Approximating Sparsest Cut in Graphs of Bounded Treewidth*

Eden Chlamtac[1]**, Robert Krauthgamer[1]***, and Prasad Raghavendra[2]

[1] Weizmann Institute of Science, Rehovot, Israel.
{eden.chlamtac,robert.krauthgamer}@weizmann.ac.il
[2] Microsoft Research New England, Cambridge, MA, USA. pnagaraj@microsoft.com

Abstract. We give the first constant-factor approximation algorithm for Sparsest-Cut with general demands in bounded treewidth graphs. In contrast to previous algorithms, which rely on the flow-cut gap and/or metric embeddings, our approach exploits the Sherali-Adams hierarchy of linear programming relaxations.

1 Introduction

The Sparsest-Cut problem is one of the most famous graph optimization problems. The problem has been studied extensively due to the central role it plays in several respects. First, it represents a basic graph partitioning task that arises in several contexts, such as divide-and-conquer graph algorithms (see e.g. [29, 40] and [42, Chapter 21]). Second, it is intimately related to other graph parameters, such as flows, edge-expansion, conductance, spectral gap and bisection-width. Third, there are several deep technical links between Sparsest-Cut and two seemingly unrelated concepts, the Unique Games Conjecture and Metric Embeddings.

Given that Sparsest-Cut is known to be NP-hard [34], the problem has been studied extensively from the perspective of polynomial-time approximation algorithms. Despite significant efforts and progress in the last two decades, we are still quite far from determining the approximability of Sparsest-Cut. This is true not only for general graphs, but also for several important graph families, such as planar graphs or bounded treewidth graphs. The latter family is the focus of this paper; we shall return to it after setting up some notation and defining the problem formally.

Problem definition. For a graph $G = (V, E)$ we let $n = |V|$. For $S \subset V$, the cutset $(S, \bar{S}) \subset V \times V$ is the set of unordered pairs with exactly one endpoint in S, i.e. $\{\{u, v\} \in V \times V : u \in S, v \notin S\}$. In the Sparsest-Cut problem (with general

* A full version appears at http://arxiv.org/abs/1006.3970

** Supported in part by a Sir Charles Clore postdoctoral fellowship.

*** Supported in part by The Israel Science Foundation (grant #452/08), and by a Minerva grant.

demands), the input is a graph $G = (V, E)$ with edge capacities $\mathsf{cap} : E \to \mathbb{R}_{\geq 0}$ and a set of *demand pairs*, $D = (\{s_1, t_1\}, \ldots, \{s_k, t_k\})$ with a demand function $\mathsf{dem} : D \to \mathbb{R}_{\geq 0}$. The goal is to find $S \subset V$ (a cut of G) that minimizes the ratio

$$\Phi(S) = \frac{\sum_{(u,v)\in(S,\bar{S})\cap E} \mathsf{cap}(u, v)}{\sum_{(u,v)\in(S,\bar{S})\cap D} \mathsf{dem}(u, v)}.$$

The demand function dem is often set to $\mathsf{dem}(s, t) = 1$ for all $(s, t) \in D$. The special case where, in addition to this, the demand set D includes all vertex pairs is referred to as *uniform demands*.

Treewidth. Let $G = (V, E)$ be a graph. A *tree decomposition* of $G = (V, E)$ is a pair $(\mathcal{B}, T)$ where $\mathcal{B} = \{B_1, \ldots, B_m\}$ is a family of subsets $B_i \subseteq V$ called *bags*, and T is a tree whose nodes are the bags B_i, satisfying the following properties: (i) $V = \bigcup_i B_i$; (ii) For every edge $(u, v) \in E$, there is a bag B_j that contains both u, v; and (iii) For each $v \in V$, all the bags B_i containing v form a connected subtree of T. The *width* of the tree decomposition is $\max_i |B_i| - 1$. The *treewidth* of G, denoted $\mathrm{tw}(G)$, is the smallest width among all tree decompositions of G. The *pathwidth* of G is defined similarly, except that T is restricted to be a path; thus, it is at least $\mathrm{tw}(G)$. It is straightforward to see that every graph G excludes as a minor the complete graph on $\mathrm{tw}(G) + 2$ vertices. Thus, the family of graphs of tree width r contains the family of graphs with pathwidth r, and is contained in the family of graphs excluding K_{r+2} as a minor (here K_{r+2} refers to the complete graph on $r + 2$ vertices).

1.1 Results

We present the first algorithm for general demand Sparsest-Cut that achieves a constant factor approximation for graphs of bounded treewidth r (the restriction is only on the structure of the graph, not the demands). Such an algorithm is conjectured to exist by [20] (they actually make a stronger conjecture, see Section 1.3 for details). However, previously such an algorithm was not known even for $r = 3$, although several algorithms are known for $r = 2$ [20, 7, 14] and for bounded-pathwidth graphs [28] (which is a subfamily of bounded-treewidth graphs).

Theorem 1. *There is an algorithm for* Sparsest-Cut *(general demands) on graphs of treewidth r, that runs in time $(2^r n)^{O(1)}$ and achieves approximation factor $C = C(r)$ (independently of n, the size of the graph).*

Table 1 lists the best approximation algorithms known for various special cases of Sparsest-Cut. We remark that the problem (with general demands) is NP-hard even for pathwidth 2 (see the full version for details).

