupta introduced the cost function (1.1) and explained why it is a good objective function for hierarchical clustering. He presented some interesting special cases (e.g. planted partitions) for which optimizing (1.1) actually finds the correct underlying HC. He showed that optimizing it is an NP-hard problem and showed that a simple heuristic based on an α_n approximation for Sparsest Cut will achieve a factor $O(\alpha_n \cdot \log n)$ approximation. The current best α_n ratio for Sparsest Cut is $O(\sqrt{\log n})$ from a breakthrough result of [2]. The heuristic starts by taking Sparsest Cut for the input graph G, splitting it into (G_1, G_2) and then applying Sparsest Cut recursively to the pieces G_1, G_2. Dasgupta also proved that a slightly modified heuristic yields basically the same approximation guarantee for optimizing (1.2).

Another natural approach for dealing with HC is to try to optimize standard popular cost functions for *flat clustering*, such as the k-means, k-median or k-center([11, 30, 27]) and synthesize solutions for different values of k so as to obtain a hierarchical clustering. However, it is not clear how to leverage guarantees for these distance based objective functions so as to obtain a guarantee for the similarity based objective for HC.

Methods of HC have also been studied in terms of statistical consistency ([18, 9, 13]). Data points are sampled from a fixed underlying distribution and we are interested in the convergence of the tree structure obtained from the data as the sample size goes to infinity. Only a few methods ([9, 13]) are known to be consistent.

Furthermore, the authors of [4] study the performance of agglomerative clustering techniques in the presence of noise and they propose a new algorithm that is more robust and performs better in cases with noisy data where traditional agglomerative algorithms fail.

Recently we were informed that independently of our work, Roy and Pokutta [33] got a similar result for Hierarchical Clustering via spreading metrics. In particular, they used an LP relaxation based on ultrametrics to obtain an $O(\log n)$ approximation. The LP relaxation they formulated was similar to ours but we also considered an SDP relaxation and managed to obtain an $O(\sqrt{\log n})$ approximation. As far as their analysis is concerned they used the extensively studied (in the context of graph partitioning) idea of sphere growing (see [25, 17, 14, 7, 15]). In a similar spirit, we got our initial $O(\log n)$ approximation by proving that the hierarchical clustering objective function falls into the *divide and conquer approximation algorithms via spreading metrics* paradigm of [15] and combining it with a result of Bartal ([5]). Finally, they also gave the same constant factor inapproximability result as we did, based on the small set expansion hypothesis.

1.2 Our results and structure of the paper. Some preliminaries are presented in Section 2. We show (Section 3) that the recursive sparsest cut (RSC) algorithm that uses any α_n-approximation algorithm for uniform sparsest cut achieves an $O(\alpha_n)$ approximation for hierarchical clustering, shaving a $\log n$ factor from Dasgupta's analysis. A similar approximation guarantee with better running time is provided when using recursive Balanced Cut instead of Sparsest Cut. The analysis can be modified to prove that the same guar-

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited

antee holds even for the generalized cost function (1.2). We also present (Section 4) a strong inapproximability result for HC, in particular, that it is hard to approximate HC to within any constant factor assuming the Small Set Expansion (SSE) Hypothesis. In Section 5, we present an SDP relaxation based on spreading metrics with integrality gap at most $O(\sqrt{\log n})$ for HC. The advantage of the SDP relative to the sparsest cut heuristic is that it provides an explicit lower bound on the optimal solution and could potentially yield an even better approximation for hierarchical clustering. In fact, we first developed a rounding algorithm for this SDP and our analysis later served as the inspiration for our improved analysis of the sparsest cut heuristic for both cost functions (1.1) and (1.2). Finally, we show how the spreading metrics paradigm of [15] in combination with a result of Bartal [5] (Section 6) can be exploited in order to get an $O(\log n)$ approximation for hierarchical clustering via a linear program (6.2). We conclude in Section 7 with questions for further research.

A key idea behind our analysis of the Recursive Sparsest Cut algorithm as well as the formulation of the SDP relaxation is to view a hierarchical clustering of n data points as a collection of partitions of the data, one for each level $t = n-1, \ldots, 0$. Here the partition for a particular level t consists of maximal clusters in the hierarchical clustering of size at most t (for convenience, we define level $t = 0$ to be the same as level $t = 1$, where every point is a separate cluster by itself). When we partition a cluster of size r, we charge this cost in the Recursive Sparsest Cut analysis to levels $t \in [r/4, r/2]$ of the aforementioned collection of partitions (and for the SDP analysis we charge it to levels $t \in [r/8, r/4]$). This is crucial for eliminating the $\log n$ term in the approximation guarantee.

2 Preliminaries

Here, we would like to briefly discuss some important problems and definitions that will frequently come up in the rest of the paper. Some additional definitions and facts may be presented in the sections for which they are relevant.

Sparsest Cut. Given a weighted, undirected graph $G = (V, E, w)$ ($|V| = n$) we want to find a set $S \neq \emptyset, V$ that minimizes the ratio: $\frac{w(S, V\setminus S)}{|S|\cdot|V\setminus S|}$. It is an NP-hard problem for which many important results are known including the LP relaxation of Leighton-Rao [26] with approximation ratio $O(\log n)$ and the SDP relaxation with triangle inequality of Arora, Rao, Vazirani [2] with approximation ratio $O(\sqrt{\log n})$; it is a major open question if we can improve this approximation ratio.

c-Balanced Cut. This is another variation of balanced partitioning where we want to have some kind of guarantee on the size of the smallest part produced by the cut. Formally, we are given a weighted undirected graph G on n vertices and the goal is to partition the vertices into 2 pieces $(S, \bar{S})$ with sizes $cn \leq |S|, |\bar{S}| \leq (1-c)n$ and of course with the total weight of the edges connecting the 2 components being small. It is known ([2]) how to get an $O(\sqrt{\log n})$ pseudo-approximation algorithm for this problem, in the sense that the algorithm will produce a c'-Balanced Cut for some constant $c' < c$ and such that the cost of the solution is at most $O(\sqrt{\log n})$ times the optimum c-Balanced Cut. Here c' and c are constants in $(0, 1/2]$. We note that the above two problems (Sparsest and Balanced Cut) can be approximated up to a $O(\sqrt{\log n/\epsilon})$ factor in time $\tilde{O}(m + n^{1+\epsilon})$ by combining [34] with [35] or [22] (see also [29]).

k-Balanced Partitioning. Given a weighted undirected graph G on n vertices, the goal is to partition the vertices into k equally sized components of size roughly n/k so that the total weight of the edges connecting different components is small. It is an important generalization of well-known graph partitioning problems, including minimum bisection (k=2) and minimum balanced cut and it has applications in VLSI design, data mining (clustering), social network analysis etc. It is an NP-hard problem and the authors of [24] present a bi-criteria (which means that pieces may have size $2n/k$ rather than n/k) approximation algorithm achieving an approximation of $O(\sqrt{\log n \log k})$. Their result will be useful in our analysis for our spreading metrics SDP in Section 5. However, for us the dependence on k will be unimportant since in our analysis we only need k to be a small constant (e.g. k=4).

Small-Set Expansion. SSE is a hardness assumption that informally tells us the following: Given a graph G, it should be hard to distinguish between the case where there exists a small set S that has only a few edges leaving it versus the case where for all small sets S there are many edges leaving the sets. For a formal statement see Section 4. This hardness assumption is closely connected to the Unique Games Conjecture (UGC) of [23] and its variants. In particular, the SSE Hypothesis implies UGC([31]) and it has been used to prove many inapproximability results for problems like balanced separator and minimum linear arrangement ([32]).

Minimum Linear Arrangement. Given a weighted undirected multigraph $G(V, E, w)$ ($|V| = n$) we want to find a permutation $\pi : V \to \{1, 2, \ldots, |V|\}$ that minimizes: $\sum_{(x,y)\in E, x<y} w(x, y) \cdot |\sigma(y) - \sigma(x)|$. A factor $O(\sqrt{\log n} \log \log n)$ approximation for MLA was shown in [8, 16]. In addition, some recent hardness results are also known: in [32] it is shown that it is

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited

SSE-hard to approximate MLA to within any fixed constant factor and in [1] the authors prove that MLA has no polynomial time approximation scheme, unless NP-complete problems can be solved in randomized subexponential time.

