# $O(\sqrt{\log n})$ Approximation Algorithms for Min UnCut, Min 2CNF Deletion, and Directed Cut Problems

Amit Agarwal[*]
aagarwal@cs.princeton.edu

Moses Charikar[†]
moses@cs.princeton.edu

Konstantin Makarychev[*]
kmakaryc@cs.princeton.edu

Yury Makarychev[*]
ymakaryc@cs.princeton.edu

Department of Computer Science
Princeton University
Princeton, NJ 08544

## ABSTRACT

We give $O(\sqrt{\log n})$-approximation algorithms for the MIN UNCUT, MIN 2CNF DELETION, DIRECTED BALANCED SEPARATOR, and DIRECTED SPARSEST CUT problems. The previously best known algorithms give an $O(\log n)$-approximation for MIN UNCUT [9], DIRECTED BALANCED SEPARATOR [17], DIRECTED SPARSEST CUT [17], and an $O(\log n \log\log n)$-approximation for MIN 2CNF DELETION [14].

We also show that the integrality gap of an SDP relaxation of the MINIMUM MULTICUT problem is $\Omega(\log n)$.

## Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms

## General Terms

Algorithms, Theory

## Keywords

Min UnCut, Min 2CNF Deletion, Directed Balanced Separator, Directed Sparsest Cut, Min Multicut

*Supported by NSF grants CCR-0205594, CCR-0237113, DOE award DE-FG02-02ER25594 and a Gordon Wu fellowship.

†Supported by NSF ITR grant CCR-0205594, DOE Early Career Principal Investigator award DE-FG02-02ER25540, NSF CAREER award CCR-0237113, an Alfred P. Sloan Fellowship and a Howard B. Wentz Jr. Junior Faculty Award.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*STOC'05,* May 22-24, 2005, Baltimore, Maryland, USA.
Copyright 2005 ACM 1-58113-960-8/05/0005 ...$5.00.

## 1. INTRODUCTION

In this paper we present $O(\sqrt{\log n})$-approximation algorithms for several important combinatorial problems: the MIN UNCUT, MIN 2CNF DELETION, DIRECTED BALANCED SEPARATOR and DIRECTED SPARSEST CUT problems.

The MIN UNCUT and MIN 2CNF DELETION problems are representatives of the family of *Minimum Constraint Satisfaction Problems (Min CSP).*

DEFINITION 1 (MIN CSP($\mathcal{F}$)). *Consider boolean variables $b_1, \ldots, b_n$ and a set of constraints $C$ from $\mathcal{F}$. The goal is to find an assignment that minimizes the number of unsatisfied constraints.*

In the MIN UNCUT problem each constraint is of the form $b_i \oplus b_j = 0$ or $b_i \oplus b_j = 1$, in the MIN 2CNF DELETION problem each constraint is of the form $b_i \vee b_j$, $\bar{b_i} \vee b_j$ or $\bar{b_i} \vee \bar{b_j}$. The MIN UNCUT problem also has equivalent graph theoretic formulations (see Section 2).

The MIN 2CNF DELETION problem is of particular interest since it is the hardest Min CSP problem that has nontrivial approximation guarantees. Khanna, Sudan, Trevisan and Williamson classified all Min CSP problems [12] and both MIN UNCUT and MIN 2CNF DELETION are complete problems for classes of Min CSP problems in their hierarchy. They showed that either a Min CSP problem is at least as hard to approximate as the NEAREST CODEWORD problem, or the problem can be A-reduced to the MIN 2CNF DELETION problem. In the former case the problem is hard to approximate to within a factor of $\Omega(2^{\log^{1-\varepsilon} n})$ unless every NP language can be decided in deterministic time $n^{poly(\log n)}$ [1]. In the latter case, our algorithm yields an $O(\sqrt{\log n})$-approximation.

In particular, if each constraint depends only on two variables then the problem is reducible to MIN 2CNF DELETION. Though MIN UNCUT is also reducible to MIN 2CNF DELETION, we present a slightly simpler algorithm that deals with MIN UNCUT.

Another application of the MIN 2CNF DELETION problem was recently found by Chlebík and Chlebíková [7]. They showed that every $k$-approximation algorithm for MIN 2CNF DELETION polynomially reduces to a $(2-\frac{2}{k+1})$-approximation algorithm for the MINIMUM VERTEX COVER problem. Combining our result with this reduction we get an algorithm for Vertex Cover with the same approximation factor $2-\Omega(\frac{1}{\sqrt{\log n}})$ as in the best known approximation algorithm for Vertex Cover by Karakostas [11].

The previously best known approximation ratio for Min UnCut is $O(\log n)$ [9], and the best previously known approximation for Min 2CNF Deletion is $O(\log n \log\log n)$ [14]. Both problems are known to be Max SNP-hard [18]. The best known lower bound for Min 2CNF Deletion is $8\sqrt{5}-15 \approx 2.88854$ [7]. Moreover, if the Unique Games Conjecture holds true, then Min 2CNF Deletion cannot be approximated within any constant factor [13].

We also study the Directed Balanced Separator and Directed Sparsest Cut problems. Recently Arora, Rao, and Vazirani [3] presented an $O(\sqrt{\log n})$-pseudo approximation algorithm for the Balanced Separator problem, and an $O(\sqrt{\log n})$-approximation for the Sparsest Cut problem. We extend their results to the directed versions of these problems.

In this paper, we introduce new methods for solving combinatorial problems on directed graphs.

In Sections 2 and 3, we formulate the Min UnCut and Min 2CNF Deletion problems, present semidefinite relaxations for them and sketch the rounding algorithms.

In Section 4, we define a *directed semimetric*, and show how to construct a separation algorithm similar to that of [3] for it. Then we present a recursive algorithm that given a symmetric unit-$\ell_2^2$ representation of a graph $G=(V,E)$ (see Section 4.1 for the definitions) partitions the vertices $V$ into sets $S$ and $T=-S$ such that the cost of the directed cut between $S$ and $T$ (the number of edges going from $S$ to $T$) is at most $O(\sqrt{\log n})$ of the volume of $G$. Applying this algorithm to the solutions of the SDP relaxations for the Min UnCut and Min 2CNF Deletion problems we get $O(\sqrt{\log n})$-approximations.

In Section 5 using this semimetric instead of the $\ell_2^2$ distance in the algorithms by Arora, Rao and Vazirani [3] we achieve an $O(\sqrt{\log n})$-pseudo approximation for the Directed Balanced Separator problem, and an $O(\sqrt{\log n})$-approximation for the Directed Sparsest Cut problem.

Finally, in Section 6 we show that the integrality gap of a rather strong SDP relaxation (with triangle inequalities) for the Minimum Multicut problem is $\Omega(\log n)$. Thus methods developed in this paper cannot be directly applied to this problem. This is quite interesting since previously the Min UnCut problem was solved by a reduction to the Minimum Multicut problem.

Recently, Charikar, Karloff and Rao [5] have extended our methods for directed semimetrics to design approximation algorithms for directed vertex ordering problems. They obtain $O(\sqrt{\log n}\log\log n)$ approximations for Minimum Linear Arrangement, Minimum Storage-Time Product, and Minimum Containing Interval Graph.