Demands	Graphs	Approximation	Based on	Reference
general	arbitrary	$\tilde{O}(\sqrt{\log\|D\|})$	SDP	[2]
	treewidth 2	2	LP (flow)	[20, 7]
	fixed outerplanarity	$O(1)$	LP (integer flow)	[12, 14]
	excluding W_4-minor	$O(1)$	LP (flow)	[7]
	fixed pathwidth	$O(1)$	LP (flow)	[28]
	fixed treewidth	$O(1)$	LP (lifted)	This work
uniform	arbitrary	$O(\sqrt{\log n})$	SDP	[3]
	excluding fixed-minor	$O(1)$	LP (flow)	[24, 18]
	fixed treewidth	O(1)	LP (flow)	[35, 13]
	fixed treewidth	1	dynamic programming	

Table 1. Approximation algorithms for Sparsest-Cut.

Techniques. Similarly to almost all previous work, our algorithm is based on rounding a linear programming (LP) relaxation of the problem. A unique feature of our algorithm is that it employs an LP relaxation derived from the hierarchy of (increasingly stronger) LPs, designed by Sherali and Adams [39]. Specifically, we use level $r + O(1)$ of this hierarchy. In contrast, all prior work on Sparsest-Cut uses either the standard LP (that arises as the dual of the concurrent-flow problem, see e.g. [29]), or its straightforward strengthening to a semidefinite program (SDP). Consequently, the entire setup changes significantly (e.g. the known connections to embeddings and flow, see Section 1.2), and we face the distinctive challenges of exploiting the complex structure of these relaxations (see Section 1.3).

While bounding the integrality gap of the standard LP (the flow-cut gap) for various graph families remains an important open problem with implications in metric embeddings (see Section 1.2), our focus is on directly approximating Sparsest-Cut. Accordingly, our LP is larger and (possibly much) stronger than the standard flow LP, and hence our rounding does not imply a bound on the flow-cut gap (akin to rounding of the SDP relaxation in [3, 10, 2]).

Finally, note that the running time stated in Theorem 1 is much better than the $n^{O(r)}$ running time typically needed to solve the $r + O(1)$ level of Sherali-Adams (or any other hierarchy). The reason is that only $O(3^r n|D|)$ of the Sherali-Adams variables and constraints are really needed for our analysis to go through (see Remark 1), thus greatly improving the time needed to solve the LP. As the rounding algorithm we use is a simple variant of the standard method of randomized rounding for LP's (adapted for Sherali-Adams relaxations on bounded-treewidth graphs), the entire algorithm is both efficient and easily implementable.

1.2 The GNRS excluded-minor conjecture

Gupta, Newman, Rabinovich and Sinclair (GNRS) conjectured in [20] that metrics supported on graphs excluding a fixed minor embed into ℓ_1 with distortion

$O(1)$ (i.e. independent of the graph size). By the results of [30, 4, 20], this conjecture is equivalent to saying that in all such graphs (regardless of the capacities and demands), the ratio between the sparsest-cut and the concurrent-flow, called the *flow-cut gap*, is bounded by $O(1)$. Since the concurrent-flow problem is polynomial-time solvable (e.g. by linear programming), the conjecture would immediately imply that Sparsest-Cut admits $O(1)$ approximation (in polynomial-time) on these graphs.

Despite extensive research, the GNRS conjecture is still open, even in the special cases of planar graphs and of graphs of treewidth 3. The list of special cases that have been resolved includes graphs of treewidth 2, $O(1)$-outerplanar graphs, graphs excluding a 4-wheel minor, and bounded-pathwidth graphs; see Table 1, where the flow LP is mentioned.

Our approximation algorithm may be interpreted as evidence supporting the GNRS conjecture (for graphs of bounded treewidth), since by the foregoing discussion, the conjecture being true would imply the existence of such approximation algorithms, and moreover that our LP's integrality gap is bounded. In fact, one consequence of our algorithm and its analysis can be directly phrased in the language of metric embeddings:

Corollary 1. *For every r there is some constant $C = C(r)$ such that every shortest-path metric on a graph of treewidth $\leq r$, for which every set of size $r + 3$ is isometrically embeddable into L_1 in a locally consistent way (i.e. the embeddings of two such sets, when viewed as probability distributions over cuts, are consistent on the intersection of the sets), can be embedded into L_1 with distortion at most C.*

If, on the other hand, the GNRS conjecture is false, then our algorithm (and its stronger LP) gives a substantial improvement over techniques using the flow LP, and may have surprising implications for the Sherali-Adams hierarchy (see Section 1.3). Either way, our result opens up several interesting questions, which we discuss in Section 1.4.

1.3 Related work

Relaxation hierarchies and approximation algorithms. A research plan that has attracted a lot of attention in recent years is the use of lift-and-project methods to design improved approximation algorithms for NP-hard optimization problems. These methods, such as Sherali-Adams [39], Lovász-Schrijver [31], and Lasserre [26] (see [27] for a comparison), systematically generate, for a given $\{0, 1\}$ program (which can capture many combinatorial optimization problems, e.g. Vertex-Cover), a sequence (aka *hierarchy*) of increasingly stronger relaxations. The first relaxation in this sequence is often a commonly-used LP relaxation for that combinatorial problem. After n steps (which are often called *rounds* or *levels*), the sequence converges to the convex hull of the integral solutions, and the k-th relaxation in the sequence is a convex program (LP or SDP) that can be solved in time $n^{O(k)}$. Therefore, the first few, say $O(1)$, relaxations in the sequence offer a great promise to approximation algorithms — they could be much stronger than the commonly-used LP relaxation, yet are polynomial-time computable. This is particularly promising for problems for which there is a gap between known approximations and proven hardness of approximation (or when the hardness relies on weaker assumptions than $P \neq NP$).