GAUSSIAN GRAPHS. For a constant $\rho \in (-1, 1)$, let $\mathcal{G}(\rho)$ denote the infinite graph over $\mathbb{R}$ where the weight of an edge (x, y) is the probability that two standard Gaussian random variables X, Y with correlation ρ equal x and y respectively. The expansion profile of Gaussian graphs is given by $\Phi_{\mathcal{G}(\rho)}(\mu) = 1 - \Gamma_\rho(\mu)/\mu$ where the quantity $\Gamma_\rho(\mu)$ defined as

$$\Gamma_\rho(\mu) \triangleq \mathbb{P}_{(x,y)\sim\mathcal{G}_\rho}(x \geq t, y \geq t),$$

where $\mathcal{G}_\rho$ is the 2-dimensional Gaussian distribution with covariance matrix:

$$\begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

and $t \geq 0$ is such that $\mathbb{P}_{(x,y)\sim\mathcal{G}_\rho}\{x \geq t\} = \mu$.

3 Better Analysis for Recursive Sparsest Cut

As discussed previously, Dasgupta [10] showed that a simple top-down Recursive Sparsest Cut (RSC) heuristic that uses an α_n-approximation algorithm for uniform sparsest cut gives an approximation of $O(\alpha_n \log n)$ for hierarchical clustering. More precisely, the RSC heuristic starts from the given graph $G = (V, E)$, uses any α_n-approximation algorithm for sparsest cut, thus splitting G into (G_1, G_2) and then recurses on G_1 and G_2. The output is a binary tree of the sequence of cuts performed by the algorithm.

In this section, by drawing inspiration from our SDP construction and analysis presented later in Section 5, we present an improved analysis for this simple heuristic, dropping the $\log n$ factor and showing that it actually yields an $O(\alpha_n)$ approximation. This is satisfying since any improvement for Sparsest Cut would immediately yield a better approximation result for hierarchical clustering.

3.1 Analysis of RSC heuristic. Let the given graph be $G = (V, E)$. We suppose for clarity of presentation that it is unweighted; the analysis applies directly to weighted graphs and later, we see how to generalize it for more general cost functions. Let OPT be the optimal solution for hierarchical clustering (we abuse notation slightly by using OPT to denote both the solution as well as its objective function value). Let $OPT(t)$ be the maximal clusters in OPT of size at most t. Note that $OPT(t)$ is a partition of V.

We denote $E_{OPT}(t)$ the edges that are cut in $OPT(t)$, i.e. edges with end points in different clusters in $OPT(t)$. For convenience, we also define $E_{OPT}(0) \triangleq E$ (see Figure 2).

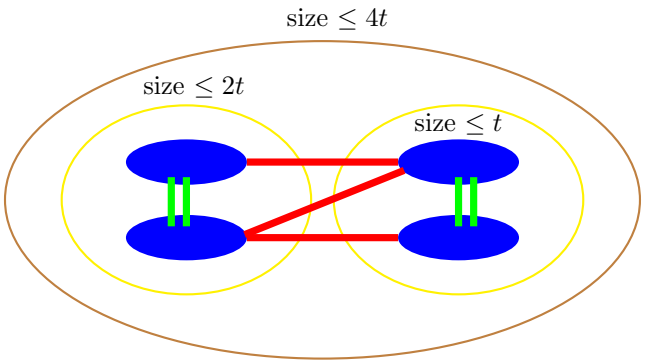

Figure 2: Illustration for the sets $E_{OPT}(t)$, $E_{OPT}(2t)$, $E_{OPT}(4t)$. Looking at clusters that are maximal and of size at most t (filled in blue), $2t$ (yellow border), $4t$ (brown border), we have that the red edges belong to $E_{OPT}(2t)$, whereas in $E_{OPT}(t)$, we have the red and the green edges. Any edge that goes out of the big cluster of size $\leq 4t$ (brown) contributes to all terms $E_{OPT}(t), E_{OPT}(2t), E_{OPT}(4t)$.

Claim 3.1. $OPT = \sum_{t=0}^{n-1} |E_{OPT}(t)|$.

Proof. Consider any edge $(u, v) \in E$. Suppose that the size of the minimal cluster in OPT that contains both u and v is r. Then the contribution of (u, v) to the LHS is r. On the other hand, $(u, v) \in E_{OPT}(t)$ for all $t \in \{0, \ldots, r-1\}$. Hence the contribution to the RHS is also r. □

It will be convenient to use the following bound that is directly implied by the above claim:

$$2 \cdot OPT = 2 \cdot \sum_{t=0}^{n-1} |E_{OPT}(t)| \geq \sum_{t=0}^{n} E_{OPT}(\lfloor t/2 \rfloor) \tag{3.3}$$

Let's look at a cluster A with size $|A| = r$ in the solution produced by RSC. Using a sparsest cut approximation algorithm, we create two clusters B_1, B_2 with sizes $s, (r-s)$ respectively, with B_1 being the smaller, i.e. $s \leq \lfloor r/2 \rfloor$. The contribution of this cut to the hierarchical clustering objective function is: $|E(B_1, B_2)| \cdot r$. We basically want to charge this cost to $OPT(\lfloor r/2 \rfloor)$ and for that we first observe that the edges cut in $OPT(\lfloor r/2 \rfloor)$, when restricted to the cluster A (i.e. having both endpoints in A), satisfy the following:

$$s \cdot |E_{OPT}(\lfloor r/2 \rfloor) \cap A| \leq \sum_{t=r-s+1}^{r} |E_{OPT}(\lfloor t/2 \rfloor) \cap A| \tag{3.4}$$

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited

This follows easily from the fact that $|E_{OPT}(t) \cap A| \leq |E_{OPT}(t-1) \cap A|$ (by definition, the smaller the value of t, the more edges are cut). Now in order to explain our charging scheme, let's look at the partition $A_1, ..., A_k$ induced inside the cluster A by $OPT(\lfloor r/2 \rfloor) \cap A$, where by design the size of each $|A_i| = \gamma_i |A|$, $\gamma_i \leq 1/2$. We have:

$$\frac{|E(A_i, A \setminus A_i)|}{|A_i||A \setminus A_i|} = \frac{|E(A_i, A \setminus A_i)|}{\gamma_i(1-\gamma_i)r^2}, \forall i \in \{1, \ldots, k\}$$

We take the minimum over all i (an upper bound on the sparsest cut in A) and we have:

$$\min_i \frac{|E(A_i, A \setminus A_i)|}{\gamma_i(1-\gamma_i)r^2} \leq \frac{\sum_i |E(A_i, A \setminus A_i)|}{\sum_i \gamma_i(1-\gamma_i)r^2} \leq$$
$$\leq 2 \cdot \frac{|E_{OPT}(\lfloor r/2 \rfloor) \cap A|}{r^2/2} = 4 \cdot \frac{|E_{OPT}(\lfloor r/2 \rfloor) \cap A|}{r^2}$$

The first inequality above, trivially follows by definition for the minimum and the second inequality holds because $\sum_{i=1}^k \gamma_i = 1, \sum_{i=1}^k \gamma_i^2 \leq 1/2$ (note that all $\gamma_i \leq 1/2$) and the factor of 2 is introduced since we double counted every edge. We partition A using an α_r-approximation for sparsest cut and we get (since $\alpha_r \leq \alpha_n$):

$$\frac{|E(B_1, B_2)|}{s(r-s)} \leq \alpha_n \cdot \frac{4}{r^2} \cdot |E_{OPT}(\lfloor r/2 \rfloor) \cap A|$$

since the RHS (without the α_n factor) is an upper bound of the optimal sparsest cut value. The contribution of this step to the hierarchical clustering objective function is:

$$r|E(B_1, B_2)| \leq \frac{4\alpha_n s(r-s)}{r} \cdot |E_{OPT}(\lfloor r/2 \rfloor) \cap A| \leq$$
$$\leq 4\alpha_n s \cdot |E_{OPT}(\lfloor r/2 \rfloor) \cap A| \tag{3.5}$$

We claim the following:

Claim 3.2. *Let A be a cluster of size r_A in our hierarchical clustering solution, that we split into 2 pieces (B_1, B_2) of sizes $s_A, r_A - s_A$ respectively with $|B_1| \leq |B_2|$ (so s_A stands for the size of the small piece B_1 after we split A). Then, summing over all clusters A we get:*

$$\sum_A \sum_{t=r_A - s_A + 1}^{r_A} |E_{OPT}(\lfloor t/2 \rfloor) \cap A| \leq \sum_{t=0}^{n} |E_{OPT}(\lfloor t/2 \rfloor)|$$

