Lovász-Schrijver Reformulation

Madhur Tulsiani *

September 13, 2010

Abstract

We discuss the hierarchies of linear and semidefinite programs defined by Lovász and Schrijver [33]. We describe recent progress on these hierarchies in the contexts of algorithm design, computational complexity and proof complexity.

Keywords: linear and semidefinite hierarchies, proof systems, integrality gaps.

A common technique for solving a 0-1 integer program is to work with its linear programming (LP) relaxation, and then relate the optimum of the relaxation to that of the integer program. Consider an integer program with a feasible region described by the m inequalities of the form $\mathbf{a}_i \cdot \mathbf{x} - b_i \geq 0$ in variables $x_1, \ldots, x_n \in \{0, 1\}$. In formulating a convex relaxation of this problem, one would like to optimize over the convex hull of the integral solutions to the above program (since for a linear objective function, the optimum over the integral solutions would be equal to the optimum over their convex hull). Most LP relaxations are obtained by simply relaxing the domain of the variables to be $[0, 1]$ instead of $\{0, 1\}$. However, in that case the resulting polytope may contain points which are outside this convex hull.

Starting from the works of Gomory [23] and Chvàtal [12], a large amount of research has been done on developing methods which generate additional constraints valid for the integral solutions of the problem, to obtain tighter and tighter convex relaxations. Lovász and Schrijver [33] describe a powerful such technique for generating additional inequalities. Given a program as above, let $\mathcal{P}$ denote the feasible polytope for the linear program obtained by relaxing the domain of the variables to be $[0, 1]$. Using the fact that any integer solution must satisfy $x_j^2 = x_j$ for all $j \in \{1, \ldots, n\}$, it is easy to check that for $\lambda_{ij}, \sigma_{ij} \geq 0$ and $\mu_j \in \mathbb{R}$, any quadratic inequality of the form

$$\sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} \lambda_{ij}(\mathbf{a}_i \cdot \mathbf{x} - b_i)x_j + \sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} \sigma_{ij}(\mathbf{a}_i \cdot \mathbf{x} - b_i)(1 - x_j) + \sum_{1 \leq j \leq n} \mu_j(x_j^2 - x_j) \quad \geq \quad 0 \tag{1}$$

is also valid for any integer solution to the program. The Lovász-Schrijver reformulation automatically adds any *linear* inequalities which can be derived as above, to the linear program describing $\mathcal{P}$. They also define a stronger version which adds even more constraints giving a semidefinite program (SDP).

Thinking of the strengthening as an operator applied to a polytope, we can also apply it iteratively, yielding a sequence of stronger and stronger relaxations, often referred to as *hierarchies* of linear

*`madhurt@math.ias.edu`. Work supported by NSF grant CCF-0832797.

and semidefinite programs. We describe two such hierarchies defined by Lovász and Schrijver, known as the LS (linear) and LS_+ (semidefinite) hierarchies. These hierarchies have been studied in different contexts such as algorithmic analysis, computational complexity and proof complexity; and somewhat different ways of looking at them were considered in each of these. The rest of this section is devoted to describing some of these viewpoints.

1 Description of the hierarchies

To describe these hierarchies it will be more convenient to work with *convex cones* rather than arbitrary convex subsets of $[0,1]^n$. Recall that a *cone* is a subset K of $\mathbb{R}^d$ such that if $\mathbf{x}, \mathbf{y} \in K$ and $\alpha, \beta \geq 0$ then $\alpha\mathbf{x} + \beta\mathbf{y} \in K$. If we are interested in a convex set $\mathcal{P} \subseteq [0,1]^n$ (which might be the feasible region of our starting convex relaxation), we first convert it into a cone in $\mathbb{R}^{n+1}$. We define

$$\text{cone}(\mathcal{P}) \quad = \quad \{\mathbf{y} = (\lambda, \lambda x_1, \ldots, \lambda x_n) \mid \lambda \geq 0, (x_1, \ldots, x_n) \in \mathcal{P}\}.$$

It is easy to check that all the quadratic constraints of the form described in (1) would follow if we could impose constraints of the form $x_j^2 = x_j$ for each $j \in \{1, \ldots, n\}$. However, this constraint is neither linear nor convex. We thus impose linear constraints implied by this, by introducing auxiliary variables $Y_{i,j}$ which are supposed to be equal to the product $x_i x_j$ when $\mathbf{y} = (1, x_1, \ldots, x_n)$. We require that $Y_{i,i} = y_i$. Also, if Y_i denotes the ith row of the matrix Y, then it must be true that $(Y_i/y_i)_{1\ldots n} \in \mathcal{P}$ when $y_i \neq 0$, where by $(Y_i/y_i)_{1\ldots n}$ we denote the vector consisting of the last n coordinates of (Y_i/y_i). This is equivalent to saying that $Y_i \in \text{cone}(\mathcal{P})$. Similarly, $y_i \neq y_0$ must imply $((Y_0 - Y_i)/(y_0 - y_i))_{1\ldots n} \in \mathcal{P}$ and hence $Y_0 - Y_i \in \text{cone}(\mathcal{P})$. Lovász and Schrijver use this to define two operators N and N_+, such that for a starting cone K, $N_+(K) \subseteq N(K) \subseteq K$.

Definition 1.1 *For a cone $K \subseteq \mathbb{R}^{n+1}$ we define the set $N(K)$ (also a cone in $\mathbb{R}^{n+1}$) as follows: a vector $\mathbf{y} = (y_0, \ldots, y_n) \in \mathbb{R}^{n+1}$ is in $N(K)$ iff there is a matrix $Y \in \mathbb{R}^{(n+1)\times(n+1)}$ such that*

1. *Y is symmetric.*
2. *For every $i \in \{0, 1, \ldots, n\}$, $Y_{0,i} = Y_{i,i} = y_i$.*
3. *Each row Y_i is an element of K.*
4. *Each vector $Y_0 - Y_i$ is an element of K.*

In such a case, Y is called the protection matrix of $\mathbf{y}$. If, in addition, Y is positive semidefinite, then $\mathbf{y} \in N_+(K)$. We define $N^0(K)$ and $N_+^0(K)$ as K, and $N^t(K)$ (respectively, $N_+^t(K)$) as $N(N^{t-1}(K))$ (respectively, $N_+(N_+^{t-1}(K))$).

Remark 1.2 Lovász and Schrijver also define a third operator which they denote by N_0 which is weaker than the ones described here. It is defined in a manner similar to the N and N_+ operators, except that the protection matrix Y is not required to be symmetric. However, we shall not discuss here.

If $\mathbf{y} = (\lambda, \lambda x_1, \ldots, \lambda x_n) \in \mathbb{R}^{n+1}$ for $x_1, \ldots, x_n \in \{0,1\}$, then we can set $Y_{i,j} = \lambda \cdot x_i x_j$. Such a matrix Y is clearly positive semidefinite, and it satisfies $Y_{i,i} = \lambda x_i^2 = \lambda x_i = y_i$. Also $Y_i = x_i \cdot \mathbf{y}$ and $Y_0 - Y_i = (1 - x_i) \cdot \mathbf{y}$. Hence they lie in the same cone as $\mathbf{y}$.