Unfortunately, since the work of Arora, Bollobás, Lovász, and Tourlakis [1] on Vertex-Cover, there has been a long line of work showing that for various problems, even after a large (super-constant) number of rounds, various hierarchies do not yield smaller integrality gaps than a basic LP/SDP relaxation (see, e.g. [38, 19, 37, 41, 8]). In particular, Raghavendra and Steurer [36] have recently shown that a superconstant number of rounds of certain SDP hierarchies does not improve the integrality gap for any constraint satisfaction problem (MAX-CSP).

In contrast, only few of the known results are positive, i.e. show that certain hierarchies give a sequence of improvements in the integrality gap in their first $O(1)$ levels — this has been shown for Vertex-Cover in planar graphs [32], Max-Cut in dense graphs [43], Knapsack [21, 6], and Maximum Matching [33]. There are even fewer results where the improved approximation is the state-of-the-art for the respective problem — such results include recent work on Chromatic Number [15], Hypergraph Independent Set [16], and MaxMin Allocation [5].

In the context of bounded-treewidth graphs, a bounded number of rounds in the Sherali-Adams hierarchy is known to be tight (i.e. give exact solutions) for many problems that are tractable on this graph family, such as CSPs [44]. This is only partially true for Sparsest-Cut — due to the exact same reason, we easily find in the graph a cut whose edge capacity exactly matches the corresponding expression in the LP. However, the demands are arbitrary (and in particular do not have a bounded-treewidth structure), and analyzing them requires considerably more work.

Hardness and integrality gaps for sparsest-cut. As mentioned earlier, Sparsest-Cut is known to be NP-hard [34], and we further show in the full version that it is even NP-hard on graphs of pathwidth 2. Two results [23, 9] independently proved that under Khot's unique games conjecture [22], the Sparsest-Cut problem is NP-hard to approximate within any constant factor. However, the graphs produced by the reductions in these two results have large treewidth.

The standard flow LP relaxation for Sparsest-Cut was shown in [29] to have integrality gap $\Omega(\log n)$ in expander graphs, even for uniform demands. Its standard strengthening to an SDP relaxation (the SDP used by the known approximation algorithms of [3, 2]) was shown in [23, 25, 17] to have integrality gap $\Omega(\log \log n)$, even for uniform demands. For the case of general demands, a stronger bound $(\log n)^{\Omega(1)}$ was recently shown in [11]. Some of these results were extended in [8, 36] to certain hierarchies and a nontrivial number of rounds, even for uniform demands. Again, the graphs used in these results have large treewidth.

Integrality gaps for graphs of treewidth r (or excluding a fixed minor of size r) follow from the above in the obvious way of replacing n with r (or so), for instance, the standard flow LP has integrality gap $\Omega(\log r)$. However, no stronger gaps are known for these families; in particular, it is possible that the integrality gap approaches 1 with sufficiently many rounds (depending on r, but not on n).

1.4 Discussion and further questions

We show that for the Sparsest-Cut problem, the Sherali-Adams (SA) LP hierarchy can yield algorithms with better approximation ratio than previously known. Moreover, our analysis exhibits a strong (but rather involved) connection between the input graph's treewidth and the SA hierarchy level. Several interesting questions arise immediately:

1. Can this approach be generalized to excluded-minor graphs?
2. Can the approximation factor be improved to an absolute constant (independent of the treewidth)?

A particularly intriguing and more fundamental question is whether this hierarchy (or a related one, or for a different input family) is strictly stronger than the standard LP (or SDP) relaxation. One possibility is that our relaxation can actually yield an absolute constant factor approximation (as in Question 2). Such an approximation factor is shown in [8] to require at least $\Omega(\log r)$ rounds of Sherali-Adams, and we would conclude that hierarchies yield strict improvement — higher (yet constant) levels of the Sherali-Adams hierarchy do give improved approximation factors, for an increasing sequence of graph families. We note, however, that this would require a different rounding algorithm (see Remark 2). Another possibility is that the GNRS conjecture does not hold even for bounded treewidth graphs, in which case the integrality gap of the standard LP exhibits a dependence on n, while, as we prove here, the stronger LP does not.

2 Technical Overview

Relaxations arising from the Sherali-Adams (SA) hierarchy, and lift-and-project techniques in general, are known to give LP (or SDP) solutions which satisfy the following property: for every subset of variables of bounded size (bounded by the level in the hierarchy used), the LP/SDP solution restricted to these variables is a convex combination of valid $\{0, 1\}$ assignments. Such a convex combination can naturally be viewed as a distribution on local assignments. In our case, for example, in an induced subgraph on $r + 1$ vertices S, an $(r + 1)$-level relaxation gives a local distribution on assignments $f : S \to \{0, 1\}$ such that for every edge (i, j) within S, the probability that $f(i) \neq f(j)$ is exactly the contribution of edge (i, j) to the objective function (which we also call the *LP-distance* of this pair). Our algorithm makes explicit use of this property, which is very useful for treewidth r graphs.