Proof. For a fixed value of t and A, the LHS is: $|E_{OPT}(\lfloor t/2 \rfloor) \cap A|$. Consider which clusters A contribute such a term to the LHS. From the fact that $r_A - s_A + 1 \leq t \leq r_A$, we need to have that $|B_2| < t$ and since B_2 is the larger piece that was created when A was split, we deduce that A is a **minimal** cluster of size $|A| \geq t > |B_2| \geq |B_1|$, i.e. if both A's children are of size less than t, then this cluster A contributes such a term. The set of all such A form a disjoint partition of V because of the definition for minimality (in order for them to overlap in the hierarchical clustering, one of them needs to be ancestor of the other and this cannot happen because of minimality). Since $E_{OPT}(\lfloor t/2 \rfloor) \cap A$ for all such A forms a disjoint partition of $E_{OPT}(\lfloor t/2 \rfloor)$, the claim follows by summing up over all t. □

Theorem 3.3. *Given an unweighted graph G, the Recursive Sparsest Cut algorithm achieves an $O(\alpha_n)$ approximation for the hierarchical clustering problem.*

Proof. The proof follows easily by combining (3.3), (3.4), (3.5), Claim 3.2 and summing over all clusters A created by RSC. In particular, we get the following result for the overall performance guarantee:

$$cost_{RSC} = \sum_A r \cdot |E(B_1, B_2)| \leq$$
$$\leq \sum_A 4\alpha_n s |E_{OPT}(\lfloor r/2 \rfloor) \cap A| \leq$$
$$\leq 4\alpha_n \sum_A \sum_{t=r-s+1}^{r} |E_{OPT}(\lfloor t/2 \rfloor) \cap A| \leq$$
$$\leq 4\alpha_n \sum_{t=1}^{n} |E_{OPT}(\lfloor t/2 \rfloor)| \leq 8\alpha_n \cdot OPT$$

□

3.2 Using Balanced Cut instead of Sparsest Cut.

We are going to give an analysis similar to the above but instead of using the Sparsest Cut, we are going to use c-Balanced Cut as a black box. At the end, we will have a brief discussion on comparing the two approaches.

We will follow the same notation as above and we will use some of the facts and inequalities we previously proved about $E_{OPT}(t)$. Suppose again that we have a cluster A of size r and take $|E_{OPT}(\lfloor r/2 \rfloor) \cap A|$. The important observation here is that the partition $A_1, ..., A_k$ induced inside the cluster A by $OPT(\lfloor r/2 \rfloor) \cap A$ (where by design the size of each $|A_i| = \gamma_i |A|$, $\gamma_i \leq 1/2$), can be separated into two groups, let's say (C_1, C_2) such that $r/3 \leq |C_1|, |C_2| \leq 2r/3$ (see Figure 3). In other words we can demonstrate a c-Balanced Cut for $c = 1/3$. We know that:

$$|E(C_1, C_2)| \leq |E_{OPT}(\lfloor r/2 \rfloor) \cap A|$$

since we cut fewer edges when creating C_1, C_2.

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited

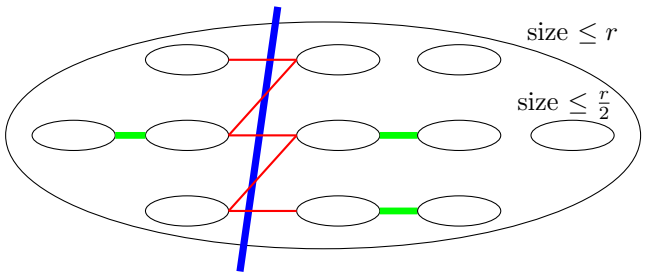

Figure 3: An illustration of $|E_{OPT}(r/2) \cap A|$ for a cluster A with size r. The small clusters have size $\leq \frac{r}{2}$ and the induced partition can be separated into two groups (C_1, C_2) that are $\frac{1}{3} : \frac{2}{3}$ balanced. The edges between those two groups (red) are also present in $E_{OPT}(r/2)$ (red and green).

At this point, we can invoke an α-pseudo-approximation algorithm for Balanced Cut [2] for the cluster A (let OPT_{BC} be the optimal cost of partitioning A into two clusters with sizes at least $r/3$) and we can get a partition into (B_1, B_2) with $c'r \leq |B_1|, |B_2| \leq (1-c')r$, (the constant $c' < 1/3$, that's why we call it pseudo-approximation) with the cost guarantee of:

$$|E(B_1, B_2)| \leq \alpha \cdot OPT_{BC} \leq \alpha \cdot |E(C_1, C_2)| \leq$$

$$\leq \alpha \cdot |E_{OPT}(\lfloor r/2 \rfloor) \cap A|$$

Looking at the cost of the hierarchical clustering which is $r \cdot |E(B_1, B_2)|$ and charging it as we did before to $s \cdot |E_{OPT}(\lfloor r/2 \rfloor) \cap A|$ where now $s \geq c'r \implies r \leq s/c'$ we get:

$$r \cdot |E(B_1, B_2)| \leq \alpha r \cdot |E(C_1, C_2)| \leq \alpha \frac{s}{c'} \cdot |E(C_1, C_2)| \leq$$

$$\leq \alpha \frac{s}{c'} \cdot |E_{OPT}(\lfloor r/2 \rfloor) \cap A| \leq \frac{\alpha}{c'} \sum_{t=r-s+1}^{r} |E_{OPT}(\lfloor t/2 \rfloor) \cap A|$$

Finally, all we have to do is sum up over all the clusters A (now in the summation we should write r_A, s_A instead of just r, s, since there is dependence in A) produced by the recursive Balanced Cut algorithm for Hierarchical Clustering and we get that we can approximate the HC objective function up to $O(\sqrt{\log n})$ (here we just substituted the best known value for the approximation guarantee α).

Remark 1. *The results for Balanced Cut generalize for the weighted case and for general cost functions. The reason we were interested in Balanced Cut is twofold: Firstly, recall that the running time for Sparsest Cut and Balanced Cut on a graph with n nodes and m edges is $\tilde{O}(m + n^{1+\epsilon})$. Now, the running time of Recursive Sparsest Cut might be worse off by a factor of n (you could have very unbalanced splits at every step) in the worst case. However, the running time for Recursive Balanced Cut is still $\tilde{O}(m + n^{1+\epsilon})$, because at each step you ensure that you have similar sized pieces and thus you make progress in the recursion. Secondly, we know that an α_n-approximation for Sparsest Cut yields an $O(\alpha_n)$-approximation for Balanced Cut, but not the other way. This means that if we could find a better Balanced Cut algorithm without improving Sparsest Cut, we could still use it for better Hierarchical Clustering.*

3.3 Generalized Cost Function and RSC.

In the original [10] paper introducing the objective function of hierarchical clustering, Dasgupta also considered the more general cost function: $cost_G(T) = \sum_{ij \in E} w_{ij} f(|\mathbf{leaves}(T[i \vee j])|)$, where f is defined on the non-negative reals, is strictly increasing, and has $f(0) = 0$ (e.g. $f(x) = \ln(1+x)$ or $f(x) = x^2$). For this more general cost function, he proved that a slightly modified greedy top-down heuristic (using $\frac{w(S, V \setminus S)}{\min((f|S|), f(|V \setminus S|))}$ with $\frac{1}{3}|V| \leq |S| \leq \frac{2}{3}|V|$, instead of Sparsest Cuts) continues to yield an $O(\alpha_n \cdot \log n \cdot c_f)$ approximation[1], where $c_f \triangleq \max_{1 \leq n' \leq n} \frac{f(n')}{f(n'/3)}$). Now, we analyze the previous RSC algorithm (with no modifications), but in the case of a weighted graph G and when we are trying to optimize the generalized cost function.

We again make the natural assumptions that the function f acting on the number of leaves in subtrees, is defined on the nonnegative reals, is strictly increasing and $f(0) = 0$ (also see Remark 2). We also define: $c_f \triangleq \max_{1 \leq n' \leq n} \frac{f(n')}{f(\lfloor n'/2 \rfloor) - f(\lfloor n'/4 \rfloor)}$. For what follows, we abuse notation slightly for ease of presentation and write $r/2, r/4$ etc. instead of $\lfloor r/2 \rfloor, \lfloor r/4 \rfloor$ etc. As in the simple unweighted case, we use here the same definitions for OPT and $E_{OPT}(t)$. Let $w(E_{OPT}(t))$ denote the total weight of the edges $E_{OPT}(t)$, i.e. the edges cut by OPT at level t, where we define $w(\emptyset) = 0$ and we also define $g(t) \triangleq f(t+1) - f(t)$. We note that $\sum_{t=0}^{r-1} g(t) = f(r) - f(0) = f(r)$.