In the case when $\mathcal{P} \in \mathbb{R}^n$ is a polytope defined by linear constraints and $\text{cone}(\mathcal{P})$ is the corresponding cone, we will often use the notation $N^t(\mathcal{P})$ and $N_+^t(\mathcal{P})$ to denote $N^t(\text{cone}(\mathcal{P})) \cap \{y_0 = 1\}$ and

$N_+^t(\text{cone}(\mathcal{P})) \cap \{y_0 = 1\}$ respectively. The following claim characterizes $N(\mathcal{P})$ and $N_+(\mathcal{P})$ in terms of the additional inequalities imposed by the reformulation.

Claim 1.3 *Let $\mathcal{P} \in \mathbb{R}^n$ be a polytope defined by the constraints $\mathbf{a}_i \cdot \mathbf{x} - b_i \geq 0$ for $i \in \{1, \ldots, m\}$. Then for a vector $\mathbf{x} \in \mathbb{R}^n$,*

1. *$x \in N(\mathcal{P})$ iff it satisfies all linear equalities $\mathbf{c} \cdot \mathbf{x} - d \geq 0$ which can be derived as*

$$\mathbf{c} \cdot \mathbf{x} - d = \sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} \lambda_{ij}(\mathbf{a}_i \cdot \mathbf{x} - b_i)x_j + \sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} \sigma_{ij}(\mathbf{a}_i \cdot \mathbf{x} - b_i)(1 - x_j) + \sum_{1 \leq j \leq n} \mu_j(x_j^2 - x_j)$$

for $\lambda_{ij}, \sigma_{ij} \geq 0$ and $\mu_j \in \mathbb{R}$.

2. *$x \in N_+(\mathcal{P})$ iff it satisfies all linear equalities $\mathbf{c} \cdot \mathbf{x} - d \geq 0$ which can be derived as*

$$\begin{aligned} \mathbf{c} \cdot \mathbf{x} - d = & \sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} \lambda_{ij}(\mathbf{a}_i \cdot \mathbf{x} - b_i)x_j + \sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} \sigma_{ij}(\mathbf{a}_i \cdot \mathbf{x} - b_i)(1 - x_j) + \sum_{1 \leq j \leq n} \mu_j(x_j^2 - x_j) \\ & + \sum_k (\mathbf{g}_k \cdot \mathbf{x} - h_k)^2 \end{aligned}$$

for $\lambda_{ij}, \sigma_{ij} \geq 0$, $\mu_j, h_k \in \mathbb{R}$ and $\mathbf{g}_k \in \mathbb{R}^n$.

The characterization in terms of the inequalities is particularly useful when reasoning about these hierarchies as proof systems for (say) proving that a certain integer program has no integral solutions. The rules for deriving the inequalities characterizing $N(\mathcal{P})$ from those characterizing $\mathcal{P}$ can then be thought of as the derivation rules of this proof system. One is then interested in the smallest t such that $N^t(\mathcal{P})$ is empty (called the *rank* of the proof system) and in the minimum number of inequalities needed (called the *size* of the proof) to derive a convex system of inequalities that has no feasible solution. We discuss two different ways of viewing Definition 1.1 below.

1.1 The prover-adversary game

It is sometimes convenient to think of the LS and LS$_+$ hierarchies in terms of a game between two parties called a *prover* and an *adversary*. This formulation was first used by Buresh-Oppenheim et al. [9] who used it to prove lower bounds on the LS$_+$ procedure as a proof system. For example, the following is a formulation that is convenient to use for proving that a certain vector belongs to $N_+^t(K)$ for a cone $K \subseteq \mathbb{R}^{n+1}$. The formulation remains the same for talking about $N(K)$, except that we omit the positive semidefinite-ness constraint in the matrix below.

A *prover* is an algorithm that, on an input vector $(y_0, \ldots, y_n)$, either fails or outputs a matrix $Y \in \mathbb{R}^{(n+1)\times(n+1)}$ and a set of vectors $O \subseteq \mathbb{R}^{n+1}$ such that

1. Y is symmetric and positive semidefinite.
2. $Y_{i,i} = Y_{0,i} = y_i$.
3. Each vector Y_i and $Y_0 - Y_i$ is a non-negative linear combination of vectors of O.
4. Each element of O is in K.

Consider now the following game played by a prover against another party called the *adversary*. We start from a vector $\mathbf{y} = (y_0, \ldots, y_n)$, and the prover, on input $\mathbf{y}$, outputs Y and O as before. Then the adversary chooses a vector $\mathbf{z} \in O$, and the prover, on input $\mathbf{z}$, outputs a matrix Y' and a set O', and so on. The adversary *wins* when the prover fails to output a matrix and vectors with the required properties.

Lemma 1.4 *Suppose that there is a prover such that, starting from a vector $\mathbf{y} \in K$, every adversary strategy requires at least $t+1$ moves to win. Then $\mathbf{y} \in N^t(K)$.*

Proof: We proceed by induction on t, with $t = 0$ being the simple base case. Suppose that, for every adversary, it takes at least $t+1$ moves to win, and let Y and O be the output of the prover on input $\mathbf{y}$. Then, for every element $\mathbf{z} \in O$, and every prover strategy, it takes at least t moves to win starting from $\mathbf{z}$. By the inductive hypothesis, each element of O is in $N_+^{t-1}(K)$, and since $N_+^{t-1}(K)$ is closed under non-negative linear combinations, the vectors Y_i and $Y_0 - Y_i$ are all in $N_+^{t-1}(K)$, and so Y is a protection matrix that shows that $\mathbf{y}$ is in $N_+^t(K)$. ■

While the framework is simply a restatement of the conditions defining $N(K)$ and $N_+(K)$, it turns out to be surprisingly convenient to think of many arguments in terms of strategies for this game. Also, note that we added the set O in the description of the conditions. In many proofs, it is convenient to think of O as a set of vectors satisfying some structural invariant. Often it may be easier to exhibit matrices Y for vectors satisfying some invariants, than for arbitrary vectors in K.

1.2 The view in terms of local probability distributions

Yet another view of the conditions which has been useful in reasoning about optimization problems is the interpretation of solutions as "locally conditioned probability distributions". Let $\mathbf{x} \in \mathcal{P}$ be expressible as $\mathbf{x} = \sum_i \lambda_i \mathbf{z}^{(i)}$ where $\sum_i \lambda_i = 1$ and $\forall i.\ \ \mathbf{z}^{(i)} \in \mathcal{P} \cap \{0,1\}^n,\ \ \lambda_i \geq 0$. Then, we can consider a random variable $\mathbf{z}$ which takes value $\mathbf{z}^{(i)}$ with probability λ_i. For $j \in \{1, \ldots, n\}$, the numbers x_j then are equal to $\mathbf{P}[z_j = 1]$ i.e. the marginals of this distribution.