Finally, we mention that the recent results of Arora, Lee and Naor [2] on embedding negative type metrics into $\ell_2$ do not yield algorithms for the problems we consider in this paper.

## 2. APPROXIMATING MIN UNCUT

Definition 2 (Min UnCut Problem). *Consider boolean variables $b_1,\dots,b_n$ and a set of constraints of the form $b_i \oplus b_j = 0$ and $b_i \oplus b_j = 1$. The goal is to minimize the number of unsatisfied constraints.*

Remark 2.1. *There are two other equivalent forms of the Min UnCut problem that are commonly used.*

*The first one is as follows: Given a graph $G=(V,E)$, find a minimum set of edges $M$ such that $G-M \equiv (V, E\setminus M)$ is a bipartite graph.*

*The second definition explains the name of the problem: Given a graph $G=(V,E)$, find a cut that minimizes the number of uncut edges i.e. the number of edges within each part.*

We will prove the following result.

Theorem 2.1. *There is a randomized polynomial-time algorithm for finding an $O(\sqrt{\log n})$ approximation for the Min UnCut problem.*

We will reduce Min UnCut to an alternate problem which will be convenient for our purposes. We start with a definition and then explain the reduction.

Definition 3. *Consider a directed or undirected graph $G=(V,E)$ on the set $V=\{-n,\dots,-1\}\cup\{1,\dots,n\}$. We say that a set of vertices $M\subset V$ is symmetric if $M=-M$, where $-M\equiv\{-i : i\in M\}$. The set of edges $E$ is symmetric if $(i,j)\in E \leftrightarrow (-j,-i)\in E$. The graph $G$ is symmetric if the set of its vertices $V$ and the set of its edges $E$ are symmetric.*

Given an instance of Min UnCut, we first add new boolean variables $b_{-1},\dots,b_{-n}$ and set $b_{-1}=\bar{b}_1,\dots,b_{-n}=\bar{b}_n$, i.e. $b_{-i}$ is a shortcut for $\bar{b}_i$. Then we replace all constraints of the form $b_i\oplus b_j=1$ with two equivalent constraints $b_{-i}\leftrightarrow b_j$ and $b_{-j}\leftrightarrow b_i$; we replace $b_i\oplus b_j=0$ with $b_i\leftrightarrow b_j$ and $b_{-j}\leftrightarrow b_{-i}$. The number of unsatisfied new constraints is exactly twice the number of unsatisfied old constraints. We now consider the graph $G=(V,E)$, where $V=\{-n,\dots,-1\}\cup\{1,\dots,n\}$ and $(i,j)\in E$ *iff* there is a constraint $b_i\leftrightarrow b_j$.

We claim that Min UnCut is equivalent to the problem of finding a minimum symmetric cut $(S,\bar{S}=-S)$ in $G$. The symmetric cut gives us an assignment of truth values to variables in the original instance – one part corresponds to the variables set to true, and the other corresponds to those set to false. Note that the cut edges in the new problem correspond to constraints that are unsatisfied in the original instance.

We now write an SDP (vector program) relaxation for the new problem:

$$\min \frac{1}{4}\sum_{(i,j)\in E}|v_i-v_j|^2$$

$$\begin{aligned}
\text{s.t.}\quad & |v_i|^2 = 1 && \forall i\in V\\
& |v_i-v_j|^2 \le |v_i-v_k|^2+|v_k-v_j|^2 && \forall i,j,k\in V\\
& v_i = -v_{-i} && \forall i\in V
\end{aligned}$$

The last constraint will ensure that the cut is symmetric. The idea of using antipodal vectors in an SDP relaxation was used before for an SDP relaxation of the Vertex Cover problem by Karakostas [11].

This SDP is indeed a relaxation. Every assignment of the boolean variables corresponds to a feasible set of vectors:

$$v_i = \begin{cases} v_0 & \text{, if } b_i = 1;\\ -v_0 & \text{, if } b_i = 0;\end{cases}$$

where $v_0$ is a fixed unit vector. The objective function is equal to the number of unsatisfied constraints.

Define the volume of a set $M \subset V$ to be

$$\mathrm{vol}(M) = \sum_{\substack{(i,j)\in E\\ i,j\in M}} |v_i - v_j|^2.$$

In other words, the volume of a set is equal the contribution of the set to the SDP value multiplied by four. Similarly the volume of an edge $(i, j)$ is $|v_i - v_j|^2$.

Definition 4. *A unit-$\ell_2^2$ representation of a graph $G$ is an assignment of vectors $v_i$ to each vertex $i$ such that*

1. *All vectors $v_i$ lie on the unit sphere:*

$$\forall i \in V\ |v_i| = 1.$$

2. *The $\ell_2^2$ triangle inequality holds:*

$$\forall i, j, k \in V\ |v_i - v_j|^2 \le |v_i - v_k|^2 + |v_k - v_j|^2.$$

*Let the set of vertices of the graph $G$ be $\{-n, \ldots, -1\} \cup \{1, \ldots, n\}$, then a unit-$\ell_2^2$ representation is symmetric if the vectors are symmetric w.r.t. the origin:*

$$\forall i \in V\ v_i = -v_{-i}.$$

*A unit-$\ell_2^2$ representation is c-spread if*

$$\sum_{i<j} |v_i - v_j|^2 \ge 4c(1 - c) \cdot (\#vertices)^2.$$

Clearly, every feasible set of vectors for the SDP relaxation is a symmetric unit-$\ell_2^2$ representation of the graph $G$.

We now informally sketch the algorithm for partitioning the graph. First we solve the SDP relaxation and get a unit-$\ell_2^2$ representation of the graph. Using the ARV separation algorithm [3] we find symmetric sets $S^*$ and $T^*$ which are $\Omega(1/\sqrt{\log n})$-separated w.r.t the squared Euclidean distance ($\ell_2^2$). Then we take their neighborhoods $S_1 \supset S^*$ and $T_1 \supset T^*$ such that the number of outgoing edges from $S_1$ is at most $O(\sqrt{\log n})$ times the volume of $G$. We apply the same procedure to the remaining part $R_1 = V \setminus (S_1 \cup T_1)$ and get sets $S_2$ and $T_2 = -S_2$ *etc.* Finally we set $S = \cup_i S_i$, $T = \cup_i T_i$ and return the cut $(S, T)$. Since all sets $S_i$ and $T_i$ are symmetric, the cut is also symmetric. The size of the cut is less than or equal to the sum of the number of outgoing edges from $S_1$, $S_2$, *etc.* Which is bounded by

$$O(\sqrt{\log n}) \cdot (\mathrm{vol}(V) + \mathrm{vol}(R_1) + \mathrm{vol}(R_2) + \ldots).$$

In order this sum to be $O(\sqrt{\log n}\,\mathrm{vol}(G))$, it suffices that the volumes of $R_i$ decrease geometrically. In other words, the sets $S_i$ and $T_i$ should contain a constant fraction of the volume of $R_{i-1}$ at each iteration of the algorithm. To guarantee this we assign to each vertex weight proportional to the volume of the outgoing edges from this vertex. We then use the weighted version of the separation algorithm to get symmetric sets $S^*$ and $T^*$ which contain a constant fraction of the volume.