Given an $(r+3)$-level Sherali-Adams relaxation, for every demand pair there is some distribution which (within every bag) matches the local distributions suggested by the LP, and also *cuts/separates* this demand pair (i.e. assigns different values to its endpoints) with the correct probability (the LP distance). Unfortunately, there might not be any single distribution which is consistent with all demand pairs, so instead our algorithm assigns $\{0,1\}$ values at random to the vertices of the graph G in a stochastic process which matches the local distributions suggested by the LP solution (per bag), but is oblivious to the structure of the demands D.

Intuition. To achieve a good approximation ratio, it suffices to ensure that every demand pair is cut with probability not much smaller than the its LP distance. To achieve this, the algorithm fixes an arbitrary bag as the root, and traverses the tree decomposition one bag at a time, from the root towards the leaves, and samples the assignment to currently unassigned vertices in the current bag. This assignment is sampled in a way that ignores all previous assignments to vertices outside the current bag, but achieves the correct distribution on assignments to the current bag. Essentially, the algorithm finds locally correct distributions while maximizing the entropy of the overall distribution. Intuitively, this should only "distort" the distribution suggested by the LP (for a given demand pair) only by introducing noise, which (if the noise is truly unstructured) mixes the correct global distribution with a completely random one in which every two vertices are separated with probability $\frac{1}{2}$. In this case, the probability of separating any demand pair would decrease by at most a factor 2. Unfortunately, we are not able to translate this intuition into a formal proof (and on some level, it is not accurate – see Remark 2). Thus we are forced to adopt a different strategy in analyzing the performance of the rounding algorithm. Let us see one illustrative special case.

Example: Simple Paths. Consider, for concreteness, the case of a single simple path $v_1, v_2, \ldots, v_n$. For every edge in the path (v_{i-1}, v_i), the LP suggests cutting it (assigning different values) with some probability p_i. Our algorithm will perform the following Markov process: pick some assignment $f(v_1) \in \{0,1\}$ at random according to the LP, and then, at step i (for $i = 2, \ldots, n$) look only at the assignment $f(v_{i-1})$ and let $f(v_i) = 1 - f(v_{i-1})$ with probability p_i, and $f(v_i) = f(v_{i-1})$ otherwise. Each edge has now been cut with exactly the probability corresponding to its LP distance. However, for (v_1, v_n), which could be a demand pair, the LP distance between them might be much greater than the probability $q_n = \Pr[f(v_1) \neq f(v_n)]$. Let us see that the LP distance can only be a constant factor more.

First, if the above probability satisfies $q_n \geq \frac{1}{3}$, then clearly we are done, as all LP distances will be at most 1. Thus we may assume that $q_n \leq \frac{1}{3}$. Let us examine what happens at a single step. Suppose the algorithm has separated v_1 from v_{i-1} with some probability $q_{i-1} \leq \frac{1}{3}$ (assuming that all $q_i \leq \frac{1}{3}$ is a somewhat stronger assumption than $q_n \leq \frac{1}{3}$, but a more careful analysis shows it is also valid). After the current step (flipping sides with probability p_i), the probability that v_i is separated from v_1 is exactly $(1 - q_{i-1})p_i + q_{i-1}(1 - p_i)$. This is an increase over the previous value q_{i-1} of at least

$$[(1 - q_{i-1})p_i + q_{i-1}(1 - p_i)] - q_{i-1} = (1 - 2q_{i-1})p_i \geq p_i/3.$$

However, the LP distance from v_1 can increase by at most p_i (by triangle inequality). Thus, we can show inductively that we never lose more than a factor 3.

In general, our analysis will consider paths of bags of size $r+1$. Even though we can still express the distribution on assignments chosen by the rounding algorithm as a Markov process (where the possible states at every step will be assignments to some set of at most r vertices), it will be less straightforward to relate the LP values to this process. It turns out that we can get a handle on the LP distances by modeling the Markov process as a layered digraph H with edges capacities representing the transitions (this is only in the analysis, or in the derandomization of our algorithm). In this case the LP distance we wish to bound becomes the value of a certain (s,t)-flow in H. We then bound the flow-value from above by finding a small cut in H. Constructing and bounding the capacity of such a cut in H constitutes the technical core of this work.

3 The Algorithm

3.1 An LP relaxation using the Sherali Adams hierarchy

Let us start with an informal overview of the Sherali-Adams (SA) hierarchy. In an LP relaxation for a 0–1 program, the linear variables $\{y_i \mid i \in [n]\}$ represent linear relaxations of integer variables $x_i \in \{0,1\}$. We can extend such a relaxation to include variables $\{y_I\}$ for larger subsets $I \subseteq [n]$ (usually, up to some bounded cardinality). These should be interpreted as representing the products $\prod_{i \in I} x_i$ in the intended (integer) solution. Now, for any pair of sets $I, J \subseteq [n]$, we will denote by $y_{I,J}$ the linear relaxation for the polynomial $\prod_{i \in I}(1 - x_i) \prod_{j \in J} x_j$. These can be derived from the variables y_I by the inclusion-exclusion principle. That is, we define

$$y_{I,J} = \textstyle\sum_{I' \subseteq I} (-1)^{|I'|} y_{I' \cup J}.$$

The constraints defined by the polytope $\mathbf{SA}_t(n)$, that is, level t of the Sherali-Adams hierarchy starting from the trivial n-dimensional LP, are simply the inclusion-exclusion constraints:

$$\forall I, J \subseteq [n] \text{ s.t. } |I \cup J| \leq t \ : \ y_{I,J} \geq 0 \tag{1}$$

For every solution other than the trivial (all-zero) solution, we can define a normalized solution $\{\tilde{y}_I\}$ as follows:

$$\tilde{y}_I = y_I / y_\emptyset,$$

and the normalized derived variables $\tilde{y}_{I,J}$ can be similarly defined.