Claim 3.4. $\sum_{t=0}^{n-1} w(E_{OPT}(t)) \cdot g(t) = OPT$

Proof. We will prove that the contributions of an edge $e = (u, v)$ to the LHS and RHS are equal. Let A ($|A| = r_e$) be the minimal cluster in the optimal solution that contains both u, v. The contribution of e to the

[1] There isn't a direct polynomial time implementation of this heuristic for arbitrary functions f to the best of our knowledge; however, a heuristic based on balanced cut will achieve similar guarantees.

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited

RHS is: $w_e \cdot f(r_e)$. As for the contribution to the LHS, since A is minimal and $|A| = r_e$, we deduce that $e \in OPT(t), \forall t < r_e$. Also for levels $t \geq r_e$ we have $e \in A$ or some superset of A and thus $e \notin OPT(t)$ Hence the contribution to the LHS is: $w_e \cdot \sum_{t=0}^{r_e-1} g(t) = w_e \cdot f(r_e)$. □

Focus on a cluster A ($|A| = r$) in the solution produced by the algorithm. Let $cut(A)$ denote the edges in A cut by partitioning A. This contributes $w(cut(A)) \cdot f(r)$ to the objective. We will charge our cost using the following quantity related to the optimum solution: $\sum_{t=r/4}^{r/2-1} w(E_{OPT}(t) \cap A) \cdot g(t)$.

For that, we look at $OPT(r/2) \cap A$ and let's say that clusters $A_1, A_2, ..., A_k$ are induced by this partition, each being of size $|A_i| = \gamma_i |A| \leq |A|/2 = r/2$ ($\gamma_i \leq 1/2$). Then,

$$SC(A) \leq \frac{\sum_i w(A_i, A \setminus A_i)}{r^2 \sum_i \gamma_i(1-\gamma_i)} \leq \frac{2 \cdot w(E_{OPT}(r/2) \cap A)}{r^2 \cdot 1/2}$$

where $SC(A)$ is the optimum sparsest cut (value) for A. Since we used an α_n-approximation,

$$\frac{w(cut(A))}{s(r-s)} \leq \alpha_n SC(A) \leq \frac{4\alpha_n (w(E_{OPT}(r/2) \cap A)}{r^2} \implies$$

$$w(cut(A)) f(r) \leq \frac{4\alpha_n s}{r} w(E_{OPT}(r/2) \cap A) f(r) \tag{3.6}$$

Since $w(E_{OPT}(t) \cap A) \geq w(E_{OPT}(t+1) \cap A)$, we have:

$$\sum_{t=r/4}^{r/2-1} w(E_{OPT}(t) \cap A) \cdot g(t) \geq$$

$$\geq w(E_{OPT}(r/2) \cap A) \cdot \sum_{t=r/4}^{r/2-1} g(t) =$$

$$= (f(r/2) - f(r/4)) \cdot w(E_{OPT}(r/2) \cap A) \tag{3.7}$$

Using equations (3.6), (3.7), we get that:

$$w(cut(A)) \cdot f(r) \leq \tag{3.8}$$

$$\leq 4\alpha_n \cdot \frac{s}{r} \cdot \frac{f(r)}{f(r/2) - f(r/4)} \cdot \sum_{t=r/4}^{r/2-1} w(E_{OPT}(t) \cap A) \cdot g(t)$$

We now sum up the cost contributions of all clusters created in our hierarchical clustering solution. Let $s(A)$ be the size of the smaller piece produced in partitioning A.

$$cost_{RSC} = \sum_A w(cut(A)) \cdot f(|A|) \leq \tag{3.9}$$

$$\leq 4\alpha_n \cdot c_f \sum_A \frac{s(A)}{|A|} \sum_{t=|A|/4}^{|A|/2-1} w(E_{OPT}(t) \cap A) \cdot g(t)$$

To complete our argument we need to make the comparison between OPT which is: $\sum_{t=0}^{n-1} w(E_{OPT}(t)) \cdot g(t)$ and the sum

$$\sum_A \frac{s(A)}{|A|} \sum_{t=|A|/4}^{|A|/2-1} w(E_{OPT}(t) \cap A) \cdot g(t) \tag{3.10}$$

where the first summation goes over all clusters A in the solution we produce.

Claim 3.5. $\sum_A \frac{s(A)}{|A|} \sum_{t=|A|/4}^{|A|/2-1} w(E_{OPT}(t) \cap A) \cdot g(t) \leq 2 \cdot \sum_{t=0}^{n-1} w(E_{OPT}(t)) \cdot g(t)$

Proof. Consider some edge $e = (u, v) \in E_{OPT}(t)$. Focus on sets A in the solution produced such that $e \in E_{OPT}(t) \cap A$ so that e contributes to the term $\sum_{t=|A|/4}^{|A|/2-1} w(E_{OPT}(t) \cap A) \cdot g(t)$ in the LHS. For all such clusters A, we need to have: $|A|/4 \leq t < |A|/2 \implies 2t < |A| \leq 4t$.

Let $A_1, A_2, ..., A_{k-1}$ be the sets for which the term $w(E_{OPT}(t) \cap A)$ appears: A_1 is the largest cluster (satisfying $2t < |A_1| \leq 4t$) that contains the edge $e = (u, v)$ and when split we call its larger piece A_2 (again this set contains e) etc., A_{k-1} is the last set for which the term appears and (u, v) does not appear in A_k (A_k is the larger piece of the two that we got when we partitioned A_{k-1}). We have:

$$\sum_{i=1}^{k-1} \frac{s(A_i)}{|A_i|} = \tag{3.11}$$

$$= \frac{|A_1| - |A_2|}{|A_1|} + \frac{|A_2| - |A_3|}{|A_2|} + ... + \frac{|A_{k-1}| - |A_k|}{|A_{k-1}|} \leq$$

$$\leq \frac{\sum_{i=1}^{k-1} |A_i| - |A_{i+1}|}{\min_i |A_i|} \leq \frac{|A_1|}{2t} \leq 2.$$

(the constant can be optimized, but it does not change the asymptotic bound). Thus the contribution of every edge $e \in E_{OPT}(t)$ to the LHS is at most $2w_e g(t)$. Note that this is exactly the contribution to the RHS. This establishes the claim.

□

Theorem 3.6. *RSC achieves an $O(c_f \cdot \alpha_n)$ approximation of the generalized objective function for Hierarchical Clustering.*

Proof. We will prove the theorem by combining (3.9),

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited

Claim 3.5 and Claim 3.4. In particular, from (3.9),

$$cost_{RSC} \leq 4\alpha_n \cdot c_f \sum_A \frac{s(A)}{|A|} \sum_{t=|A|/4}^{|A|/2-1} w(E_{OPT}(t) \cap A) \cdot g(t)$$

Combining the above with Claim 3.5, we get that the total cost of the RSC is at most: $cost_{RSC} \leq 8c_f\alpha_n \cdot \sum_{t=0}^{n-1} w(E_{OPT}(t)) \cdot g(t)$. Finally, using the Claim 3.4 we get: $cost_{RSC} \leq (8c_f\alpha_n) \cdot OPT = O(c_f \cdot \alpha_n) \cdot OPT$. □

Remark 2. *In order for our guarantee to be useful, we need c_f to be a constant (or a slowly growing quantity). This would mean that f is polynomially growing. We observe that in the case where the function f is exponentially growing then our guarantee is not interesting (and in fact we may need to use a different strategy than RSC) and in the case f is logarithmic, then we would get a factor $\approx O(\alpha_n \log n)$ approximation, which is the same guarantee as [10].*

4 Hierarchical Clustering Hardness and the Small Set Expansion Hypothesis

In this section, we prove a strong inapproximability result, showing that, even in unweighted graphs (i.e. unit cost edges), the Hierarchical Clustering objective is hard to approximate to within any constant factor, assuming the *Small Set Expansion* hypothesis.

4.1 *SSE* and hardness amplification.

Given a graph $G(V, E)$, define the following quantities for non-empty subsets $S \subset V$: normalized set size $\mu(S) \triangleq |S|/|V|$, and edge expansion $\Phi_G(S) \triangleq \frac{|E(S, V \setminus S)|}{\sum_{i \in S} d_i}$ (here d_i is the degree of i). The *Small Set Expansion* hypothesis was introduced by Raghavendra and Steurer [31].