To prove that $\mathbf{x} \in \mathcal{P}$, we then require a matrix $Y \in \mathbb{R}^{n+1}$ which satisfies the conditions stated in section 1.1. For each $\mathbf{z}^{(i)}$ such a matrix $Y^{(i)}$ can be given as $Y^{(i)} = (1, \mathbf{z}^{(i)})(1, \mathbf{z}^{(i)})^T$ where $(1, \mathbf{z}^{(i)}) \in \mathbb{R}^{n+1}$. The matrix Y for $\mathbf{x}$ can then be exhibited as $Y = \sum_i \lambda_i Y^{(i)}$, where each entry $Y_{ij} = \mathbf{P}[(z_i = 1) \wedge (z_j = 1)]$. Arguing that the vector $Y_i \in \text{cone}(\mathcal{P})$ is then equivalent to arguing that the vector $\mathbf{x}^{(i,1)} \in \mathbb{R}^n$ with

$$x_j^{(i,1)} \;=\; Y_{ij}/x_i \;=\; \mathbf{P}[z_j = 1 \mid z_i = 1]$$

is in $\mathcal{P}$. Similarly, $Y_0 - Y_i \in \text{cone}(\mathcal{P})$ is equivalent to proving that the vector $\mathbf{x}^{(i,0)}$ with coordinates $x_j^{(i,0)} = (Y_{0j} - Y_{ij})/(1 - x_i) \;=\; \mathbf{P}[z_j = 1 | z_i = 0]$, is in $\mathcal{P}$.

Thus, the conditions for membership in $N(\mathcal{P})$ and $N_+(\mathcal{P})$ can be thought of as conditions for checking that a vector $\mathbf{x}$ indeed represents the marginals of a distribution over $\mathcal{P} \cap \{0,1\}^n$. The test is the existence of conditional distributions, conditioned on each coordinate i being either 0 or 1. Similarly, exhibiting membership in $N^t(\mathcal{P})$ then corresponds to exhibiting distributions after conditioning on any t variables instead of just one.

Combining this with the prover-adversary framework, we can think that at each step, the adversary conditions a particular variable to be 0 or 1. The task of the prover is then to provide a matrix corresponding to the conditional distribution. We must emphasize that this is *not an exact* *characterization* of the hierarchies, but this intuition has been quite useful in arguing about their performance on optimization problems, as discussed in Section 3.

1.3 Relation to Sherali-Adams and Lasserre hierarchies

Hierarchies of relaxations similar to the ones by Lovász and Schrijver were also defined by Sherali and Adams [39] and Lasserre [29]. For a starting linear program with feasible region $\mathcal{P}$, let $SA^t(\mathcal{P})$ and $Las^t(\mathcal{P})$ denote the feasible regions of the relaxations obtained at the tth level of these hierarchies. The set $SA^t(\mathcal{P})$ can be described as the projection of the feasible region of a linear program in $n^{O(t)}$ variables, while $Las^t(\mathcal{P})$ is the projection of the feasible set for a semidefinite program in $n^{O(t)}$ dimensions.

It is also known that $SA^t(\mathcal{P}) \subseteq N^t(\mathcal{P})$ and $Las^t(\mathcal{P}) \subseteq N_+^t(\mathcal{P})$ and thus the Sherali-Adams and Lasserre hierarchies provide tighter relaxations, respectively, than the LS and LS_+ hierarchies. Figure 1 shows the relationships between these different hierarchies, with an arrow $A \to B$ indicating that $A \supseteq B$ and thus B is tighter than A. A more detailed comparison can be found in the excellent survey by Laurent [30].

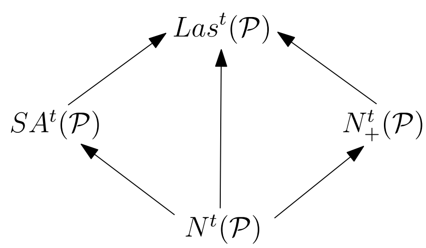

Figure 1: A comparison of the hierarchies

2 Algorithmic aspects and applications

We first note that if K is defined by a linear program with m constraints in the $n+1$ variables $y_0, \ldots, y_n$, then $N(K)$ is defined by a linear program with $O(mn)$ constraints in the $O(n^2)$ variables corresponding to the entries of the matrix Y. Similarly, $N_+(K)$ is defined by a semidefinite program in Y with $O(mn)$ constraints. More generally, Lovàsz and Schrijver show the following.

Theorem 2.1 *Given a weak separation oracle for the cone K, the weak separation problem for $N(K)$ and $N_+(K)$ can be solved in polynomial time.*

Hence, whenever it is possible to easily optimize over the cone K, one can also optimize over the cone $N(K)$ (or $N_+(K)$) with a polynomial overhead. Using the above, it is also easy to deduce that weak separation oracles for $N^t(K)$ and $N_+^t(K)$ can be implemented in time $n^{O(t)}$, given one for K running in polynomial time. For a convex region $\mathcal{P} \subseteq \mathbb{R}^n$, let $K_I \subseteq \mathbb{R}^{n+1}$ denote the cone generated by the integral vectors $\{(1, \mathbf{x}) \mid \mathbf{x} \in \mathcal{P} \cap \{0,1\}^n\}$ and let $K = \text{cone}(\mathcal{P})$. Then it is known that n applications of $N(\cdot)$ are sufficient to converge to K_I.

Theorem 2.2 $N^n(K) = N_+^n(K) = K_I$.

However, note that n applications take time $n^{O(n)}$, which is super-exponential in n. For general cones, this bound is known to be tight for both N and N_+ operators, as was shown by Cook and Dash [15] and also by Goemans and Tunçel [21]. Typically, the interesting question is if for a given problem, optimizing over $N^t(K)$ or $N_+^t(K)$ for a small t, can give some non-trivial approximation guarantee.

2.1 Application to Independent Set

As an example, we consider the constraints obtained by applying the $N(\cdot)$ and $N_+(\cdot)$ operators to the feasible set of the LP relaxation for Maximum Independent Set; which is the problem of finding the largest subset of vertices in a graph not containing an edge (also known as the Maximum Stable Set problem). Let $G = (V, E)$ be a graph on n vertices. Then the linear programming relaxation for the Maximum Independent Set problem has variables $x_1, \ldots, x_n \in [0, 1]$ and constraints $x_i + x_j \leq 1$ for all $(i, j) \in E$. We denote the feasible region of these constraints as $IS(G)$. The objective is to maximize the sum $\sum_i x_i$.

The set cone($IS(G)$) is then given by the inequalities $0 \leq y_i \leq y_0$ for all $i \in \{1, \ldots, n\}$ and $y_i + y_j \leq y_0$ for all $(i, j) \in E$. We will be interested in the sets given by $N(IS(G))$ and $N_+(IS(G))$.