As is well-known, in a non-trivial level t Sherali-Adams solution, for every set of (at most) t vertices, constraints (1) imply a distribution on $\{0,1\}$ assignments to these vertices matching the LP values:

Lemma 1. *Let $\{y_I\}$ be a non-zero vector in the polytope $\boldsymbol{SA}_t(n)$. Then for every set $L \subseteq [n]$ of cardinality $|L| \leq t$, there is a distribution μ_L on assignments $f : L \to \{0,1\}$ such that for all $I, J \subseteq L$,*

$$\Pr_{\mu_L}\left[(\forall i \in I : f(i) = 0) \ \wedge \ (\forall j \in J : f(j) = 1)\right] = \tilde{y}_{I,J}.$$

In a Sparsest Cut relaxation, we are interested in the event in which a pair of vertices is cut (i.e. assigned different values). This is captured by the following linear variable:

$$y_{i \neq j} = y_{\{i\},\{j\}} + y_{\{j\},\{i\}}.$$

We can now define our relaxation for Sparsest Cut, $\mathbf{SC}_r(G)$:

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in E} \mathsf{cap}(i,j) y_{i\neq j} && (2)\\
\text{s.t.} \quad & \sum_{i,j\in D} \mathsf{dem}(i,j) y_{i\neq j} = 1 && (3)\\
& \{y_I\} \in \mathbf{SA}_{r+3}(n) && (4)\\
& y_{I,J} = y_{J,I} \qquad \forall I, J \text{ s.t. } |I \cup J| \leq r+3 && (5)
\end{aligned}$$

Note that constraint (3) is simply a normalization ensuring that the objective function is really a relaxation for the ratio of the two sums. Also note that constraint (5), which ensures that the LP solution is fully symmetric, does not strengthen the LP, in the following sense: For any solution $\{y'_I\}$ to the above LP without constraint (5), a new solution to the symmetric LP (with the same value in the objective function) can be achieved by taking $y_I = (y'_I + y'_{I,\emptyset})/2$ without violating any of the other constraints. In particular, for every vertex $i \in V$ this gives $\tilde{y}_i = 1 - \tilde{y}_i = \frac{1}{2}$. While our results hold true without imposing this constraint, we will retain it as it simplifies our analysis.

Remark 1. The size of this LP (and the time needed to solve it) is $n^{O(r)}$. Specifically for bounded-treewidth graphs, we could also formulate a much smaller LP, where constraint (4) would be replaced with the condition $\{y_I \mid I \subseteq B \cup \{i,j\}\} \in \mathbf{SA}_{r+3}(r+3)$ for every bag B and demand pair $(i,j) \in D$. This would reduce the size of the LP to (and time needed to solve it) to at most $\text{poly}(2^r n)$, and our rounding algorithm and analysis would still hold.

3.2 Rounding the LP

Before we present the rounding algorithm, let us introduce some notation which will be useful in describing the algorithm. This notation will allow us to easily go back-and-forth between the LP solution and the local distributions on assignments described in Lemma 1. For ease of notation, whenever two functions f_1, f_2 have disjoint domains, we will denote by $f_1 \cup f_2$ the unique function from the union of the domains which is an extension of both f_1 and f_2.

- For every set of vectors $\{y_I\}$ and subset $L \subseteq [n]$ as in Lemma 1, we will denote by $\mu_L^{\{y_I\}}$ the distribution on random assignments to L guaranteed by the lemma. We will omit the superscript $\{y_I\}$, and simply write μ_L, when it is clear from the context.
- Conversely, for any fixed assignment $f' : L \to \{0,1\}$, we will write $\tilde{y}_{f'} = \tilde{y}_{L_0,L_1}$, where $L_b = \{i \in L \mid f'(i) = b\}$ for $b = 0,1$. Thus, for a random assignment $f : L \to \{0,1\}$ distributed according to μ_L, we have $\Pr[f = f'] = \tilde{y}_{f'}$.
- For any nonempty subset $L' \subseteq L$, and a given assignment $f_0 : L \setminus L' \to \{0,1\}$ in the support of $\mu_{L \setminus L'}$, we will denote by μ_{L',f_0} the distribution on random assignments $f \sim \mu_L$ conditioned on the partial assignment f_0. Formally, a random assignment $f' : L' \to \{0,1\}$, distributed according to μ_{L',f_0} satisfies $\Pr_{f'}[f' = f_1] = \tilde{y}_{f_0 \cup f_1}/\tilde{y}_{f_0}$ for every choice of $f_1 : L' \to \{0,1\}$.

Let G be an graph with treewidth r for some integer $r > 0$, and let $(\mathcal{B}, T)$ be the corresponding tree decomposition. Let $\{y_I\}$ be a vector satisfying $\mathbf{SC}_r(G)$. We now present the rounding algorithm:

> Algorithm $\mathsf{SC\text{-}Round}(G, (\mathcal{B}, T), \{y_I\})$ [Constructs a random assignment f]
>
> 1. Pick an arbitrary $B_0 \in \mathcal{B}$ as the root of $\mathcal{T}$, and sample $f|_{B_0} \sim \mu_{B_0}$.
> 2. Traverse the rest of the tree T in any order from the root towards the leaves. For each bag B traversed, do the following:
>   (a) Let B^+ be the set of vertices in B for which f is already defined, and let $B^- = B \setminus B^+$. Let f_0 be the existing assignment $f_0 = f|_{B^+}$.
>   (b) If B^- is non-empty, sample $f|_{B^-}$ at random according to μ_{B^-,f_0}.