Problem 4.1 (Small-Set Expansion(η, δ))**.** *Given a regular graph $G(V, E)$, distinguish between the following two cases:*
Yes: *There exists a non-expanding set $S \subseteq V$ with $\mu(S) = \delta$ and $\Phi_G(S) \leq \eta$.*
No: *All sets $S \subseteq V$ with $\mu(S) = \delta$ are highly expanding with $\Phi_G(S) \geq 1 - \eta$.*

Hypothesis 4.2 (Hardness of approximating Small-Set Expansion)**.** *For all $\eta > 0$, there exists $\delta > 0$ such that the promise problem* Small-Set Expansion(η, δ) *is NP-hard.*

[31] showed that the Small Set Expansion Hypothesis implies the Unique Games Conjecture of Khot [23]. A decision problem is said to be SSE-hard if Small-Set Expansion(η, δ) reduces to it by a polynomial time reduction for some constant η and all $\delta > 0$. Raghavendra, Steurer and Tulsiani [32] showed the following hardness amplification result for graph expansion (see Preliminaries for Gaussian Graphs definitions):

Theorem 4.3. *For all $q \in \mathbb{N}$ and $\epsilon, \gamma > 0$, it is SSE-hard to distinguish between the following two cases for a given graph $H = (V_H, E_H)$:*
Yes: *There exist q disjoint sets $S_1, ..., S_q \subseteq V_H$ satisfying for all $l \in [q]$: $\mu(S_l) = 1/q$ and $\Phi_H(S_l) \leq \epsilon + o(\epsilon)$.*
No: *For all sets $S \subseteq V_H$: $\Phi_H(S) \geq \Phi_{\mathcal{G}(1-\epsilon/2)}(\mu(S)) - \gamma/\mu(S)$, where $\Phi_{\mathcal{G}(1-\epsilon/2)}(\mu(S))$ is the expansion of sets of volume $\mu(S)$ in the infinite Gaussian graph $\mathcal{G}(1-\epsilon/2)$.*

4.2 Hierarchical Clustering Hardness.

Now we are ready to prove our main hardness result. Our proof follows the argument of [32] for establishing the hardness of Minimum Linear Arrangement. We prove the following:

Theorem 4.4. ***(Hardness of Hierarchical Clustering).*** *For every $\epsilon > 0$, it is SSE-hard to distinguish between the following two cases for a given graph $G = (V, E)$, with $|V| = n$:*
Yes: *There exists a decomposition tree T of the graph such that $cost_G(T) \leq \epsilon n|E|$*
No: *For any decomposition tree T of the graph $cost_G(T) \geq c\sqrt{\epsilon} n|E|$.*

Proof. We apply Theorem 4.3 for the following values: $q = \lceil 2/\epsilon \rceil, \epsilon' = \epsilon/3$ and $\gamma = \epsilon$. We need to first handle the **Yes** case. We get that the vertices can be divided into sets $S_1, S_2, ..., S_q$, each having size $n/q = n\epsilon/2$, such that at most $\epsilon' + o(\epsilon')$ fraction of edges leave the sets (i.e. go across sets). Now consider the hierarchical clustering solution that first partitions the vertices into the sets $S_1, S_2, ..., S_q$ and then partitions each S_i arbitrarily. Edges inside the set S_i contribute at most $|S_i|$ to the objective function and this is $|S_i| = n/q = \epsilon n/2$. Moreover, edges whose endpoints are in different sets will have contribution at most n; but there are at most $\epsilon/2$ fraction of such edges and so the overall objective for this hierarchical clustering solution is at most $\epsilon n|E|$.

Now, we handle the **No** case by using the argument of [32] for Minimum Linear Arrangement that follows from an observation of [12] and the fact that the objective function of Minimum Linear Arrangement is always less than the cost of Hierarchical Clustering. To see the latter, observe that when we have a hierarchical clustering tree T, if we consider the ordering of the vertices (leaves) as in the DFS order, then the stretch of an edge (u, v) that is cut, can be at most the size of the subtree which corresponds to that edge and this is exactly the quantity: $|\mathbf{leaves}(T[u \vee v])|$. Since we know ([32, 12]) that in the **No** case, for all orderings

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited

$\pi : V \to [n], \mathbb{E}_{(u,v)\sim E}[|\pi(u) - \pi(v)|] \geq c\sqrt{\epsilon}n$, it immediately follows that: $cost_G(T) \geq c\sqrt{\epsilon}n|E|$.

□

5 Approximation for HC using SDP

In this section, we present our SDP relaxation for HC based on spreading metrics, we point out its relation with the SDP relaxation of *k-balanced partitioning* in [24] and we prove that it is an $O(\sqrt{\log n})$ approximation for both the simple and the generalized cost function.

5.1 Writing the SDP. We view a hierarchical clustering of n data points as a collection of partitions of the data, one for each level $t = n-1, \ldots, 0$ (where level 0 is identical to level 1, for convenience). The partition for a particular level t satisfies the property that every cluster has size at most t; additionally, for every vertex i, the cluster containing vertex i at level t is the maximal cluster in the hierarchy with size at most t. The partition at level $(t-1)$ is a refinement of the partition at level t. Note that the partition corresponding to $t = 1$ must consist of n singleton clusters. We represent the partition at level t by the set of variables x_{ij}^t, $i, j \in V$, where $x_{ij}^t = 1$ if i and j are in different clusters in the partition at level t and $x_{ij}^t = 0$ if i and j are in the same cluster. We point out some properties of these variables x_{ij}^t satisfied by an integer solution corresponding to an actual hierarchical clustering:

1. **refinement:** $x_{ij}^t \leq x_{ij}^{t-1}$. If i and j are separated at level t, then they continue to be separated at level $t-1$.
2. **triangle inequality:** $x_{ij}^t + x_{jk}^t \geq x_{ik}^t$. In the clustering at level t, if i and j are in the same cluster, j and k are in the same cluster, then i and k are in the same cluster.
3. **ℓ_2^2 metric:** The triangle inequality condition implies that x_{ij}^t is a metric. Further, we can associate unit vectors v_i^t with vertices i at level t such that $x_{ij}^t = \frac{1}{2}||v_i^t - v_j^t||_2^2$. In order to do this, all vertices in the same cluster at level t are assigned the same vector, and vertices in different clusters are assigned orthogonal vectors.
4. **spreading:** $\sum_j x_{ij}^t \geq n - t$. For the clustering at level t, there are at most t vertices in the same cluster as i. Hence there are at least $n - t$ vertices in different clusters. For each such vertex j, $x_{ij}^t = 1$ implying the inequality.
5. **cluster size:** The size of the smallest cluster in the hierarchy containing both vertices i and j is given by $1 + \sum_{t=1}^{n-1} x_{ij}^t$. Suppose C is the smallest cluster containing both i and j. Then for $t \geq |C|$, the partition at level t must contain C or some superset of C. Hence $x_{ij}^t = 0$ for $t \geq |C|$. For $t < |C|$, the clustering at level t must have i and j in different clusters, hence $x_{ij}^t = 1$. Hence $\sum_{t=1}^{n-1} x_{ij}^t = |C| - 1$. Finally, we can write the SDP relaxation SDP-HC as follows:

$$\min \sum_{t=0}^{n-1} \sum_{ij \in E} x_{ij}^t w_{ij} = \min \sum_{t=0}^{n-1} \sum_{ij \in E} \frac{1}{2}\|v_i^t - v_j^t\|_2^2 w_{ij},$$

(SDP-HC)

$$\text{such that:} \quad x_{ij}^t \leq x_{ij}^{t-1}, \quad t = n-1, n-2, \ldots 1$$

$$x_{ij}^0 = 1, \forall i, j \in V \text{ and } x_{ij}^t \leq 1, \forall i, j, t$$

$$x_{ij}^t = \frac{1}{2}\|v_i^t - v_j^t\|_2^2 \text{ and } \|v_i^t\|_2^2 = 1, \forall i \in V$$

$$x_{ij}^t \leq x_{jk}^t + x_{ik}^t, \forall i, j, k \in V, \ \forall t \text{ and } \sum_j x_{ij}^t \geq n - t, \forall i, t$$

It is easy to see that an optimal solution to SDP-HC can be computed in polynomial time (see for example the arguments in [24]). By the preceding discussion, we have shown that SDP-HC is a valid relaxation for HC:

Lemma 5.1. *The value of an optimal solution to SDP-HC can be computed in polynomial time, and gives a lower bound on the cost of an optimal solution to the hierarchical clustering problem.*

5.2 Connections of SDP-HC with Balanced Partitioning. The authors of [24] write an SDP relaxation for the problem of k-Balanced Partitioning (k-BP) which was the following (SDP-k-BP):

$$\text{(SDP-}k\text{-BP)} \qquad \min \sum_{ij \in E} w_{ij} \cdot \frac{1}{2}\|v_i - v_j\|_2^2,$$

$$\text{such that: } \|v_i - v_j\|_2^2 + \|v_j - v_k\|_2^2 \geq \|v_i - v_k\|_2^2, \ \forall i, j, k \in V$$

$$\sum_{j \in S} \frac{1}{2}\|v_i - v_j\|_2^2 \geq |S| - \frac{n}{k}, \ \forall S \subseteq V, i \in S$$

Their result was that the above relaxation is an $O(\sqrt{\log k \log n})$ approximation (bi-criteria $\nu = 2$) algorithm for k-BP, that will create pieces of size at most $2n/k$.