We will give a detailed explanation and analysis of the algorithm in Section 4.

# 3. APPROXIMATING MIN 2CNF DELETION

Definition 5 (Min 2CNF Deletion Problem). *Consider boolean variables $b_1, \ldots, b_n$ and a set of constraints of the form $b_i \vee b_j$, $\bar{b_i} \vee b_j$ and $\bar{b_i} \vee \bar{b_j}$. The goal is to minimize the number of unsatisfied constraints.*

We will prove the following result.

Theorem 3.1. *There is a randomized polynomial-time algorithm for finding an $O(\sqrt{\log n})$ approximation for the* Min 2CNF Deletion *problem.*

Similarly to the Min UnCut problem, we introduce new variables $b_{-1}, \ldots, b_{-n}$; set $b_{-1} = \bar{b_1}, \ldots, b_{-n} = \bar{b_n}$. Then we replace each constraint $b_i \vee b_j$ with two equivalent constraints $b_{-i} \to b_j$ and $b_{-j} \to b_i$. We now want to minimize the number of unsatisfied constraints of the new form. We consider the graph $G = (V, E)$, where $V = \{-n, \ldots, -1\} \cup \{1, \ldots, n\}$ and $(i, j) \in E$ *iff* there is a constraint $b_i \to b_j$. It is easy to see that the graph is symmetric.

We write an SDP relaxation for Min 2CNF Deletion:

$$\begin{aligned} \min\ & \frac{1}{8} \sum_{(i,j)\in E} |v_i - v_j|^2 - |v_0 - v_i|^2 + |v_0 - v_j|^2 \\ \text{s.t}\ & |v_i|^2 = 1 && \forall i \in V \cup \{0\} \\ & |v_i - v_j|^2 \le |v_i - v_k|^2 + |v_k - v_j|^2 && \forall i, j, k \in V \cup \{0\} \\ & v_i = -v_{-i} && \forall i \in V \end{aligned}$$

where $v_i$ ($i \in V$) corresponds to the boolean variable $b_i$; $v_0$ corresponds to true; and $-v_0$ corresponds to false. Note that this is indeed a valid relaxation. For every constraint $b_i \to b_j$, we have the term $\frac{1}{8}(|v_i - v_j|^2 - |v_0 - v_i|^2 + |v_0 - v_j|^2)$ in the objective function. If $v_i = v_j$ or if $v_i = -v_0, v_j = v_0$ (i.e. the constraint is satisfied), the value of this expression is 0. On the other hand, the value is 1 if $v_i = v_0, v_j = -v_0$ (i.e. the constraint is not satisfied).

An SDP solution is a symmetric unit-$\ell_2^2$ representation. Using techniques described in Section 4 we partition the set of vertices into two sets $S$ and $T = -S$ such that the cost of the directed cut $(S, -S)$ is at most $O(\sqrt{\log n})$ of the SDP value. We set $b_i$ to be true, if $i \in S$ and false otherwise. Then each unsatisfied constraint $b_i \to b_j$ corresponds to an edge from $S$ to $-S$. Therefore, the number of unsatisfied constraints is $O(\sqrt{\log n})$ of the SDP value.

# 4. TECHNICAL DETAILS

In this section, we will describe a general framework for the Min UnCut and Min 2CNF Deletion problems. Then we prove weighted separation theorems for undirected and directed cases. Finally we present a partitioning algorithm for finding a small directed symmetric cut.

## 4.1 Definitions

Definition 6. *Let $G = (V, E)$ be a directed graph; and $S$ be a subset of its vertices. We denote the set of edges outgoing from $S$ by $\delta^{out}(S)$; the set of edges incoming to $S$ by $\delta^{in}(S)$. $\delta^{out}_M(S)$ [$\delta^{in}_M(S)$] denotes the number of edges outgoing from [incoming to] $S$ in the subgraph $G[M]$ of $G$ induced by a vertex set $M$.*

DEFINITION 7. *Let $G = (V, E)$ be a directed graph. A* directed semimetric[1] *on $G$ is a function $d : V \times V \to \mathbb{R}^+ \cup \{0\}$ such that*

1. *$\forall i \in V\ d(i,i) = 0$.*
2. *The triangle inequality holds:*

$$\forall i,j,k \in V\ d(i,j) + d(j,k) \geq d(i,k).$$

*We say that the directed semimetric is* symmetric *if*

$$\forall i,j \in V\ d(i,j) = d(-j,-i).$$

*We define the distance between sets and points in the natural way:*

- $d(S,T) = \min_{i\in S; j \in T} d(i,j)$;
- $d(i,S) = d(\{i\}, S)$;
- $d(S,i) = d(S, \{i\})$;

DEFINITION 8. *Define the volume of $M \subset V$ w.r.t. a directed semimetric $d$ as follows:*

$$\mathrm{vol}_d(M) = \sum_{\substack{(i,j)\in E\\ i,j\in M}} d(i,j).$$

DEFINITION 9. *Let $d$ be a directed semimetric. Let $S$ and $T$ be two sets of vertices. $S$ and $T$ are $\Delta$-separated w.r.t. $d$ if $d(S,T) \geq \Delta$.*

DEFINITION 10. *Let $v_i$ be a symmetric unit-$\ell_2^2$ representation of a graph $G$. And let $v_0$ be a fixed unit vector, such that all vectors $v$ (including $v_0$) satisfy the $\ell_2^2$ triangle inequality.*

*We define two functions*

- $d_1(i,j) = |v_i - v_j|^2$;
- $d_2(i,j) = |v_i - v_j|^2 - |v_0 - v_i|^2 + |v_0 - v_j|^2$.

OBSERVATION 4.1. *$d_1$ and $d_2$ are symmetric semimetrics.*

PROOF. The triangle inequality for $d_1$ and the fact that $d_2$ is nonnegative trivially follow from the triangle inequality constraint from the definition of the unit-$\ell_2^2$ representation. The triangle inequality for $d_2$ is derived as follows:

$$\begin{aligned}
d_2(i,k) \quad + \quad & d_2(k,j) = |v_i - v_k|^2 - |v_0 - v_i|^2 \\
+ \quad & |v_0 - v_k|^2 + |v_k - v_j|^2 - |v_0 - v_k|^2 + |v_0 - v_j|^2 \\
= \quad & |v_i - v_k|^2 + |v_k - v_j|^2 - |v_0 - v_i|^2 + |v_0 - v_j|^2 \\
\geq \quad & |v_i - v_j|^2 - |v_0 - v_i|^2 + |v_0 - v_j|^2 = d_2(i,j)
\end{aligned}$$

□

REMARK 4.1. *We gave the definitions above for directed graphs, since, generally speaking, the volume of an edge depends on its orientation. However, when the metric $d$ is "undirected", i.e. $d(x,y) = d(y,x)$, these definitions also apply to undirected graphs. We think of an undirected graph as a directed graph where each edge is oriented in both directions. In particular, when we use the metric $d_1$ (to analyze the* MIN UNCUT *problem) the graph $G$ will be undirected.*

Note that the SDP value for MIN UNCUT is equal to the volume of the graph w.r.t. $d_1(i,j)$ multiplied by 4; the SDP value for MIN 2CNF DELETION is equal to the volume of the graph w.r.t. $d_2(i,j)$ multiplied by 8.