There are also various additional constraints that are satisfied by an integer solution to the independent set problem. Lovász and Schrijver consider the so called "odd-cycle constraints" and "clique constraints" and show that they are implied by one application of $N(\cdot)$ and $N_+(\cdot)$ respectively. The former type of constraints say that if C is a set of vertices forming a cycle of odd length, then $\sum_{i \in C} x_i \leq (|C| - 1)/2$. The latter require that if B is a set of vertices forming a clique, then $\sum_{i \in B} x_i \leq 1$. We give a proof below for the sake of illustration.

Claim 2.3 *For a graph $G = (V, E)$, let $\mathbf{x} = (x_1, \ldots, x_n)$ be a vector in $IS(G)$. Then*

1. *If $\mathbf{x} \in N(IS(G))$, then for every odd cycle C in G, $\sum_{i \in C} x_i \leq (|C| - 1)/2$.*
2. *If $\mathbf{x} \in N_+(IS(G))$, then for every clique B in G, $\sum_{i \in B} x_i \leq 1$.*

Proof: For this part, it is convenient to think of $N(IS(G))$ in terms of the rules for deriving new inequalities by a combination of old ones, as given in Claim 1.3. Let $C = \{i_1, \ldots, i_k\}$ be a cycle of odd length in the graph. Since $i_1, \ldots, i_{k-1}$ can be covered by $(k-1)/2$ edges, adding their inequalities we have $x_{i_1} + \ldots + x_{i_{k-1}} \leq (k-1)/2$. Similarly, $x_{i_2} + \ldots + x_{i_{k-2}} \leq (k-3)/2$. Using these and Claim 1.3, we get that $(x_1, \ldots, x_n)$ must satisfy

$$\left(\sum_{j=1}^{k-1} x_{i_j} - \frac{(k-1)}{2}\right)(1 - x_{i_k}) + \left[\left(\sum_{j=2}^{k-2} x_{i_j} - \frac{(k-3)}{2}\right) + (x_{i_1} + x_{i_k} - 1) + (x_{i_{k-1}} + x_{i_k} - 1)\right] x_{i_k} - 2(x_{i_k}^2 - x_{i_k}) \quad \leq \quad 0$$

Simplifying the above yields $\sum_{j=1}^{k} x_{i_j} \leq (k-1)/2$.

For the second part, it will be more convenient to think of $\mathbf{x} \in N_+(IS(G))$ in terms of the protection matrix Y certifying that $\mathbf{y} = (1, x_1, \ldots, x_n) \in N(\text{cone}(IS(G)))$, as in Definition 1.1. Since the matrix must be positive semidefinite, we consider its Gram decomposition, say given by the vectors $\mathbf{v}_0, \ldots, \mathbf{v}_n$. The conditions $Y_{i,0} = Y_{i,i} = y_i$ imply that $|\mathbf{v}_0|^2 = 1$ and $\langle \mathbf{v}_0, \mathbf{v}_i \rangle = |\mathbf{v}_i|^2 = x_i$.

If (i, j) is an edge in the graph, then since Y_i must be in the cone $IS(G)$, we get $Y_{i,i} + Y_{i,j} \leq Y_{i,0}$ and hence $Y_{i,j} \leq 0$. However, since a vector in the cone must have non-negative entries, we must have $Y_{i,j} = \langle \mathbf{v}_i, \mathbf{v}_j \rangle = 0$ for any edge (i,j). Thus, all the vectors corresponding to vertices in a clique must be orthogonal. By Pythagoras' theorem,

$$\sum_{i \in B, |\mathbf{v}_i| > 0} \frac{\langle \mathbf{v}_0, \mathbf{v}_i \rangle^2}{|\mathbf{v}_i|^2} \le |\mathbf{v}_0|^2 \implies \sum_{i \in B, x_i > 0} \frac{x_i^2}{x_i} \le 1 \implies \sum_{i \in B} x_i \le 1.$$

■

The graphs for which the clique constraints exactly characterize the convex hull of 0/1 independent sets, are called *perfect* graphs. Hence, the above shows that optimizing over $N_+(IS(G))$, finds exactly the size of the maximum independent set for a perfect graph G. Similarly $N(IS(G))$ finds the maximum independent set in graphs for which the odd-cycle constraints characterize the 0/1 independent sets. Such graphs are known as *h-perfect*.

2.2 Deriving the SDP relaxation for Sparsest Cut

Alekhnovich, Arora and Tourlakis [1] sketch how the well known semidefinite programming relaxation for Sparsest Cut by Arora, Rao and Vazirani [5] can be derived by three levels of the LS_+ hierarchy starting from a basic linear program. We give a proof below showing that in fact, one level of the LS_+ hierarchy already derives the relaxation in [5].

Sparsest Cut is the problem of finding a cut $(S, \bar{S})$ minimizing $|E(S, \bar{S})|/|S|$, the ratio of the number of outgoing edges to the size of S. In the discussion below, we instead describe the argument for the c-Balanced Separator problem, which is known to be closely related to Sparsest Cut (see [32, 31, 5] for details). Instead of asking for the sparsest cut of any size, the c-Balanced Separator problem asks for the sparsest cut with at least cn vertices on each side of the cut. For a graph $G = (V, E)$ on n vertices, the linear program involves $n + \binom{n}{2}$ variables: $x_i \in [0,1]$ for each vertex i, representing which side of the cut it is, and $d_{ij} \in [0,1]$ for every pair $\{i, j\}$ of vertices representing the distance between them (which in an integral solution is 0 if they are on the same side and 1 otherwise).

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{(i,j)\in E} d_{ij} & \\
\text{subject to} & d_{ij} \le x_i + x_j & \forall\ \{i,j\} \\
& d_{ij} \le (1 - x_i) + (1 - x_j) & \forall\ \{i,j\} \\
& d_{ij} \ge x_i - x_j & \forall\ \{i,j\} \\
& d_{ij} \ge x_j - x_i & \forall\ \{i,j\} \\
& \displaystyle\sum_{\{i,j\}\in\binom{V}{2}} d_{ij} \ge c(1-c)n^2 & \\
& x_i, d_{ij} \in [0,1] &
\end{array}$$

Figure 2: LP relaxation for Balanced Separator

The constraint $\sum d_{ij} \ge c(1-c)n^2$ imposes the condition (on integer solutions) that the number of vertices on each side of the cut is at least cn. Note that the above program is quite weak (in fact its integrality gap, as defined in Section 3 is infinite). This can be seen by using the solution

$x_i = 1/2$ for all i and $d_{ij} = 0$ when $(i, j) \in E$ and $d_{ij} = 1$ otherwise. This is a feasible solution with objective value 0 for any graph in which $|E| \leq \binom{n}{2} - c(1-c)n^2$, while the integer optimum may be arbitrarily large. Adding the inequality $d_{ij} + d_{jk} \geq d_{ik}$ for every triple i, j, k makes it at least as strong as the linear program used by Leighton and Rao [32, 31] which was shown to achieve an $O(\log n)$ approximation. However, even without this added inequality, one application of the N_+ operator derives the relaxation in [5] which achieves an $O(\sqrt{\log n})$ approximation.