Let us first see that every edge $(i,j) \in E$ is cut with probability exactly $\tilde{y}_{i \neq j}$. Since every edge is contained in at least one bag, it suffices to show that within every bag B, the assignment $f|_B$ is distributed according to μ_B. This is shown by the following lemma, whose straightforward proof appears in the full version.

Lemma 2. *For every bag B, the assignment $f|_B$ produced by running algorithm $\mathsf{SC\text{-}Round}(G, (\mathcal{B}, T), \{y_I\})$ is distributed according to μ_B.*

This lemma shows that the expected value of the cut is $\sum_{(i,j)\in E} \mathsf{cap}(i,j)\tilde{y}_{i\neq j}$, which is exactly the value of the objective function (2) scaled by $1/y_\emptyset$. In particular, for a host of other problems where the objective function and constraints depend only on the edges (e.g. Minimum Vertex Cover, Chromatic Number), this type of LP relaxation (normalized by setting $y_\emptyset = 1$), along with the above rounding, always produces an optimal solution for bounded-treewidth graphs. Thus, in some sense, we consider this to be a "natural" rounding algorithm.

Before we analyze the expected value of the cut demands (or specifically, the probability that each demand is cut), let us show that the order in which the tree T is traversed has no effect on the distribution of cuts produced (it will suffice to show a slightly weaker claim – that the joint distribution of cuts in any two bags is not affected). This is shown in the following lemma, whose proof appears in the appendix.

Lemma 3. *Let $B_1, B_2 \in \mathcal{B}$ be two arbitrary bags. Then the distribution on assignments $f|_{B_1 \cup B_2}$ is invariant under any connected traversal of T.*

4 Markov Flow Graphs

We show the following lemma, whose proof appears in the full version, which together with Lemma 2 implies Theorem 1 (see Remark 3).

Lemma 4. *For every integer $r > 0$ there exists a constant $c_r > 0$ such that for any treewidth-r graph G with tree decomposition $(\mathcal{B}, T)$, and vectors $\{y_I\}$ satisfying $SC_r(G)$, algorithm SC-Round$(G, (\mathcal{B}, T), \{y_I\})$ outputs a random $f : V \to \{0,1\}$ s.t. for every $i, j \in V$,*

$$\Pr[f(i) \neq f(j)] \geq c_r \tilde{y}_{i \neq j}. \tag{6}$$

Remark 2. The constant c_r arising in our analysis is quite small (roughly 2^{-r2^r}). While we believe this can be improved, we cannot eliminate the dependence on r, as a lower bound on the performance of our rounding algorithm (which appears in the full version) shows that c_r cannot be more than $2^{-r/2}$.

Remark 3. In fact, Lemmas 2 and 4 taken together show the following: Given any solution to $\mathrm{SC}_r(G)$ with objective function value $\alpha > 0$, algorithm SC-Round produces a random assignment f satisfying

$$\mathbb{E}\left[\sum_{(i,j)\in E} \mathsf{cap}(i,j)\,|f(i) - f(j)| - \tfrac{\alpha}{c_r} \sum_{(i,j)\in D} \mathsf{dem}(i,j)\,|f(i) - f(j)|\right] \leq 0.$$

This means the algorithm produces a $1/c_r$-approximation with positive probability, but does not immediately imply a lower bound on that probability. Fortunately, following the analysis in this section, the algorithm can be derandomized by the method of conditional expectations, since, at each step, finding the probability of separating each demand pair reduces to calculating the probability of reaching a certain state at a certain phase in some Markov process, which simply involves multiplying $O(n)$ transition matrices of size at most $2^r \times 2^r$ (in fact, these can be consolidated so that every step of the algorithm involves a total of $O(n|T|)$ small matrix multiplications for all demands combined, where T is the set of vertices participating in demand pairs).

For vertices $i, j \in V$ belonging to (at least) one common bag, Lemma 2 implies equality in (6) for $c_r = 1$. For $i, j \in V$ which do not lie in the same bag, consider the path of bags $B_1, \ldots, B_N$ in tree T from the (connected) component of bags containing i to the component of bags containing j. By Lemma 3, we may assume that the algorithm traverses the path in order from B_1 to B_N.

To understand the event that vertices i and j are separated, it suffices to consider the following incomplete (but consistent) description of the stochastic process involved: Let $S_0 = \{i\}$ and $S_N = \{j\}$, and let $S_l = B_l \cap B_{l+1}$ for

$l = 1, \ldots, N-1$. The algorithm assigns $f(i)$ a value in $\{0,1\}$ uniformly at random, and then for $l = 1, \ldots, N$, samples $f|_{S_l}$ from the distribution $\mu_{S_l, f|_{S_{l-1}}}$ (we extend the definition of $\mu_{S,f'}$ in the natural way to include the case where S may intersect the domain of f').

This is a Markov process, and can be viewed as a Markov flow graph. That is, a layered graph, where each layer consists of nodes representing the different states (in this case, assignments to S_l), with exactly one unit of flow going from the first to the last layer, with all edges having flow at full capacity. Since all edges in the flow graph represent pairs of assignments within the same bag, Lemma 2 implies that the capacity of an edge (transition) (f_1, f_2) is exactly $\tilde{y}_{f_1 \cup f_2}$, and the amount of flow going through each node f_0 is $\tilde{y}_{f_0}$.