Claim 5.2. *Let A be a cluster of size r. SDP-HC solution restricted to set A, at level $t = r/4$ is a valid solution for k-balanced partitioning based on the SDP-k-BP relaxation, where $k = 4$.*

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited

Proof. First of all, we have that:

$$SDP_A(t) \triangleq \sum_{ij\in E, i,j\in A} x_{ij}^t w_{ij} = \sum_{ij\in E, i,j\in A} \frac{1}{2}\|v_i^t - v_j^t\|_2^2 w_{ij}$$

Basically, we take the SDP-HC solution for the entire instance and we focus on some part of it, i.e. we focus on the terms of the objective function that are relevant for the cluster A.

To prove the claim all we need to do is to compare the set of constraints imposed by SDP-HC and SDP-k-BP. In SDP-HC, we have some additional constraints: $x_{ij}^t \leq 1$ and $v_i^t \leq 1$, but that is fine since imposing extra constraints just makes a stricter relaxation. Now let's look at the spreading constraints: In SDP-HC we have $\sum_j x_{ij}^t \geq n - t \implies \sum_{j\in S} x_{ij}^t \geq |S| - t$ which is basically the SDP-k-BP spreading constraints. Thus, by looking at the SDP-HC solution restricted to set A ($|A| = r$), at level $t = r/4$, we can get a valid 4-balanced partitioning solution of A. □

In order to produce a hierarchical clustering from the SDP solution, we recursively partition V in a top down fashion: while partitioning a cluster A, we use the SDP-HC solution restricted to set A at level $t = |A|/4$ as a valid solution for 4-balanced partitioning and invoke the algorithm of [24] as a black box. Let E_A be the edges cut by the algorithm when splitting cluster A. From the analysis of [24] , we get that (for us $k = 4$, so $\log k$ is constant):

$$w(E_A) \leq O(\sqrt{\log n}) \cdot SDP_A(r/4) \tag{5.12}$$

and we partition A into pieces of size at most $\leq 2 \cdot r/4 = r/2$ (bi-criteria). In the analysis that follows, we will use this result as a black box.

5.3 $O(\sqrt{\log n})$ approximation for Hierarchical Clustering. Now we go on to see that the integrality gap of our SDP-HC is $O(\sqrt{\log n})$. Let r_A be the size of a cluster A in the solution produced. For our charging argument, we observe that we pay $r_A \cdot w(E_A)$ where E_A are the edges cut by the KNS [24] algorithm when partitioning A. We will charge this cost to $\sum_{t=r_A/8+1}^{r_A/4} SDP_A(t) \geq \frac{r_A}{8} SDP_A(r_A/4)$ (note that as t decreases more edges are cut). Thus, using [24], the total cost of the solution produced (where $cost_{HC}$ is the cost of the rounding algorithm's solution and r_A depends on A):

$$cost_{HC} = \sum_A r_A \cdot w(E_A) \leq \tag{5.13}$$

$$\leq O(\sqrt{\log n}) \sum_A \sum_{t=r_A/8+1}^{r_A/4} SDP_A(t)$$

Claim 5.3. $\sum_A \sum_{t=|A|/8+1}^{|A|/4} SDP_A(t) \leq O(SDP\text{-}HC)$.

Proof. The flavor of this analysis is similar to our RSC result from Section 3. Let's look at an edge $e = (u, v)$ at a fixed level t. For which sets A do we get the term $\text{SDP}_A(t)$ (i.e. both endpoints $u, v \in A$) ? Since $t \in \big(|A|/8, |A|/4\big] \implies 4t \leq |A| < 8t$. There can be at most one such $|A|$ containing both u, v, so LHS is charged only once (of course the RHS is charged $x_{ij}^t w_{ij}$ for that edge). To see why A is unique, suppose we had two such clusters $|A_1|, |A_2|$ that both contained u, v with their sizes $|A_1|, |A_2| \in [4t, 8t)$. Since we have a hierarchical decomposition, one of A_1, A_2 is ancestor of the other. Let's say, wlog, A_1 is ancestor of A_2. But then, all of its descendants are of size below the range $[4t, 8t)$ due to the 4-partition, which is a contradiction. □

Remark 3. *In the above analysis, whenever we write $|A|/4$ we mean $\lfloor |A|/4 \rfloor$. However, this will not affect the result. Additionally, we used the bound $O(SDP\text{-}HC)$, because some extra constants might be introduced whenever the set A is small ($|A| < 8$).*

Theorem 5.4. *The cost of the solution produced by the SDP-HC rounding algorithm is within a factor of $O(\sqrt{\log n})$ from the SDP value.*

Proof. Using Claim 5.3 and (5.13) we get that $cost_{HC} \leq O(\sqrt{\log n})\cdot$SDP-HC. □

5.4 The case of the generalized cost function. Now, we consider the performance of SDP-HC-gen for the generalized cost function and we show essentially the same approximation guarantees. We note that the SDP-HC-gen is essentially the same as SDP-HC where each term in the objective function is multiplied by $g(t) = f(t+1) - f(t)$. Formally:

$$\text{(SDP-HC-gen)} \qquad \min \sum_{t=0}^{n-1} g(t) \sum_{ij\in E} x_{ij}^t w_{ij} =$$

$$= \min \sum_{t=0}^{n-1} g(t) \sum_{ij\in E} \frac{1}{2}\|v_i^t - v_j^t\|_2^2 w_{ij}$$

We can easily prove the following claim for the generalized cost:

Claim 5.5. $\sum_A \sum_{t=|A|/8+1}^{|A|/4} SDP_A(t) \cdot g(t) \leq$
$\leq O(SDP\text{-}HC\text{-}gen)$.

Proof. The proof of the claim is identical to the proof we gave above for Claim 5.3 with only difference being that we need to multiply by $g(t)$ each term. □

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited

Theorem 5.6. *The cost of the solution produced by the SDP-HC-gen rounding algorithm is within a factor of* $O(\sqrt{\log n} \cdot c_f)$ *from the SDP value where* $c_f \triangleq \max_{r \in \{1,\ldots,n\}} \frac{f(r)}{f(r/4) - f(r/8)}$.

Proof. Let A be a cluster of size $|A| = r$ and let $g(t) = f(t+1) - f(t)$. We want to compare the cost of OPT for splitting A with our solution $\text{SDP}_A(t)$ for levels $t = r/8+1, ..., r/4$. Using (5.12) (once again $cost_{HC}(A)$ is just the cost incurred by the rounding algorithm when we partitioned cluster A):

$$cost_{HC}(A) = f(r) \cdot w(E_A) \leq$$

$$\leq O(\sqrt{\log n}) f(r) \cdot SDP_A(r/4) \leq$$

$$\leq O(\sqrt{\log n}) \frac{f(r)}{f(r/4) - f(r/8)} \sum_{t=r/8+1}^{r/4} SDP_A(t) \cdot g(t) \leq$$

$$\leq O(\sqrt{\log n}) \cdot c_f \cdot \sum_{t=r/8+1}^{r/4} SDP_A(t) \cdot g(t).$$

Using now Claim 5.5 and summing over all clusters A in the hierarchical clustering we get (for the sum we should substitute r by r_A): $cost_{HC} \leq O(\sqrt{\log n}) \cdot c_f \cdot$ SDP-HC-gen. □

Remark 4. *As in Remark 2, here f should be polynomially growing.*

6 An LP-based $O(\log n)$ approximation via spreading metrics

In this section, we present an approximation algorithm with ratio $O(\log n)$ based on an LP relaxation. The key ingredients for our purposes are the spreading metrics paradigm of [15] and a graph decomposition lemma by Bartal ([5]).