Arora, Rao and Vazirani [5] use a semidefinite program with $n \times n$ semidefinite matrix Y whose Gram decomposition is given by the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$. A solution to their program would be a feasible solution to the above LP if we take $x_i = |\mathbf{v}_i|^2$ and $d_{ij} = |\mathbf{v}_i - \mathbf{v}_j|^2$. They also impose the additional inequality that $|\mathbf{v}_i - \mathbf{v}_j|^2 + |\mathbf{v}_j - \mathbf{v}_k|^2 \geq |\mathbf{v}_i - \mathbf{v}_k|^2$. We argue below that this inequality is derived by one application of N_+.

Claim 2.4 *Let $(1, \mathbf{x}, \mathbf{d}) \in N_+(cone(BS(G)))$ and let Y be its protection matrix with the Gram decomposition $\left\{\mathbf{v}_0, \{\mathbf{v}_i\}_{i \in V}, \{\mathbf{v}_{ij}\}_{\{i,j\} \in \binom{V}{2}}\right\}$. Then the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ satisfy for all i, j, k*

$$d_{ij} = |\mathbf{v}_i - \mathbf{v}_j|^2 \qquad \textit{and} \qquad |\mathbf{v}_i - \mathbf{v}_j|^2 + |\mathbf{v}_j - \mathbf{v}_k|^2 \geq |\mathbf{v}_i - \mathbf{v}_k|^2$$

Proof: If $x_1, \ldots, x_n$ were indeed an integer solution, we would have $d_{ij} = (x_i - x_j)^2$, which is 1, when exactly one of x_i, x_j is 1. To derive this for an integer solution, we use $d_{ij} \leq (1-x_i) + (1-x_j)$ and $d_{ij} \leq x_i + x_j$ to get $d_{ij} x_j \leq (2 - x_i - x_j) x_j$ and $d_{ij}(1 - x_j) \leq (x_i + x_j)(1 - x_j)$. Adding these two, and using $x_i^2 = x_i$ and $x_j^2 = x_j$ gives $d_{ij} \leq (x_i - x_j)^2$. Similarly, starting with the inequalities $d_{ij} \geq x_i - x_j$ and $d_{ij} \geq x_j - x_i$, we can get $d_{ij} \geq (x_i - x_j)^2$.

The proof essentially simulates the "linearized" version of this reasoning, replacing a quadratic term $x_i x_j$ by the entry $Y_{i,j}$ of the matrix. We also use $Y_{i,jk}$ to refer to the entry of the matrix corresponding to variables x_i and d_{jk}. Since $Y_j \in \text{cone}(BS(G))$ and $Y_0 - Y_j \in \text{cone}(BS(G))$, we get

$$Y_{j,ij} \leq 2Y_{j,0} - Y_{j,i} - Y_{j,j} \quad \text{and} \quad Y_{0,ij} - Y_{j,ij} \leq (Y_{0,i} - Y_{j,i}) + (Y_{0,j} - Y_{j,j}).$$

Again, adding the two and using $Y_{i,i} = Y_{0,i}$ and $Y_{j,j} = Y_{0,j}$ gives

$$d_{ij} \;=\; Y_{0,ij} \;\leq\; Y_{i,i} + Y_{j,j} - 2Y_{i,j} \;=\; |\mathbf{v}_i - \mathbf{v}_j|^2 .$$

The proof of the inequality in the other direction is similar.

For the second part, we again note for an integer solution, $d_{ij} = (x_i - x_j)^2 = x_i(1-x_j) + x_j(1-x_i)$. Hence, it would suffice to prove $x_j[(1 - x_i) + (1 - x_k)] + (1 - x_j)[x_i + x_k] \geq d_{ik}$. However, this follows by using $x_j[(1 - x_i) + (1 - x_k) - d_{ik}] \geq 0$ and $(1 - x_j)[x_i + x_k - d_{ik}] \geq 0$. For the matrix version, we again use $Y_j, Y_0 - Y_j \in \text{cone}(BS(G))$ to get

$$2Y_{j,0} - Y_{j,i} - Y_{j,k} \geq Y_{j,ik} \quad \text{and} \quad (Y_{0,i} - Y_{j,i}) + (Y_{0,k} - Y_{j,k}) \geq (Y_{0,ik} - Y_{j,ik}).$$

Adding the two and replacing every entry $Y_{i,j}$ by $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$ gives $|\mathbf{v}_i - \mathbf{v}_j|^2 + |\mathbf{v}_j - \mathbf{v}_k|^2 \geq |\mathbf{v}_i - \mathbf{v}_k|^2$. ■

2.3 Finding dense subgraphs in a graph

Both the above arguments start with a polytope $\mathcal{P}$ and only consider $N(\mathcal{P})$ and $N_+(\mathcal{P})$. Tighter relaxations which are still efficient, can be obtained by considering $N^t(\mathcal{P})$ and $N_+^t(\mathcal{P})$, where $t = O(1)$ or even when $t = \omega(1)$ is a slowly growing function of n. However, not many arguments are known which use these, as the sets $N^t(\mathcal{P})$ or $N_+^t(\mathcal{P})$ are often difficult to understand for large t.

Very recently, an application for the Densest k-Subgraph problem was obtained by Bhaskara et al. [8]. Given a graph G and a size parameter k, the problem requires finding a subgraph of k vertices in G, with the maximum number of edges. The previously best known approximation factor for this problem was $O(n^{0.3159})$ (see [17], [22]). The authors achieve an approximation factor of $O(n^{1/4+\epsilon})$ using $N^{O(1/\epsilon)}(\cdot)$.

Starting with a basic linear program, they strengthen it to a linear program implied by $O(1/\epsilon)$ applications of $N(\cdot)$. The strengthening uses intuition of the solution as a distribution over dense subgraphs. The constraints require that when one conditions $O(1/\epsilon)$ vertices to be in the subgraph, they all must have large expected degree according to the conditional distribution. The full algorithm, however, is significantly more involved and uses various other combinatorial arguments in addition to the LP.

3 Hardness results for optimization problems

As discussed in the algorithmic applications, for hard optimization problems, one is often interested in how well the relaxation obtained by $N^t(\mathcal{P})$ approximates the value of the integer program. For a maximization problem, if FRAC denotes the value of a relaxation, and OPT denotes the combinatorial optimum for the optimization problem (i.e. the maximum objective value for an integer solution), then we define the *integrality gap* of the relaxation as the *supremum* of the ratio FRAC/OPT over all instances of the problem. For a minimization problem, we take it to be the inverse ratio. When the integrality gap varies with the size of the problem, we take the supremum over all instances of size n and express it as a function of n. Note that according to our definition, the integrality gap is always larger than 1, and the closer it is to 1, the better the performance of the convex relaxation.