We now would like to analyze the contribution of a demand pair to the LP. By constraint (5), this contribution (up to a factor $\mathsf{dem}(i,j)$) is $\tilde{y}_{i \neq j} = 2\tilde{y}_{\{i\},\{j\}} = 2\tilde{y}_{f^*}$, where $f^* : \{i,j\} \to \{0,1\}$ is the function assigning 0 to i and 1 to j. Now consider a layer graph as above where each edge (f_1, f_2) has flow $\tilde{y}_{f^* \cup f_1 \cup f_2}$. To see that this is indeed a flow, note that two consecutive layers along with i and j only involve at most $r+3$ vertices in G, and so by Lemma 1 for any $l > 0$ and function $f_2 : S_l \to \{0,1\}$ the incoming flow at f_2 must be $\sum_{f_1 \in S_{l-1}} \tilde{y}_{f^* \cup f_1 \cup f_2} = \tilde{y}_{f^* \cup f_2}$, and so is the outgoing flow. The total flow in this graph is exactly $\tilde{y}_{f^*}$ (half the LP contribution $\tilde{y}_{i \neq j}$). Moreover, for each such edge (transition) we also have $\tilde{y}_{f^* \cup f_1 \cup f_2} \leq \tilde{y}_{f_1 \cup f_2}$. Hence, the flow with values $\{\tilde{y}_{f^* \cup f_1 \cup f_2}\}$ is a legal flow respecting the capacities $\{\tilde{y}_{f_1 \cup f_2}\}$ in the Markov flow graph which represents the rounding algorithm.

Thus it suffices to show the following theorem (proved in the full version):

Theorem 2. *For every integer $k > 1$, there is a constant $C = C(k) > 0$ such that for any symmetric Markov flow graph $G = (L_0, \ldots, L_N, E)$ representing a Markov process $X_0, \ldots, X_N$ with sources $L_0 = \{s_0, s_1\}$ and sinks $L_N = \{t_0, t_1\}$ and at most k nodes per layer, the total amount of capacity-respecting flow in G from s_0 to t_1 can be at most $C \cdot \Pr[X_0 = s_0 \wedge X_N = t_1]$.*

Applying this theorem to the Markov flow graph described above with $k = 2^r$ immediately implies Lemma 4. As usual, to bound the amount of flow in a graph from above, it suffices to find a suitable cut. See the full version for details.

Acknowledgments We would like to thank Claire Mathieu for a series of helpful conversations.

References

[1] Arora, S., Bollobás, B., Lovász, L., Tourlakis, I.: Proving integrality gaps without knowing the linear program. Theory of Computing 2(1), 19–51 (2006)

[2] Arora, S., Lee, J.R., Naor, A.: Euclidean distortion and the sparsest cut. J. Amer. Math. Soc. 21(1), 1–21 (2008)

[3] Arora, S., Rao, S., Vazirani, U.: Expander flows, geometric embeddings and graph partitioning. J. ACM 56(2), 1–37 (2009)

[4] Aumann, Y., Rabani, Y.: An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. SIAM J. Comput. 27(1), 291–301 (1998)

[5] Bateni, M., Charikar, M., Guruswami, V.: Maxmin allocation via degree lower-bounded arborescences. In: 41st annual ACM symposium on Theory of computing. pp. 543–552. ACM, New York, NY, USA (2009)

[6] Bienstock, D.: Approximate formulations for 0-1 knapsack sets. Oper. Res. Lett. 36(3), 317–320 (2008)

[7] Chakrabarti, A., Jaffe, A., Lee, J.R., Vincent, J.: Embeddings of topological graphs: Lossy invariants, linearization, and 2-sums. In: 49th Annual IEEE Symposium on Foundations of Computer Science. pp. 761–770 (2008)

[8] Charikar, M., Makarychev, K., Makarychev, Y.: Integrality gaps for Sherali-Adams relaxations. In: 41st ACM symposium on Theory of computing. pp. 283–292 (2009)

[9] Chawla, S., Krauthgamer, R., Kumar, R., Rabani, Y., Sivakumar, D.: On the hardness of approximating multicut and sparsest-cut. Computational Complexity 15(2), 94–114 (2006)

[10] Chawla, S., Gupta, A., Räcke, H.: Embeddings of negative-type metrics and an improved approximation to generalized sparsest cut. ACM Transactions on Algorithms 4(2) (2008)

[11] Cheeger, J., Kleiner, B., Naor, A.: A $(\log n)^{\Omega(1)}$ integrality gap for the sparsest cut SDP. In: 50th Annual IEEE Symposium on Foundations of Computer Science. vol. 0, pp. 555–564. IEEE (2009)

[12] Chekuri, C., Gupta, A., Newman, I., Rabinovich, Y., Sinclair, A.: Embedding k-outerplanar graphs into ℓ_1. SIAM J. Discret. Math. 20(1), 119–136 (2006)

[13] Chekuri, C., Khanna, S., Shepherd, F.B.: A note on multiflows and treewidth. Algorithmica 54(3), 400–412 (2009)

[14] Chekuri, C., Shepherd, B., Weibel, C.: Flow-cut gaps for integer and fractional multiflows. In: 21st ACM-SIAM Symposium on Discrete Algorithms (2010)