[1]Directed semimetrics are sometimes called quasi-semimetrics.

## 4.2 Separation Algorithm

In this section, we will describe an algorithm which given a symmetric unit-$\ell_2^2$ representation of a graph $G$ finds $\Delta = \Omega(1/\sqrt{\log n})$-separated w.r.t. $d_1$ $[d_2]$ sets $S$ and $T = -S$ such that the set $S$ contains a constant fraction of the total volume of the graph w.r.t. $d_1$ $[d_2]$.

The following result by Arora, Rao, Vazirani [3] plays a central role in our paper (see also Lee's analysis of the algorithm in [16]).

THEOREM 4.2 (ARV ALGORITHM). *For every $c > 0$, every $c$-spread unit-$\ell_2^2$ representation with $n$ points contains $\Delta$-separated w.r.t. the $\ell_2^2$ distance subsets $S$, $T$ of size $\Omega(n)$, where $\Delta = \Omega(1/\sqrt{\log n})$. Furthermore, there is a randomized polynomial-time algorithm for finding these subsets $S$, $T$.*

OBSERVATION 4.3. *Every symmetric unit-$\ell_2^2$ representation is $1/3$-spread (for $n \geq 9$).*

PROOF.

$$\begin{aligned}
&\sum_{i<j} |v_i - v_j|^2 \\
&\quad = \sum_{i>0}\sum_{j>i} \left(|v_i - v_j|^2 + |v_i - v_{-j}|^2\right. \\
&\qquad\qquad \left. + |v_{-i} - v_j|^2 + |v_{-i} - v_{-j}|^2\right) \\
&\quad = \sum_{i>0}\sum_{j>i} 8 = 4(n-1)n \geq 4 \cdot \frac{1}{3} \cdot \frac{2}{3} \cdot 4n^2.
\end{aligned}$$

□

Since every symmetric unit-$\ell_2^2$ representation is a $1/3$-spread the theorem is also applicable to symmetric unit-$\ell_2^2$ representations. Moreover, we may assume that the algorithm returns sets $S$ and $T$ that are reflections of each other about the origin: $S = -T$. Indeed, the first step of the algorithm partitions $V$ into symmetric sets $S'$ and $T'$: $S' = -T'$. At the deletion step we have some freedom in choosing matchings. We should always choose symmetric matchings, that is if $(i,j)$ belongs to the matching, $(-i,-j)$ should also belong to the matching.

COROLLARY 4.4. *Any symmetric unit-$\ell_2^2$ representation with $2n$ points contains $\Delta$-separated w.r.t. the $\ell_2^2$ distance subsets $S$, and $T = -S$ of size $\Omega(n)$, where $\Delta = \Omega(1/\sqrt{\log n})$. Furthermore, there is a randomized polynomial-time algorithm for finding these subsets $S$, $T$.*

Now we are ready to present an algorithm that given a symmetric unit-$\ell_2^2$ representation and weights for each vertex finds $\Omega(1/\sqrt{\log n})$-separated w.r.t. $\ell_2^2$ sets $S$ and $T = -S$ with a constant fraction of the total weight (the details are below). We use the algorithm from Corollary 4.4 as a subroutine. The algorithm is based on the approach of Chawla, Gupta and Räcke [6].

ALGORITHM 1 (WEIGHTED SEPARATION).
*Input: a directed graph $G$; a symmetric unit-$\ell_2^2$ representation of $G$; a symmetric subset of vertices $M$; weights $w_i$ of all vertices $i \in M$ (where $w_i = w_{-i}$);*
*Output: sets $S$ and $-S$, such that*
*i. $S$ and $-S$ are $\Delta = \Omega(1/\sqrt{\log n})$-separated w.r.t. the $\ell_2^2$ distance;*
*ii. $\sum_{i\in M\setminus(S\cup -S)} w_i \leq c_1 W$, where $W = \sum_{i\in M} w_i$ and $c_1$ ($c_1 < 1$) is a fixed constant.*

1. For all $i \in M$ let

$$m_i = \left\lfloor w_i \Big/ \frac{W}{n^2} \right\rfloor.$$

2. Duplicate each point $i \in M$ $m_i$ times; assign the same vector $v_i$ to each copy $(i, j)$ of $i$:

$$V_{dup} = \{(i, j) : i \in M, j = 1, \dots, m_i\};$$

$$\forall i' = (i, j) \in V_{dup}\ v_{i'} = v_i.$$

Note that the set $V_{dup}$ is of polynomial size (at most $2n^3$ vertices); and it is also a symmetric unit-$\ell_2^2$ representation.

3. Run the $\Delta$-separation algorithm from Corollary 4.4 on $V_{dup}$. Denote the output sets by $S_{dup}$ and $T_{dup}$.

4. Let $S$ be the set of vertices $i$ such that at least one duplicate $i'$ of $i$ belongs to $S_{dup}$.

5. Return $S$ and $-S$.

ANALYSIS. First note that $T_{dup} = -S_{dup}$. Then if $i \in S$ and $j \in -S$, the distance between $i$ and $j$ is the same as the distance between their duplicates $i' \in S_{dup}$ and $j' \in -S_{dup}$. The sets $S_{dup}$ and $-S_{dup}$ are $\Delta$-separated, so $S$ and $-S$ are also $\Delta$-separated.

Finally we verify that the weight of the set $S$ is a constant fraction of the total weight. By Corollary 4.4, $S_{dup}$ and $T_{dup}$ contain some constant fraction $c_0$ of vertices $V_{dup}$:

$$|S_{dup}| = |T_{dup}| \geq c_0 |V_{dup}|$$

The size of $V_{dup}$ is at least $n^2 - 2n$:

$$|V_{dup}| = \sum_{i \in M} m_i = \sum_{i \in M} \left\lfloor w_i \Big/ \frac{W}{n^2} \right\rfloor \geq \sum_{i \in M} \left( w_i \Big/ \frac{W}{n^2} - 1 \right) \geq n^2 - 2n$$

So the weight of $S$ and $-S$ is

$$\sum_{i \in S} w_i \geq \sum_{i \in S} \frac{W}{n^2} m_i \geq \frac{W}{n^2} |S_{dup}| \geq \frac{W}{n^2} c_0 |V_{dup}| \geq c_0 \frac{(n^2 - 2n) W}{n^2} \geq \frac{c_0}{2} W$$

Thus, $c_1 \leq 1 - c_0 < 1$. □

We now use this algorithm to describe separation algorithms for symmetric semimetrics $d_1(i, j)$ and $d_2(i, j)$.