6.1 Bartal's decomposition.

Bartal presented a graph decomposition lemma and used it in order to prove an $O(\log n)$ approximation guarantee for the spreading metrics paradigm in undirected graphs; thus, he improved the results for many problems considered in [15]. How does the decomposition work? At a high level, it tries to find a low diameter cluster within the graph, such that the weight of the cut created is small with respect to the weight of the cluster. The decomposition is essentially based on a careful implementation of the decomposition of [17]. In what follows, we state Bartal's improved approximation guarantee, summarized in the following theorem, and then briefly highlight the main steps in achieving it; proofs can be found in [5].

Theorem 6.1. *There exists an $O(\log n)$ approximation for problems in the spreading metrics paradigm.*

We first need to introduce some notation, before explaining the main steps of the proof. Let $G = (V, E)$ be an undirected graph with two weight functions $w, l : E \to \mathbb{R}^+$. We interpret $l(e)$ to be the length of the edge e, and the distance $d(u, v)$ between pairs of vertices u, v in the graph, is determined by the length of the shortest path between them. Given a subset $S \subseteq V$, $G(S)$ denotes the subgraph of G induced by S. Given partition $(S, \bar{S})$, let $\Gamma(S) = \{(u, v) \in E; u \in S, v \in \bar{S}\}$ and $cut(S) = \sum_{e \in \Gamma(S)} w(e)$. Given a subgraph $H = (V_H, E_H)$ of G, let d_H denote the distance in H, let $\Delta(H)$ denote the diameter of H, and $\Delta = \Delta(G)$. We also define the volume of H, $\phi(H) = \sum_{e \in E_H} w(e) l(e)$.

Informally, the first step needed is to find in G, a partition $(S, \bar{S})$ which is "good", i.e. with low cut value $cut(S)$ with respect to a generalized notion of its volume. The decomposition becomes useful when it is applied recursively. Note, that this is particularly important for our main application which is hierarchical clustering. This gives rise to a recursive approach where we find a good cut, creating two subgraphs and then recuse on each of them. Towards this direction, for the second step needed, we focus on applications which are associated with a cost function *cost* over subgraphs $\hat{G}$ of G, that is nonnegative, 0 on singletons and obeys the following natural recursion rule:

$$cost(\hat{G}) \leq cost(\hat{G}(S)) + cost(\hat{G}(\bar{S})) + \Delta(\hat{G}) \cdot cut(S).$$

Lemma 6.2. *Any cost function defined by the above recursion rule obeys $cost(G) \leq O(\log(\phi/\phi_0)) \cdot \phi(G)$, where $\phi = \phi(G)$ and ϕ_0 is the minimum value of $\phi(\hat{G})$ on non-singleton subgraphs $\hat{G}$.*

Finally, we can obtain a $O(\log n)$ approximation bound (depending only on n), by modifying the above lemma slightly, by associating a volume $\phi(G)/n$ with the nodes, like in [17]. This ensures that $\phi_0 \geq \phi(G)/n$ and by substituting we get what we want:

Lemma 6.3. *The function defined by the above recursion rule obeys $cost(G) \leq O(\log n) \cdot \phi(G)$.*

Now we turn our attention to the connection with the spreading metrics paradigm. Having the definition of a spreading metric in mind (see Section 6.2) and the previous three steps, we may obtain Theorem 6.1. (for details see [5])

6.2 LP relaxation and the Spreading Metrics paradigm.

We prove here that the hierarchical clustering objective function defined above falls into the *divide and conquer approximation algorithms via spreading metrics* paradigm of [15].

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited

The spreading metric paradigm applies to minimization problems on undirected graphs $G = (V, E)$ with edge weights $w(e) \geq 1$. We also have an auxiliary graph H and a scaler function on subgraphs of H (e.g. size of the components of H). A decomposition tree T is a tree with nodes corresponding to non-overlapping subsets of V, forming a recursive partition of the nodes V. For a node t of T, we denote by V_t the set of vertices in the subtree rooted at t. Associated are the subgraphs G_t, H_t induced by V_t. Let F_t be the set of edges that connect vertices that belong to different children of t, and $w(F_t) = \sum_{e \in F_t} w(e)$. The cost of T is $cost(T) = \sum_{t \in T} scaler(H_t) \cdot w(F_t)$.

Definition 6.1. *A spreading metric is a function on the edges of the graph $l : E \to \mathbb{R}^+$ satisfying the following two properties:*

1. *Lower bound property: The volume of the graph $\sum_{e \in E} w(e)l(e)$ is a lower bound on the optimal cost.*
2. *Diameter property: For any $U \subseteq V$ and H_U the subgraph of H induced by U, has diameter $\Delta(U) \geq scaler(H_U)$.*

We closely follow their formulation for the Linear Arrangement problem, which also falls into the spreading metrics paradigm, but we make the necessary semantic changes. We need to show the divide and conquer applicability and the spreading metrics applicability of their result for our problem.

Firstly, to establish the divide and conquer applicability we consider any binary decomposition tree T that fully decomposes the problem (we normalize the edge weights by dividing with the minimum edge weight). Note that there is a $1-1$ correspondence between the leaves of T and the vertices of G. The solution to the hierarchical clustering problem that is represented by T is easily given by the cuts, in G, induced by the internal nodes of T. The cost of the tree T is:

$$cost_G(T) = \sum_{t \in T} |V_t| w(F_t). \tag{6.14}$$

where V_t and F_t are the set of vertices and cut corresponding to the tree node t and $w(F_t)$ is the total weight of the edges cut at this internal node t. We need to show that this cost bounds the cost of solutions built up from T. For this we prove that for every tree node t the cost of the subtree rooted at t, denoted T_t, bounds the cost of solutions built up from T_t to the hierarchical clustering problem for the subgraph of G induced by the set of vertices V_t. We prove the claim by induction on the level of the tree nodes. The claim clearly holds for all leaves of T. Consider an internal tree node $t \in T$ and denote its two children by t_L and t_R. By induction the claim holds for both t_L and t_R. The solution represented by T_t is given by concatenating the solutions represented by T_{t_L} and T_{t_R}. Note that the additional cost is at most $|V_t|$ times the capacity of the cut F_t that separates V_{t_L} from V_{t_R}. We get

$$cost_G(T_t) \leq cost_G(T_{t_L}) + cost_G(T_{t_R}) + |V_t| w(F_t).$$

The inductive claim follows.

We now show how to compute the spreading metric that assigns length $l(e)$ to an edge $e \in E$ of the graph. Consider the following linear program (LP1):

$$\min \sum_{e \in E} w(e) \cdot l(e) \quad \text{such that:} \tag{6.15}$$

$$\forall U \subseteq V, \forall v \in V : \sum_{u \in U} dist_l(u, v) \geq \frac{1}{2}(|U|^2 - 1) \tag{6.16}$$

$$\forall e \in E : l(e) \geq 0 \tag{6.17}$$

In the linear program, we follow the notation that regards $l(e)$ as edge lengths, and $dist_l(u, v)$ is the length of the shortest path from u to v. We will refer to constraint (6.16) as the spreading constraint. The linear program can be solved in polynomial time since we can construct a separation oracle. In order to verify that the spreading constraint (6.16) is satisfied, for each vertex v, we sort the vertices in V in increasing order of distance $dist_l(u, v)$ and verify the spreading constraint for all prefixes U of this sorted order.

Lemma 6.4. *Let $l(e)$ denote a feasible solution of the linear program. For every $U \subseteq V$ with $(|U| > 1)$, and for every vertex $v \in U$ there is a vertex $u \in U$ for which $dist_l(u, v) \geq \frac{1}{2}(|U| - 1)$.*

Proof. The average distance of a node $u \in U - \{v\}$ from v is greater than $\frac{1}{2}(|U| - 1)$, because of the constraint corresponding to U and v. Therefore, there exists a vertex $u \in U$ whose distance from v is at least the average distance from v, and the lemma follows, since $dist_l(u, v) \geq \frac{1}{2}(|U| - 1)$. $(|U| > 1)$ □

Note that the previous lemma comes short of the diameter guarantee by a factor of 2: while the diameter guarantee requires that the diameter of a subset U be greater than $|U|$, the proven bound is only $|U|/2$. However, this only affects the constant in the approximation factor. In the next lemma, we prove that the volume of an optimal solution of the linear program satisfies the lower bound property.