[15] Chlamtac, E.: Approximation algorithms using hierarchies of semidefinite programming relaxations. In: 48th Annual IEEE Symposium on Foundations of Computer Science. pp. 691–701 (2007)

[16] Chlamtac, E., Singh, G.: Improved approximation guarantees through higher levels of SDP hierarchies. In: 11th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX). pp. 49–62 (2008)

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

[18] Fakcharoenphol, J., Talwar, K.: Improved decompositions of graphs with forbidden minors. In: 6th International workshop on Approximation algorithms for combinatorial optimization. pp. 36–46 (2003)

[19] Georgiou, K., Magen, A., Pitassi, T., Tourlakis, I.: Integrality gaps of 2 - o(1) for vertex cover SDPs in the Lovász-Schrijver hierarchy. In: 48th Annual IEEE Symposium on Foundations of Computer Science. pp. 702–712 (2007)

[20] Gupta, A., Newman, I., Rabinovich, Y., Sinclair, A.: Cuts, trees and ℓ_1-embeddings of graphs. Combinatorica 24(2), 233–269 (2004)

[21] Karlin, A.R., Mathieu, C., Nguyen, C.T.: Integrality gaps of linear and semidefinite programming relaxations for knapsack (2009), manuscript

[22] Khot, S.: On the power of unique 2-prover 1-round games. In: 34th Annual ACM Symposium on the Theory of Computing. pp. 767–775 (Jul 2002)

[23] Khot, S., Vishnoi, N.K.: The unique games conjecture, integrality gap for cut problems and the embeddability of negative type metrics into ℓ_1. In: 46th IEEE Annual Symposium on Foundations of Computer Science. pp. 53–62 (2005)
[24] Klein, P., Plotkin, S.A., Rao, S.: Excluded minors, network decomposition, and multicommodity flow. In: 25th Annual ACM Symposium on Theory of Computing. pp. 682–690 (May 1993)
[25] Krauthgamer, R., Rabani, Y.: Improved lower bounds for embeddings into L_1. SIAM J. Comput. 38(6), 2487–2498 (2009)
[26] Lasserre, J.B.: Semidefinite programming vs. LP relaxations for polynomial programming. Math. Oper. Res. 27(2), 347–360 (2002)
[27] Laurent, M.: A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0–1 programming. Math. Oper. Res. 28(3), 470–496 (2003)
[28] Lee, J.R., Sidiropoulos, A.: Pathwidth, trees, and random embeddings. CoRR abs/0910.1409 (2009)
[29] Leighton, T., Rao, S.: Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. J. ACM 46(6), 787–832 (1999)
[30] Linial, N., London, E., Rabinovich, Y.: The geometry of graphs and some of its algorithmic applications. Combinatorica 15(2), 215–245 (1995)
[31] Lovász, L., Schrijver, A.: Cones of matrices and set-functions and 0-1 optimization. SIAM J. Optim. 1(2), 166–190 (1991)
[32] Magen, A., Moharrami, M.: Robust algorithms for maximum independent set on minor-free graphs based on the Sherali-Adams hierarchy. In: 12th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX). pp. 258–271 (2009)
[33] Mathieu, C., Sinclair, A.: Sherali-Adams relaxations of the matching polytope. In: 41st annual ACM symposium on Theory of computing. pp. 293–302 (2009)
[34] Milman, V.D., Schechtman, G.: Asymptotic theory of finite-dimensional normed spaces. Springer-Verlag, Berlin (1986)
[35] Rabinovich, Y.: On average distortion of embedding metrics into the line and into L_1. In: 35th annual ACM symposium on Theory of computing. pp. 456–462 (2003)
[36] Raghavendra, P., Steurer, D.: Integrality gaps for strong SDP relaxations of unique games. In: 50th IEEE Symposium on Foundations of Computer Science. pp. 575–585 (2009)
[37] Schoenebeck, G.: Linear level Lasserre lower bounds for certain k-CSPs. In: 49th IEEE Symposium on Foundations of Computer Science. pp. 593–602 (2008)
[38] Schoenebeck, G., Trevisan, L., Tulsiani, M.: A linear round lower bound for Lovász-Schrijver SDP relaxations of vertex cover. In: 22nd Annual IEEE Conference on Computational Complexity. pp. 205–216 (2007)
[39] Sherali, H.D., Adams, W.P.: A hierarchy of relaxation between the continuous and convex hull representations. SIAM J. Discret. Math. 3(3), 411–430 (1990)
[40] Shmoys, D.: Cut problems and their applications to divide-and-conquer. In: Hochbaum, D. (ed.) Approximation Algorithms for NP-Hard Problems. PWS Publishing Company (1997)
[41] Tulsiani, M.: CSP gaps and reductions in the Lasserre hierarchy. In: 41st annual ACM symposium on Theory of computing. pp. 303–312 (2009)
[42] Vazirani, V.V.: Approximation algorithms. Springer-Verlag, Berlin (2001)
[43] de la Vega, W.F., Kenyon-Mathieu, C.: Linear programming relaxations of maxcut. In: 18th ACM-SIAM symposium on Discrete algorithms. pp. 53–61 (2007)
[44] Wainwright, M.J., Jordan, M.I.: Treewidth-based conditions for exactness of the Sherali-Adams and Lasserre relaxations. Tech. Rep. 671, University of California, Berkeley, Department of Statistics (September 2004)