LEMMA 4.5. *[Separation Algorithm for $d_1$] There exists an algorithm which given a symmetric unit-$\ell_2^2$ representation of a symmetric graph $G = (V, E)$, and a symmetric set $M \subset V$, finds $\Delta = \Omega(1/\sqrt{\log n})$-separated w.r.t. $d_1$ sets $S \subset M$ and $T = -S \subset M$ such that the volume of the remaining set $M \setminus (S \cup -S)$ is less than some constant fraction of the volume of $M$.*

PROOF. We run the weighted separation algorithm with weights $w_i$ (for $i \in M$) equal to the total volume of edges incident to the vertex $i$:

$$w_i = \sum_{\substack{j:(i,j) \in E \\ j \in M}} d_1(i, j).$$

The algorithm produces $\Delta$-separated sets $S$ and $-S$. Note that

$$\mathrm{vol}_{d_1}(M) = \frac{1}{2} w(M) \equiv \sum_{i \in M} w_i$$

$$\mathrm{vol}_{d_1}(M \setminus (S \cup -S)) \leq \frac{1}{2} w(M \setminus (S \cup -S))$$

Therefore,

$$\mathrm{vol}_{d_1}(M \setminus (S \cup -S)) \leq \frac{1}{2} w(M \setminus (S \cup -S)) \leq \frac{1}{2} c_1 w(M) = c_1 \,\mathrm{vol}_{d_1}(M)$$

□

LEMMA 4.6. *[Separation Algorithm for $d_2$] There exists an algorithm which given a symmetric unit-$\ell_2^2$ representation of a symmetric graph $G = (V, E)$, and a symmetric set $M \subset V$, finds $\Delta = \Omega(1/\sqrt{\log n})$-separated w.r.t. $d_2$ sets $S \subset M$ and $T = -S \subset M$ such that the volume of the remaining set $M \setminus (S \cup -S)$ is less than some constant fraction of the volume of $M$.*

PROOF. As in the previous lemma we set

$$w_i = \sum_{\substack{j:(i,j) \in E \\ j \in M}} d_2(i, j) + \sum_{\substack{j:(j,i) \in E \\ j \in M}} d_2(j, i).$$

(the first term is the volume of all outgoing edges, the second term is the volume of all incoming edges)

Note that $w_i = w_{-i}$ since the set of edges is symmetric: if $(i, j) \in E$ then $(-j, -i) \in E$, and since $d_2(i, j) = d_2(-j, -i)$.

The weighted separation algorithm returns $\Delta$-separated w.r.t. $\ell_2^2$ sets $S$ and $-S$. Let

$$S^+ = \{i \in S : \langle v_0, v_i \rangle \ \geq 0\}; \ \ S^- = \{i \in S : \langle v_0, v_i \rangle \ \leq 0\}.$$

Note that $S^+$ and $-S^+$ are $\Delta$-separated w.r.t. $\ell_2^2$: If $i \in S^+$, $j \in -S^+$, then

$$\begin{aligned} d_2(i, j) &= |v_i - v_j|^2 - |v_0 - v_i|^2 + |v_0 - v_j|^2 \\ &= |v_i - v_j|^2 + 2\langle v_0, v_i - v_j \rangle \geq |v_i - v_j|^2 \geq \Delta. \end{aligned}$$

Hence $S^+$ and $-S^+$ are $\Delta$-separated w.r.t. $d_2$. Similarly $-S^-$ and $S^-$ are $\Delta$-separated. Since $S = S^+ \cup S^-$, one of the sets $S^+$ or $S^-$ contains $(1 - c_1)/4$ fraction of the total weight. The output $S$ of the algorithm is the largest of the sets $S_1$ and $S_2$. By the same argument as in the previous lemma, $\mathrm{vol}_{d_2}(M \setminus (S \cup -S)) \leq \frac{1+c_1}{2} \mathrm{vol}_{d_2}(M)$. □

## 4.3 Partitioning

Let $d$ be a directed symmetric semimetric $d_1$ or $d_2$. First we construct an algorithm that takes a symmetric set $M$ and partition it into three disjoint sets $S, R, -S$ with the following properties.

ALGORITHM 2 (FINDING $(S, R, -S)$ PARTITIONING).
*Input: a symmetric directed graph $G$; a symmetric unit-$\ell_2^2$ representation of $G$; $M \subset V$ such that $M = -M$.*
*Output: disjoint subsets $S, R, -S \subset M$ such that*
*i.*

$$|\delta_M^{out}(S)| + |\delta_M^{in}(-S)| = O\big(\frac{\mathrm{vol}_d(M)}{\Delta}\big).$$

*Here we consider only edges of the induced subgraph $G[M]$ of $G$ by the vertex subset $M$.*

*ii. The volume of $R$ is at most a constant fraction of the volume of $M$:*

$$\mathrm{vol}_d(R) \leq c_2 \,\mathrm{vol}_d(M), \text{ where } c_2 < 1 \text{ is a constant}$$

1. Run the separation algorithm for the semimetric $d$. Denote the result by $S^*$ and $-S^*$.

2. Define the set of edges $E_t$ $(t \in (0, \Delta))$ as follows

$$E_t = \{(i,j) \in E : d(S^*, i) \leq t \text{ and } d(S^*, j) \geq t\} \cup \{(i,j) \in E : d(j, -S^*) \leq t \text{ and } d(i, -S^*) \geq t\}.$$

   Note that if $d$ is a metric then $E_t$ is the set of edges intersecting with the boundary of the $t$-neighborhood of $S \cup -S$.

3. Find $t_0 \in (0, \Delta/4)$ which minimize the size of $E_t$.

4. Remove the set of edges $E_{t_0}$ from the graph $G[M]$. Let $S$ be the set of vertices that are reachable from $S^*$ in the remaining graph $G[M] - E_{t_0}$.

5. Let $R = M \setminus (S \cup -S)$.

6. Return $S$, $R$ and $-S$.

Analysis. First, note that for all $i \in S$ $d(S^*, i) \leq \Delta/4$. Since $d(i,j) = d(-j, -i)$, for all $i \in -S$ $\quad d(i, -S^*) \leq \Delta/4$. The sets $S^*$ and $-S^*$ are $\Delta$-separated, so $S$ and $-S$ are $\Delta/2$-separated (here we use the triangle inequality); and thus $S$ and $-S$ are disjoint.