Lemma 6.5. *The cost of an optimal solution of the linear program is a lower bound on the cost of an optimal hierarchical clustering of G.*

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited

Proof. Consider any binary hierarchical clustering given by the sequence of cuts in the decomposition tree T and define $l(e) = |\mathbf{leaves}(T[i \vee j])|$ for edge $e = (i, j) \in E$. It is easy to see that this is indeed a metric and it is actually an ultrametric. We show that $l(e)$ is a feasible solution for the linear program above. The cost $\sum_{e \in E} w(e) \cdot l(e)$ equals the cost of the hierarchical clustering induced by the tree T. The feasibility of $l(e)$ is proved as follows: Consider a subset $U \subseteq V$ and a vertex $v \in U$. We observe that the average distance from v of the vertices in U will be minimized when U is "packed around" v, meaning that with each cut we peel off only one vertex at a time. We have that:

$$\sum_{u \in U} dist_l(u, v) = 2 + 3 + ... + |U| \geq \frac{1}{2}(|U|^2 - 1)$$

Hence, $l(\cdot)$ is a feasible solution and the lemma follows. □

With the above two lemmas we have proved that our hierarchical clustering objective function falls into the spreading metrics paradigm, because it satisfies the lower bound property and the diameter property. Using Bartal's decomposition and specifically Theorem 6.1 from Section 6 we get an approximation guarantee of $O(\log n)$:

Theorem 6.6. *There exists an $O(\log n)$ approximation for the hierarchical clustering objective function defined by (1.1).*

7 Conclusion and Further Research

We proved that the recently introduced objective function for hierarchical clustering in [10], can be approximated within a factor of $O(\alpha_n)$ by recursive application of an α_n-approximate Sparsest Cut (or c-Balanced Cut) algorithm and within $O(\sqrt{\log n})$ using a spreading metric SDP relaxation. We also proved that it is hard to approximate the HC objective function to within any constant factor assuming the *Small Set Expansion Hypothsesis*. We finally presented an LP based $O(\log n)$ approximation by showing that HC falls into the spreading metrics paradigm of [15].

A natural question is whether better approximation factors are possible for this particular problem. This may be difficult because of the much more basic problem of c-Balanced Cut. It seems implausible that we would get a better approximation for hierarchical clustering without getting an improvement in the current best approximation guarantee for c-Balanced Cut.

Another direction for research is going beyond worst case analysis for this problem. What can we say about exact recovery on γ-stable instances under the Bilu-Linial [6] notion of stability? Roughly speaking, a stable instance under this notion has the property that the structure of the optimum solution does not change even if weights in the instance are changed by a factor of γ. For example, in [28] they show that the standard SDP relaxation for Max-Cut is integral if the instance is sufficiently stable ($\gamma \geq c\sqrt{\log n}\log n$ for some absolute constant $c > 0$). Stability for clustering problems has been extensively studied; see [3, 6, 28] and references therein. It would be interesting to study stable instances for hierarchical clustering, in particular the performance of the recursive sparsest cut algorithm (or our SDP-HC) on such instances. This would not only explain the success of certain heuristics for HC based on finding sparsest cuts, but also justify their use in practice (assuming that stability is a good model for instances in real applications). Finally, we also find interesting the scenario where the input graph is drawn from a probability distribution for which there is a truly hierarchical structure. Can we then prove that our algorithms presented here or another suitable algorithm will indeed find a hierarchical structure close to the actual underlying hierarchy?

Acknowledgements

This research was supported by NSF grants CCF-1565581, CCF-1617577, CCF-1302518 and a Simons Investigator Award. The authors would like to thank the anonymous SODA reviewers for their time and their useful comments.

References

[1] C. Ambühl, M. Mastrolilli, and O. Svensson. Inapproximability results for maximum edge biclique, minimum linear arrangement, and sparsest cut. *SIAM Journal on Computing*, 40(2):567–596, 2011.

[2] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):5, 2009.

[3] M. F. Balcan and Y. Liang. Clustering under perturbation resilience. In *International Colloquium on Automata, Languages, and Programming*, pages 63–74. Springer, 2012.

[4] M.-F. Balcan, Y. Liang, and P. Gupta. Robust hierarchical clustering. *Journal of Machine Learning Research*, 15:3831, 2014.

[5] Y. Bartal. Graph decomposition lemmas and their role in metric embedding methods. In *European Symposium on Algorithms*, pages 89–97. Springer, 2004.

[6] Y. Bilu and N. Linial. Are stable instances easy? *Combinatorics, Probability and Computing*, 21(05):643–660, 2012.

[7] M. Charikar, V. Guruswami, and A. Wirth. Clustering with qualitative information. In *Foundations of Com-*

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited

puter Science, 2003. Proceedings. 44th Annual IEEE Symposium on, pages 524–533. IEEE, 2003.

[8] M. Charikar, M. T. Hajiaghayi, H. Karloff, and S. Rao. l_2^2 spreading metrics for vertex ordering problems. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1018–1027. Society for Industrial and Applied Mathematics, 2006.

[9] K. Chaudhuri, S. Dasgupta, S. Kpotufe, and U. Luxburg. Consistent procedures for cluster tree estimation and pruning. *IEEE Transactions on Information Theory*, 60(12):7900–7912, 2014.

[10] S. Dasgupta. A cost function for similarity-based hierarchical clustering. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2016, pages 118–127, New York, NY, USA, 2016. ACM.

[11] S. Dasgupta and P. M. Long. Performance guarantees for hierarchical clustering. *Journal of Computer and System Sciences*, 70(4):555–569, 2005.

[12] N. R. Devanur, S. A. Khot, R. Saket, and N. K. Vishnoi. Integrality gaps for sparsest cut and minimum linear arrangement problems. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 537–546. ACM, 2006.

[13] J. Eldridge, M. Belkin, and Y. Wang. Beyond hartigan consistency: merge distortion metric for hierarchical clustering. 2015.

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

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

[16] U. Feige and J. R. Lee. An improved approximation ratio for the minimum linear arrangement problem. *Information Processing Letters*, 101(1):26–29, 2007.

[17] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi) cut theorems and their applications. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 698–707. ACM, 1993.

[18] J. A. Hartigan. Statistical theory in clustering. *Journal of Classification*, 2(1):63–76, 1985.

[19] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2009.

[20] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[21] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[22] J. A. Kelner, Y. T. Lee, L. Orecchia, and A. Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 217–226. Society for Industrial and Applied Mathematics, 2014.

[23] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 767–775. ACM, 2002.

[24] R. Krauthgamer, J. S. Naor, and R. Schwartz. Partitioning graphs into balanced components. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 942–949. Society for Industrial and Applied Mathematics, 2009.

[25] T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Foundations of Computer Science, 1988., 29th Annual Symposium on*, pages 422–431. IEEE, 1988.

[26] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM (JACM)*, 46(6):787–832, 1999.

[27] G. Lin, C. Nagarajan, R. Rajaraman, and D. P. Williamson. A general approach for incremental approximation and hierarchical clustering. *SIAM Journal on Computing*, 39(8):3633–3669, 2010.

[28] K. Makarychev, Y. Makarychev, and A. Vijayaraghavan. Bilu-linial stable instances of max cut and minimum multiway cut. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 890–906, Philadelphia, PA, USA, 2014. Society for Industrial and Applied Mathematics.

[29] R. Peng. Approximate undirected maximum flows in o (m polylog (n)) time. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1862–1867. SIAM, 2016.

[30] C. G. Plaxton. Approximation algorithms for hierarchical location problems. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 40–49. ACM, 2003.

[31] P. Raghavendra and D. Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 755–764. ACM, 2010.

[32] P. Raghavendra, D. Steurer, and M. Tulsiani. Reductions between expansion problems. In *Computational Complexity (CCC), 2012 IEEE 27th Annual Conference on*, pages 64–73. IEEE, 2010.

[33] A. Roy and S. Pokutta. Hierarchical clustering via spreading metrics. *NIPS*, 2016.

[34] J. Sherman. Breaking the multicommodity flow barrier for o (vlog n)-approximations to sparsest cut. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 363–372. IEEE, 2009.

[35] J. Sherman. Nearly maximum flows in nearly linear time. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 263–269. IEEE, 2013.

Copyright © by SIAM
Unauthorized reproduction of this article is prohibited