From the perspective of complexity theory, one can view the hierarchies of programs defined by the N and N_+ operators as restricted models of computation, with the number of applications of these operators as a resource. This also corresponds naturally to time of computation in this model as optimizing over $N^t(\mathcal{P})$ takes time $n^{O(t)}$. One is then interested in the tradeoff between the level of the relaxation in the hierarchy (denoted by t) and the integrality gap of the relaxation. A lower bound in this computational model corresponds to showing that the integrality gap remains large even for the relaxations obtained at very high levels of the hierarchy.

3.1 Integrality gaps for Vertex Cover

Minimum Vertex Cover, the problem of finding the smallest subset of vertices in a graph such that every edge is incident to some vertex in the subset, is perhaps the most studied problem with regard to integrality gaps in the Lovász-Schrijver hierarchy. The study of integrality gaps in this model was initiated by the works of Arora, Bollobás, Lovász and Tourlakis [3, 4]. They showed that the integrality gap remains at least $2-\epsilon$ even after $\Omega(\log n)$ levels of the hierarchy. Since the integrality gap can be easily shown to be *at most* 2 even for the starting linear relaxation, this showed that even using time $n^{O(\log n)}$ in this model yields no significant improvement. The results were later improved to 3/2 for $\Omega(\log^2 n)$ levels by Tourlakis [40] and to an optimal lower bound of $2-\epsilon$ for $\Omega(n)$ levels by Schoenebeck, Trevisan and Tulsiani [38]. Note that the last result even rules out exponential time algorithms (note that $\Omega(n/\log n)$ levels would correspond to $2^{\Omega(n)}$ time) in this model.

These results work implicitly or explicitly with the "locally conditioned distributions" interpretation discussed in Section 1.2. One considers sparse random graphs which have no cycles of size less than $\Omega(\log n)$ so that any subgraph with $O(\log n)$ vertices is a tree. One then starts with a solution which has fractional value $1/2+\epsilon$ on every vertex. The move of the adversary then corresponds to selecting a vertex where the solution has a fractional value, and fixing it to 1 or 0 i.e. conditioning it to be in or out of the vertex cover. As long as the adversary conditions on $O(\log n)$ vertices, the set of conditioned vertices form a tree, restricted to which there is an *actual distribution of vertex covers* with marginal values $1/2+\epsilon$. Hence, one can use this to produce the required conditional distribution. We remark that this is just an intuition for the proof in [4]. The actual proof proceeds by looking at the duals of the linear programs obtained by the LS hierarchy and involves significantly more work. The result in [38] for $\Omega(n)$ levels uses an explicit version of the above intuitive argument together with a more careful use of the sparsity of these graphs.

Integrality gaps for the semidefinite version of this hierarchy have been somewhat harder to prove. It was shown by Goemans and Kleinberg [26] that the integrality gap for the relaxation obtained by one application of N_+ (which is equivalent to the ϑ-function of Lovász) is at least $2-\epsilon$. The result was strengthened by Charikar [10], who showed that the same gap holds even when the relaxation is augmented with "triangle inequalities" discussed in Section 2.2. The gap was extended to $\Omega(\sqrt{\log n/\log\log n})$ levels of the LS_+ hierarchy by Georgiou et al. [20]. Interestingly, all the these results for LS_+ were proven for the same family of graphs, inspired by a paper of Frankl and Rödl [19]. It is an interesting problem to construct an alternate family which is also an integrality gap instance, or to extend the above results to even $\Omega(\log n)$ levels.

Somewhat incomparable to the above results, integrality gaps of factors 7/6 for $\Omega(n)$ levels and 1.36 for $n^{\Omega(1)}$ levels of the semidefinite hierarchy were obtained by Schoenebeck et al. [37] and Tulsiani [41]. These results also differ from the ones discussed above in that they do not directly exhibit a family of integrality gap instances for vertex cover. Instead, they start with an integrality gap instance for a constraint satisfaction problem, and proceed by using a reduction from the constraint satisfaction problem to vertex cover.

3.2 Results for Constraint Satisfaction Problems

For constraint satisfaction problems with 3 or more variables in each constraint, strong integrality gaps have been shown for the Lovász-Schrijver semidefinite (and hence also the linear) hierarchy. For these problems one studies how well convex relaxations approximate the maximum number of constraints that can be satisfied by any assignment to the variables.

Optimal integrality gaps of a factor $2^k/(2^k-1)$ for $\mathsf{MAX\ k-SAT}$ with $k \geq 5$ and $\Omega(n)$ levels were shown by Buresh-Oppenheim et al. [9]. These were later extended to the important remaining case of $\mathsf{MAX\ 3-SAT}$ by Alekhnovich et al. [1], who also proved strong lower bounds for approximating Minimum Vertex Cover in hypergraphs. $\Omega(n)$ level integrality gaps for other constraint satisfaction problems also follow from the corresponding results in the Lasserre hierarchy by Schoenebeck [36] and Tulsiani [41]. All these results crucially rely on the ideas of expansion from proof complexity, explained in Section 4.

3.3 Results for other problems

The performance of the LS hierarchy for the Traveling Salesman Problem was considered by Cook and Dash [15] and by Cheung [11]. A basic LP relaxation for the problem with n cities involves $n(n-1)/2$ variables and is due to Dantzig, Fulkerson and Johnson. This relaxation is known to have integrality gap at least $4/3$. Cook and Dash showed that exactly characterizing the integer optimum requires at least $\lceil n/8 \rceil$ levels of the LS_+ hierarchy. They also exhibited an inequality valid for the integer solutions, such that any LS_+-derivation of it requires to use at least $2^{n/8}/n^4$ intermediate inequalities. Cheung showed that that the integrality gap of the relaxation remains at least $4/3$ after any constant number of levels in the LS hierarchy.

Feige and Krauthgamer [18] studied how well the programs in the Lovász-Schrijver hierarchy do in determining the size of the largest clique in random graphs. Let $G_{n,1/2}$ denote the probabilistic model for generating a random graph by picking each pair of vertices to be an edge independently with probability $1/2$. They were motivated by the the problem of distinguishing a random graph in $G_{n,1/2}$ from one in which a clique of size k ($\approx \sqrt{n}$) has been planted, which has been suggested as the basis of a cryptographic scheme [25, 28].

They showed that the value of the relaxation for Maximum Clique obtained by N_+^t is almost surely between $\sqrt{n/(2+\epsilon)^{t+1}}$ and $4\sqrt{n/(2-\epsilon)^{t+1}}$ for any $\epsilon \in (0,2)$ and $t = o(\log n)$. Hence, their result shows that one can use $N_+^t(\cdot)$ to find planted cliques of size greater than $4\sqrt{n/(2-\epsilon)^t}$, but not much smaller than that.