By the definition of $S$ $\delta^{out}_M(S) \subset E_{t_0}$; $\delta^{in}_M(-S) \subset E_{t_0}$. Now, using standard arguments we get

$$\begin{aligned}\mathrm{vol}_d(M) &= \sum_{\substack{(i,j)\in E\\ i,j\in M}} d(i,j) \geq \sum_{\substack{(i,j)\in E\\ i,j\in M\\ d(S^*,j)\geq d(S^*,i)}} d(S^*, j) - d(S^*, i) \\ &= \sum_{\substack{(i,j)\in E\\ i,j\in M\\ d(S^*,j)\geq d(S^*,i)}} \int_{d(S^*,i)}^{d(S^*,j)} dt \geq \int_0^{\Delta/2} |E_t| dt \\ &\geq \int_0^{\Delta/4} |E_t| dt\end{aligned}$$

hence $|E_{t_0}| \leq \frac{4}{\Delta}\mathrm{vol}_d(M)$ and

$$|\delta^{out}_M(S)| + |\delta^{in}_M(S)| \leq \frac{4}{\Delta}\mathrm{vol}_d(M) = O\big(\frac{\mathrm{vol}_d(M)}{\Delta}\big).$$

The set $R$ is a subset of $M \setminus (S^* \cup -S^*)$, so

$$\mathrm{vol}_d(R) \leq \mathrm{vol}_d(M \setminus (S^* \cup -S^*)) \leq c_2 \,\mathrm{vol}(M),$$

where $c_2$ is the constant guaranteed by Lemma 4.5 and Lemma 4.6. □

Applying this algorithm recursively, we get an algorithm for finding a symmetric directed cut.

Algorithm 3 (Finding Symmetric Directed Cut).

*Input: a directed symmetric graph $G$; a symmetric unit-$\ell_2^2$ representation of $G$.*

*Output: a symmetric directed cut $(S, -S)$. The cost of the cut is at most $O(1/\Delta) \cdot \mathrm{vol}_d(V)$.*

1. Set $i = 0$. Set $R_0 = V$.

2. **while** $R_i$ is not empty

   (a) Find $(S, R, -S)$ partitioning of $R_i$.

   (b) Let $S_{i+1} = S$. Let $R_{i+1} = R$. Let $i = i + 1$.

3. Return $S = S_1 \cup \cdots \cup S_i$ and $-S$.

Analysis. Clearly, the algorithm returns a symmetric cut. The cost of the directed cut $(S, -S)$ is less than or equal to

$$\begin{aligned}&|\delta^{out}(S_1)| + |\delta^{in}(-S_1)| + |\delta^{out}_{R_1}(S_2)| + |\delta^{in}_{R_1}(-S_2)| + \dots \\ &\qquad = O(\sqrt{\log n}) \cdot (\mathrm{vol}(V) + \mathrm{vol}(R_1) + \dots)\end{aligned}$$

The key observation is that the volume of $R_i$ decreases geometrically, so the cost of the cut is $O(1/\Delta) \cdot \mathrm{vol}_d(V)$. □

This finishes the proofs of Theorems 2.1 and 3.1.

# 5. APPROXIMATING DIRECTED BALANCED SEPARATOR AND DIRECTED SPARSEST CUT

In this section, we present approximation algorithms for Directed Balanced Separator and Directed Sparsest Cut. The algorithms are very similar to their undirected counterparts from [3] except that they use the directed semimetric $d(i,j) = |v_i - v_j|^2 - |v_0 - v_i|^2 + |v_0 - v_j|^2$ instead of the $\ell_2^2$ distance used by the authors of [3].

Definition 11. *Let $G = (V, E)$ be a directed graph. The directed edge expansion of a cut $(S, \bar{S})$ is $|\delta^{out}(S)| / \min(|S|, |\bar{S}|)$.*

*The minimum directed sparsest cut (with uniform demands) is the cut with minimum directed edge expansion. A $c$-balanced cut is a cut $(S, \bar{S})$ s.t. $|S| \geq c|V|$, and $|\bar{S}| \geq c|V|$. Finally, minimum directed $c$-balanced separator is the $c$-balanced cut with minimum directed edge expansion.*

We consider the following SDP relaxation of the Directed $c$-Balanced Separator problem:

$$\min \frac{1}{8} \sum_{(i,j)\in E} d(i,j)$$

s.t.

$$\begin{aligned}|v_i|^2 &= 1 && \forall i \in V\\ |v_k - v_i|^2 &\leq |v_j - v_i|^2 + |v_k - v_j|^2 && \forall i,j,k \in V \cup \{0\}\\ \sum_{i<j} |v_i - v_j|^2 &\geq 4c(1-c)n^2\end{aligned}$$

This is an SDP relaxation, since for each cut $(S, \bar{S})$ the solution $\{v_i\}_i$ defined by $v_i = v_0 = e$ for $i \in S$; and $v_i = -e$ for $i \notin S$ has the value that does not exceed the edge expansion of the cut scaled by $cn$ (here, $e$ is an arbitrary unit vector). The SDP relaxation of the Directed Sparsest Cut problem is:

$$\min \frac{1}{8} \sum_{(i,j)\in E} d(i,j)$$

s.t.

$$\begin{aligned}|v_k - v_i|^2 &\leq |v_j - v_i|^2 + |v_k - v_j|^2 && \forall i,j,k \in V \cup \{0\}\\ \sum_{i<j} |v_i - v_j|^2 &= 1\end{aligned}$$

Similarly, the solution defined by $v_i = v_0$ *for* $i \in S$; and $v_i = -v_0$ *for* $i \notin S$ (where $v_0$ is a vector of length $\frac{2}{(n-1)n}$) corresponds to the cut $(S, \bar{S})$ (with scaling factor $n$).

ALGORITHM 4.
*Solving the Directed c-Balanced Separator Problem.*
*Input: a directed graph G;*
*Output: a $c'$-balanced cut that approximates minimum directed c-balanced separator within a factor of $O(\sqrt{\log n})$ (where $c'$ is 2 times smaller than that in the case of undirected cut).*

1. Solve the SDP relaxation for DIRECTED $c$-BALANCED SEPARATOR.
2. Apply the ARV separation algorithm (see Theorem 4.2) to find $\Delta$-separated (w.r.t. the $\ell_2^2$ distance) sets $S$ and $T$ s.t. each of them contains at least $2c'$ fraction of vertices.
3. Find radius $r$ s.t. at least half of the vectors corresponding to vertices from $S$ lie inside the ball of radius $r$ with center at the point $v_0$, and at least half of the vectors lie outside the ball (we count points on the boundary of the ball as lying inside as well as outside the ball).
4. Let $S^+ = \{i \in S : |v_0 - v_i|^2 \le r^2\}$. Let $S^- = \{i \in S : |v_0 - v_i|^2 \ge r^2\}$.
5. Let $T^+ = \{i \in T : |v_0 - v_i|^2 \le r^2\}$. Let $T^- = \{i \in T : |v_0 - v_i|^2 \ge r^2\}$.
6. If $|T^+| \ge |T^-|$ then $S^* = T^+$; $T^* = S^-$; else $S^* = S^+$; $T^* = T^-$.
7. Find the minimum cut $(A, \bar{A})$ between $S^*$ and $T^*$. Output $(A, \bar{A})$.

ANALYSIS. First of all, notice that the $c$-spreading constraint is one of the constraints in the SDP relaxation of the DIRECTED $c$-BALANCED SEPARATOR problem. Therefore, we can apply the separation algorithm at the first step.

Now, notice that the sets $S^*$ and $T^*$ are $\Delta$-separated w.r.t. the distance $d$: Indeed, let us say $i \in S^* = T^+$, and $j \in T^* = S^-$. Then $d(i,j) = |v_i - v_j|^2 - |v_i - v_0|^2 + |v_j - v_0|^2 \ge |v_i - v_j|^2 - r^2 + r^2 \ge \Delta$. The case $S^* = S^+$, and $T^* = T^-$ is similar. Standard arguments (see Algorithm 2) show that the minimum cut between $S^*$ and $T^*$ costs at most $O(\sqrt{\log n} \cdot SDP)$.