Also, it is well known (see for example [2]) that the size of the largest clique in a random graph generated according to $G_{n,1/2}$ is almost surely equal to $(2+o(1))\log_2 n$. It was shown by Lovász and Schrijver [33] that if largest clique in G has size $\omega(G)$, then the value of the relaxation obtained by $t = \omega(G)$ levels of the LS_+ hierarchy is exactly $\omega(G)$. Hence, while the relaxation obtained by $(2+o(1))\log n$ levels of the hierarchy almost surely is tight, any relaxation obtained by $t = o(\log n)$ levels has a large integrality gap.

4 Proof Complexity

Proof complexity studies the efficiency with which the proof of a given proposition can be derived by a proof system. As discussed earlier, the Lovász-Schrijver hierarchies can be viewed as a proof system, where one starts with a given set of inequalities defining a convex region and derives new ones using the rules in Claim 1.3. This proof system is known to be at least as powerful as the well known Resolution proof system [14].

Consider a formula φ in conjunctive normal form (CNF), the feasibility of which can be phrased as an integer program in n variables. If $\mathcal{P}$ is the feasible polytope for an LP relaxation of this program, and if φ is unsatisfiable, then we must have $\mathcal{P} \cap \{0,1\}^n = \emptyset$. Hence, t applications of $N(\cdot)$ will *prove* that φ is unsatisfiable, if $N^t(\mathcal{P}) = \emptyset$. The smallest t for which $N^t(\mathcal{P})$ (or $N_+^t(\mathcal{P}) = \emptyset$) is known as the N-rank (resp. N_+-rank) of φ. For proving a lower bound on proof complexity, one is then interested in exhibiting a family of formulas with large (typically $\Omega(n)$) rank.

The techniques developed for proving lower bounds in proof complexity have been also useful in reasoning about general constraint satisfaction problems, and in proving hardness of approximating various optimization problems (using convex relaxations), as discussed in Section 3.

4.1 Expansion and rank lower bounds for formulas

An important technique for proving lower bounds in proof complexity has been the use of *expansion* in formulas. It was used for proving that if φ is an unsatisfiable 3-CNF formula in n variables, with each set of s clauses (for say $s \leq n/1000$) involves at least (say) $3s/2$ variables, then any proof of the unsatisfiability of φ in the *resolution* proof system, must have exponential size (see [7], [6]).

Rank lower bounds for LS-proof systems were first proved by Grigoriev et al. [24]. Buresh-Oppenheim et al. [9] were the first to use expansion arguments in the context of Lovász-Schrijver proof systems. They also proved lower bounds for various optimization problems which were later extended by Alekhnovich et al. [1]. Both these papers phrase their arguments in terms of the prover-adversary game discussed in Section 1.1.

In their argument, they start with a vector with a fractional value for each variable and prove that the vector is in $N_+^t(\mathcal{P})$ for $t = \Omega(n)$. A move of the adversary in their setting amounts to fixing one of the fractional variables to 1 or 0 (i.e. true or false) and the prover is supposed to provide a fractional assignment consistent with the fixing which is still in the polytope of feasible solutions. They then obtain the set O by fixing *additional* variables such that the desired fractional vector is a convex combination of the assignments in O. The additional variables are fixed to maintain the invariant that if one considers the formula restricted only to the variables which have not been fixed to 0 or 1, then the formula is still expanding (in the sense that a set of s clauses will contain at least $3s/2$ unfixed variables). Expansion essentially means even when $O(n)$ variables are fixed, most clauses still have a large number of unfixed variables, whose value can be modified suitably to satisfy the constraints of the convex program.

4.2 Lower bounds on size of proofs

A finer measure than the *rank* of a proof that φ is unsatisfiable, is the *size* of the proof. For the Lovász-Schrijver proof system, where one proves unsatisfiability by progressively deriving new inequalities until one gets an obvious contradiction like $0 = 1$, the size may be thought of as the *number* of inequalities one needs to derive before one can obtain a contradiction. A lower bound of t on the rank implies that one will need to use at least $n^{\Omega(t)}$ inequalities, if one uses *all the inequalities describing* $N^t(\mathcal{P})$. On the other hand, a size lower bound of $n^{\Omega(t)}$ would rule out *any way* of using $n^{O(t)}$ inequalities to prove unsatisfiability. We remark that a more accurate definition would involve the sum of the sizes of the coefficients in all the inequalities used. However, the lower bounds described here also hold for the definition involving number of inequalities.

Dash [16] provided exponential size lower bounds for version of the Lovász-Schrijver proof systems, which uses an operator weaker than the N and N_+ operators. For "cutting-plane" proof systems based on the cut procedures of Gomory and Chvàtal [23, 12], exponential lower bounds on the size of proofs were proved by Pudlák [35]. Kojevnikov and Itsykson [27] later proved exponential lower bounds for "tree-like" proofs in the Lovász-Schrijver proof system, using expansion-based arguments. One says that a given proof is tree-like when each derived inequality is used in at most once for another derivation i.e. the derived inequalities form a tree, where the parent is derived using the children.

Pitassi and Segerlind [34] later derived a general tradeoff between the rank and size of tree-like proofs in the LS and LS_+ proof systems. They proved that if such a proof has size S and rank t, then one must have $t = \Omega(\sqrt{n \log S})$. Using a rank lower bound of $t = \Omega(n)$, this implies a lower bound of $2^{\Omega(n)}$ on S. However, they also showed that this tradeoff did not hold for general (non-tree like) proofs in these proof systems.

Size lower bounds for general proofs in the LS and LS_+ can now be derived via an indirect route, by combining the the results of Pitassi and Segerlind with those of Schoenebeck [36]. Although the tradeoff of Pitassi and Segerlind did not work for general proofs in the LS-proof system, they showed (extending an argument of [13]) that it *does* hold for general proofs in the proof systems corresponding to the Sherali-Adams and Lasserre hierarchies. This means that an $\Omega(n)$ lower bound on the rank of Lasserre proofs would imply an exponential lower bound on the size in the Lasserre proof system; and hence also on the size of *general* proofs in the LS and LS_+ proofs systems (each LS and LS_+ proof can be simulated by a Lasserre proof of the same size). Such rank lower bounds for Lasserre proofs were provided by Schoenebeck [36], thus also proving the required size bounds.

5 Conclusion

The hierarchies of Lovász and Schrijver are an interesting computational model, and can be seen to capture many known approximation algorithms within the first few levels. Perhaps the most interesting problem about these is finding algorithmic applications of programs at the higher levels of these hierarchies. For problems such as Sparsest Cut and Approximate Graph Coloring, the known lower bounds are far from tight, and it is an intriguing possibility that programs at higher levels can provide a better approximation guarantee for these problems.

In the context of lower bounds, it still remains to settle the integrality gap for the much investigated Minimum Vertex Cover problem in the LS_+ hierarchy. Also, it is interesting to investigate the integrality gaps for other optimization problems such as Sparsest Cut and Unique Games.

Acknowledgements

I thank Sanjeev Arora and Pavel Hrubes for helpful comments and references. I also thank the anonymous referees for suggesting various corrections and improvements to the previous version of this manuscript, and for pointing out that the LP relaxation in figure 2 has infinite integrality gap.