Finally, let us show that the cut $(A, \bar{A})$ is $c'$-balanced. By the construction both $S^+$, and $T^-$ contain at least $|S|/2 \ge c'n$ vertices. Since $T = T^+ \cup T^-$, the larger of the sets $T^+$ and $T^-$ also contains at least $|S|/2 \ge c'n$ vertices. Therefore, both sets $A \supset S^*$ and $\bar{A} \supset T^*$ contain at least $|S|/2 \ge c'n$ vertices. □

Now, let us consider the SDP for the DIRECTED SPARSEST CUT problem. In Appendix A we prove a directed version of Lemma 14 from [3] (the proof is almost identical to the proof for the undirected case):

LEMMA 5.1. *There is a polynomial-time algorithm for the following task. Given any feasible SDP solution with $\beta = \sum_{(i,j)\in E} d(i,j)$, and a vertex $k$ such that the ball of squared-radius $1/(8n^2)$ around $v_k$ contains at least $n/2$ vectors (other than $v_0$), the algorithm finds a cut $(S, \bar{S})$ with directed expansion at most $O(\beta n)$.*

The lemma shows how to find an approximation for the DIRECTED SPARSEST CUT if the hypothesis holds true for some $k$. Otherwise, we scale all vectors by $2\sqrt{2}n$. Now, $\sum_{k\in V}\sum_{i\in V} |v_k - v_i|^2 = 16n^2$. Therefore, for some vertex $k$, $\sum_{i\in V} |v_k - v_i|^2 \ge 16n$. This implies that at least 9/10 fraction of vectors lie inside the ball of radius 160 around $v_k$. And since the hypothesis of lemma holds true for $k$, at most half of the vectors lie inside the ball of radius 1. In other words, a constant fraction of all vertices lie in a spherical annulus of inner radius 1 and outer radius 160. The authors of [3] note that their algorithm works for such set of vertices (with parameter $c'$ equal to some constant). So we can apply Algorithm 4 to vertices in the spherical annulus and vector $v_0$ (note that the algorithm does not require that the vector $v_0$ is a unit vector). It produces a cut with directed edge expansion $O(\sqrt{\log n} \cdot n \cdot SDP)$.

# 6. INTEGRALITY GAP FOR MIN MULTICUT

In this section, we show that the integrality gap of an SDP relaxation of the MINIMUM MULTICUT problem is $\Omega(\log n)$. First, we show that the construction by B. Yu, J. Cheriyan, and P. E. Haxell [19] used in their proof of the integrality gap for an LP relaxation of this problem also yields the integrality gap of $\Omega(\log n)$ for a strong SDP relaxation. For completeness we give our (somewhat shorter) analysis of this construction. Then we show how to modify the construction so that it satisfies additional constraints.

DEFINITION 12. *Consider a graph $G = (V, E)$ and $m$ source–terminal pairs $(s_i, t_i)$ $(1 \le i \le m)$. A multicut $S$ is a subset of edges whose deletion separates each source $s_i$ from the correspondent terminal $t_i$. The cost of the multicut is the number of edges in $S$. The Minimum Multicut problem is to find a multicut of minimal cost.*

Consider the following SDP: min $\frac{1}{2}\sum_{\substack{(i,j)\in E \\ i<j}} |v_i - v_j|^2$, subject to: 1) $|v_i| = 1$, for every vertex $i$; 2) $\langle v_i, v_j\rangle = 0$, for every source–terminal pair $(i,j)$; 3) $|v_i - v_j|^2 + |v_j - v_k|^2 \ge |v_i - v_k|^2$ for all vertices $i$, $j$ and $k$; 4) the metric $d(i,j) = |v_i - v_j|^2$ is embeddable in $\ell_1$.

This program is an SDP relaxation of the MINIMUM MULTICUT problem: For a multicut $S$, we assign the same unit vector to $v_i$ and $v_j$ if $i$ and $j$ are in the same connected component of $G - S$; and let us assign orthogonal unit vectors otherwise. Note that we cannot efficiently compute the value of this SDP relaxation due to the last set of constraints. Usually, one would use weaker constraints instead to ensure polynomial time solvability. However, the point is that even with these very strong constraints, the SDP still has a large integrality gap.

THEOREM 6.1. *The integrality gap of this relaxation is $\Omega(\log(n))$.*

PROOF. Let us take $d = 4k$, and $n = 2^d$. Consider the graph $G$ on the vertices of the hypercube $\{-1/\sqrt{d}, 1/\sqrt{d}\}^d \subset \mathbb{R}^d$; the edges of the graph are the edges of the hypercube. We want to separate every pair of orthogonal vectors. Clearly, the vertices of the hypercube form a feasible solution of the SDP program. The number of edges in the hypercube is $2^{d-1}d$; the squared length of each edge is $4/d$. Thus the value of the SDP is $O(2^d)$.

Now let us lower bound the cost of the minimal multicut. We use the strong version of Larman and Rogers conjecture [15, Conjecture 2] proved by Frankl and Rödl [8, Theorem 1.11].

THEOREM 6.2 (FRANKL AND RÖDL). *Given $r$, $d = 4k$, $k \geq r \geq 2$, there exists a positive constant $\varepsilon = \varepsilon(r)$ so that in any set of more than $(2 - \varepsilon(r))^d$ $(\pm 1)$-vectors there are $r$ pairwise orthogonal vectors.*

Let $S$ be a multicut. Denote the connected components of $G - S$ by $C_1, \dots, C_l$. Since $S$ separates all pairs of orthogonal vectors, each set $C_i$ does not contain two orthogonal vectors. Applying Theorem 6.2 (with $r = 2$) to $C_i$ (scaled by a factor of $\sqrt{d}$) we get that $|C_i| \leq (2 - \varepsilon)^d$.