References

[1] Michael Alekhnovich, Sanjeev Arora, and Iannis Tourlakis. Towards strong nonapproximability results in the Lovasz-Schrijver hierarchy. In *Proceedings of the 37th ACM Symposium on Theory of Computing*, pages 294–303, Baltimore, MD, USA, 2005.

[2] Noga Alon and Joel Spencer. *The Probabilistic Method*. Wiley-Interscience Series, 2008.

[3] Sanjeev Arora, Béla Bollobás, and László Lovász. Proving integrality gaps without knowing the linear program. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, pages 313–322, Vancouver, BC, Canada, 2002.

[4] Sanjeev Arora, Béla Bollobás, László Lovász, and Iannis Tourlakis. Proving integrality gaps without knowing the linear program. *Theory of Computing*, 2(2):19–51, 2006.

[5] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows and a $\sqrt{\log n}$-approximation to sparsest cut. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, Chicago, IL, USA, 2004.

[6] Eli Ben-Sasson. *Expansion in Proof Complexity.* PhD thesis, Hebrew University, 2001.

[7] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow: Resolution made simple. *Journal of the ACM*, 48(2), 2001.

[8] Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, and Aravindan Vijayaraghavan. Detecting high log-densities – an $O(n^{1/4})$ approximation for Densest k-Subgraph. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, Cambridge, MA, USA, 2010.

[9] Josh Buresh-Oppenheim, Nicola Galesi, Shlomo Hoory, Avner Magen, and Toniann Pitassi. Rank bounds and integrality gaps for cutting planes procedures. In *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science*, pages 318–327, Cambridge, MA, USA, 2003.

[10] Moses Charikar. On semidefinite programming relaxations for graph coloring and vertex cover. In *Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms*, pages 616–620, San Francisco, CA, USA, 2002.

[11] Kevin K. H. Cheung. On Lovász-Schrijver lift-and-project procedures on the Dantzig-Fulkerson-Johnson relaxation of the TSP. *SIAM J. on Optimization*, 16(2):380–399, 2005.

[12] V Chvàtal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, (4), 1973.

[13] Matthew Clegg, Jeff Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 174–183, Philadelphia, Pennsylvania, USA, 1996.

[14] W. Cook, C. R. Coullard, and G. Turán. On the complexity of cutting-plane proofs. *Discrete Appl. Math.*, 18(1):25–38, 1987.

[15] William Cook and Sanjeeb Dash. On the matrix-cut rank of polyhedra. *Math. Oper. Res*, 26(1):19–30, 2001.

[16] Sanjeeb Dash. An exponential lower bound on the length of some classes of branch-and-cut proofs. In *IPCO*, pages 145–160, 2002.

[17] Uriel Feige, Guy Kortsarz, and David Peleg. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[18] Uriel Feige and Robert Krauthgamer. The probable value of the Lovász–Schrijver relaxations for maximum independent set. *SIAM J. Comput.*, 32(2):345–370, 2003.

[19] Peter Frankl and Vojtech Rodl. Forbidden intersections. *Transactions of the American Mathematical Society*, 300(1):259–286, 1987.

[20] Konstantinos Georgiou, Avner Magen, Toniann Pitassi, and Iannis Tourlakis. Integrality gaps of 2 - o(1) for vertex cover SDPs in the Lovász-Schrijver hierarchy. In *Proceedings of the 48th IEEE Symposium on Foundations of Computer Science*, pages 702–712, Providence, RI, USA, 2007.

[21] Michel X. Goemans and Levent Tunçel. When does the positive semidefiniteness constraint help in lifting procedures? *Math. Oper. Res.*, 26(4):796–815, 2001.

[22] Doron Goldstein and Michael Langberg. The dense k subgraph problem. 2009. arXiv:math/0912.5327.

[23] R E Gomory. Outline of an algorithm for integer solutions to linear programs. *Bull. Amer. Math. Soc*, (64), 1958.

[24] Dima Grigoriev, Edward A. Hirsch, and Dmitrii V. Pasechnik. Complexity of semi-algebraic proofs. In *STACS*, pages 419–430, 2002.

[25] Mark Jerrum. Large cliques elude the metropolis process. *Random Struct. Algorithms*, 3(4):347–360, 1992.

[26] Jon M. Kleinberg and Michel X. Goemans. The Lovász Theta function and a semidefinite programming relaxation of vertex cover. *SIAM Journal on Discrete Mathematics*, 11:196–204, 1998.

[27] Arist Kojevnikov and Dmitry Itsykson. Lower bounds of static Lovász-Schrijver calculus proofs for tseitin tautologies. In *ICALP (1)*, pages 323–334, Venice, Italy, 2006.

[28] Luděk Kučera. Expected complexity of graph partitioning problems. *Discrete Appl. Math.*, 57(2-3):193–212, 1995.

[29] Jean B. Lasserre. An explicit exact SDP relaxation for nonlinear 0-1 programs. In *Proceedings of the 8th Conference on Integer Programming and Combinatorial Optimization*, pages 293–303, London, UK, 2001. Springer-Verlag.

[30] M. Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for 0-1 programming. *Mathematics of Operations Research*, 28:470–496, 2003.

[31] Frank T. Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46:787–832, 1999.

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

[33] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. on Optimization*, 1(12):166–190, 1991.

[34] Toniann Pitassi and Nathan Segerlind. Exponential lower bounds and integrality gaps for tree-like Lovász-Schrijver procedures. In *Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms*, pages 355–364, New York, NY, USA, 2009.

[35] Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symb. Log.*, 62(3):981–998, 1997.

[36] Grant Schoenebeck. Linear level Lasserre lower bounds for certain k-csps. In *Proceedings of the 49th IEEE Symposium on Foundations of Computer Science*, Philadelphia, Pennsylvania, USA, 2008.

[37] Grant Schoenebeck, Luca Trevisan, and Madhur Tulsiani. A linear round lower bound for Lovász-Schrijver SDP relaxations of vertex cover. In *Proceedings of the 22nd IEEE Conference on Computational Complexity*, San Diego, CA, USA, 2007.

[38] Grant Schoenebeck, Luca Trevisan, and Madhur Tulsiani. Tight integrality gaps for Lovasz-Schrijver LP relaxations of vertex cover and max cut. In *Proceedings of the 39th ACM Symposium on Theory of Computing*, San Diego, CA, USA, 2007.

[39] Hanif D. Sherali and Warren P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.*, 3(3):411–430, 1990.

[40] Iannis Tourlakis. New lower bounds for vertex cover in the Lovasz-Schrijver hierarchy. In *Proceedings of the 21st IEEE Conference on Computational Complexity*, Prague, Czech Republic, 2006.

[41] Madhur Tulsiani. CSP gaps and reductions in the Lasserre hierarchy. In *Proceedings of the 41st ACM Symposium on Theory of Computing*, Bethesda, MD, USA, 2009.