DEFINITION 13. *Let $T \subset V$. Denote the set of edges from $T$ to $V \setminus T$ by $\delta(T)$.*

LEMMA 6.3 (ISOPERIMETRIC INEQUALITY [4, 10]). *Let $T \subset V$, then $|\delta(T)| \geq |T|(d - \log_2 |T|)$.*

Applying the isoperimetric inequality to a component $C_i$, we get $|\delta(C_i)| \geq |C_i|(d - \log_2((2-\varepsilon)^d)) = d|C_i|\varepsilon'$, where $\varepsilon' = 1 - \log_2(2 - \varepsilon) > 0$ is a constant. Summing up $|\delta(C_i)|$ over all connected components, we get $2|S| = \sum_{i=1}^{l} |\delta(C_i)| \geq \varepsilon' d \sum_{i=1}^{l} |C_i| = \varepsilon' dn$. So the cost of the multicut is $\Omega(d) = \Omega(\log n)$ times more than the value of the SDP. This concludes the proof. □

OBSERVATION 6.4. *We can add an additional constraint that $\langle v_i, v_j \rangle \geq 0$ (for all $i$, $j$), which, in particular, ensures that the diameter of the set $\{v_i\}$ does not exceed the distance between vertices in a source–terminal pair. The integrality gap of this SDP is still $\Omega(\log n)$.*

PROOF. Consider the same example as in Theorem 6.1. Vectors $u_i = v_i \otimes v_i$ form a feasible SDP solution. Indeed, $|u_i|^2 = |v_i|^2 \cdot |v_i|^2 = 1$; for every source–terminal pair $(i, j)$ $\langle u_i, u_j \rangle = \langle v_i, v_j \rangle^2 = 0$. All points $v_i \otimes v_i$ belong to the $d^2$ dimensional hypercube $\left\{\sum_{i,j} \epsilon_{ij} e_i \otimes e_j : \epsilon_{ij} = \pm\frac{1}{d}\right\}$, therefore the distance function $d(i, j) = |v_i - v_j|^2$ is a metric embeddable in $\ell_1$. Finally, $\langle u_i, u_j \rangle = \langle v_i, v_j \rangle^2 \geq 0$.

Now let us compute the value of this solution. The cost of an edge $(i, j)$ is $|u_i - u_j|^2 = |v_i \otimes v_i - v_j \otimes v_j|^2 = 2(1 - \langle v_i, v_j \rangle^2) = 2(1 - (\frac{d-2}{d})^2) = O(\frac{1}{d})$. Hence the value of the SDP is $O(2^d)$, *i.e* the integrality gap is $\Omega(\log n)$. □

# 7. REFERENCES


[1] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997.

[2] S. Arora, J. R. Lee, and A. Naor. Euclidean distortion and the sparsest cut. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, 2005.

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

[4] B. Bollobás. *Combinatorics: Set Systems, Hypergraphs, Families of Vectors and Probabilistic Combinatorics*, pages 122–130. 1986.

[5] M. Charikar, H. Karloff, and S. Rao. Improved approximation algorithms for vertex ordering problems. In *manuscript*, 2004.

[6] S. Chawla, A. Gupta, and H. Räcke. Approximations for generalized sparsest cut and embeddings of $l_2$ into $l_1$. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2005.

[7] M. Chlebík and J. Chlebíková. On approximation hardness of the minimum 2sat-deletion problem. In *Proceedings of 29th International Symposium on Mathematical Foundations of Computer Science*, pages 263–273, 2004.

[8] P. Frankl and V. Rödl. Forbidden intersections. *Trans. Amer. Math. Soc.*, 300:259–286, 1987.

[9] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC)*, pages 698–707, 1993.

[10] S. Hart. A note on the edges of the n-cube. *Discrete Mathematics*, 14:157–163, 1976.

[11] G. Karakostas. A better approximation ratio for the vertex cover problem. In *Electronic Colloquium on Computational Complexity Report TR04-084*, 2004.

[12] S. Khanna, M. Sudan, L. Trevisan, and D. P. Williamson. The approximability of constraint satisfaction problems. *SIAM Journal on Computing*, 30(6):1863–1920, 2001.

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

[14] P. Klein, S. Plotkin, S. Rao, and E. Tardos. Approximation algorithms for steiner and directed multicuts. *Journal of Algorithms*, 22:241–269, 1997.

[15] D. G. Larman and C. A. Rogers. Realization of distances within sets in euclidean space. *Mathematika*, 19:1–24, 1972.

[16] J. R. Lee. On distance scales, embeddings, and efficient relaxations of the cut cone. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2005.

[17] 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.

[18] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *J. Computer and System Sciences*, 43:425–440, 1991.

[19] B. Yu, J. Cheriyan, and P. E. Haxell. Hypercubes and multicommodity flows. *SIAM Journal on Discrete Mathematics*, 10(2):190–200, 1997.


# APPENDIX

# A. PROOF OF LEMMA 5.1

LEMMA 5.1. *There is a polynomial-time algorithm for the following task. Given any feasible SDP solution with $\beta = \sum_{(i,j) \in E} d(i, j)$, and a vertex $k$ such that the ball of squared-radius $1/(8n^2)$ around $v_k$ contains at least $n/2$*

*vectors (other than $v_0$) the algorithm finds a cut $(S, \bar{S})$ with directed expansion at most $O(\beta n)$.*

PROOF. Let $X = \{i \in V : |v_i - v_k|^2 \le 1/(8n^2)\}$. Replacing each summand $\sum_{i<j} |v_i - v_j|^2 = 1$ with its upper bound $|v_i - v_j|^2 \le |v_k - v_i|^2 + |v_k - v_j|^2$ we get

$$\sum_{i \in V} |v_k - v_i|^2 \ge 1/(2n).$$

The contribution of the vertices from $X$ to the sum is at most $1/(8n^2) \cdot n = 1/(8n)$. Hence

$$\sum_{i \notin X} |v_k - v_i|^2 \ge 3/(8n).$$

Since $d(k,i) + d(i,k) = 2|v_k - v_i|^2$, either $\sum_{i \notin X} d(k,i)$ or $\sum_{i \notin X} d(i,k)$ is greater than or equal to $3/(16n)$. Let us assume that $\sum_{i \notin X} d(k,i) \ge 3/(16n)$ (the other case is similar). Notice that, for every $j \in X$

$$\begin{aligned} d(k,j) &= |v_k - v_j|^2 - (|v_k - v_0|^2 - |v_j - v_0|^2) \\ &\le 2|v_k - v_j|^2 \le 1/(4n^2). \end{aligned}$$

Consider a vertex $i \notin X$. Let $j$ be the closest to $i$ vertex in $X$ then

$$d(X,i) = d(j,i) \ge d(k,i) - d(k,j) \ge d(k,i) - 1/(4n^2).$$

Therefore,

$$\sum_{i \notin X} d(X,i) \ge \sum_{i \notin x} d(k,i) - \frac{n}{2} \cdot \frac{1}{4n^2} \ge \frac{1}{16n}.$$

Now, let $X_\varepsilon$ be the $\varepsilon$-neighborhood of $X$ w.r.t. the distance $d$, and $n(\varepsilon) = |\bar{X}_\varepsilon|$. We claim that one of the cuts $(X_\varepsilon, \bar{X}_\varepsilon)$ has directed edge expansion $O(n\beta)$. Indeed, let $\alpha$ be the minimum directed edge expansion among the cuts $(X_\varepsilon, \bar{X}_\varepsilon)$. In each of these cuts $\bar{X}_\varepsilon$ is smaller than $X_\varepsilon$ (since $|X_\varepsilon| \ge |X| \ge n/2$ by the condition). Therefore, $(X_\varepsilon, \bar{X}_\varepsilon)$ cuts at least $\alpha n(\varepsilon)$ edges. Hence

$$\int_0^1 \alpha n(\varepsilon) d\varepsilon \le \beta.$$

But

$$\int_0^1 n(\varepsilon) d\varepsilon = \sum_{i \notin X} d(X,i) \ge \frac{1}{16n}.$$

Therefore, $\alpha \le 16n\beta$. □