New lower bounds for Vertex Cover in the Lovász-Schrijver hierarchy

Iannis Tourlakis*
Department of Computer Science
Princeton University
35 Olden Street, Princeton, NJ 08540
itourlak@cs.princeton.edu

Abstract

Lovász and Schrijver [13] defined three progressively stronger procedures LS_0, LS and LS_+, for systematically tightening linear relaxations over many rounds. All three procedures yield the integral hull after at most n rounds. On the other hand, constant rounds of LS_+ can derive the relaxations behind many famous approximation algorithms such as those for MAX-CUT, SPARSEST-CUT. *So proving round lower bounds for these procedures on specific problems may give evidence about inapproximability.*

We prove new round lower bounds for VERTEX COVER *in the LS hierarchy. Arora et al. [3] showed that the integrality gap for* VERTEX COVER *relaxations remains $2-o(1)$ even after $\Omega(\log n)$ rounds LS. However, their method can only prove round lower bounds as large as the girth of the input graph, which is $O(\log n)$ for interesting graphs.*

We break through this "girth barrier" and show that the integrality gap for VERTEX COVER *remains $1.5-\epsilon$ even after $\Omega(\log^2 n)$ rounds of LS. In contrast, the best PCP-based results only rule out 1.36-approximations. Moreover, we conjecture that the new technique we introduce to prove our lower bound, the "fence" method, may lead to linear or nearly linear LS round lower bounds for* VERTEX COVER.

1. Introduction

Lovász and Schrijver [13] introduced three progressively stronger *lift-and-project* procedures LS_0, LS, LS_+ for tightening linear relaxations over many rounds (also see Sherali and Adams [14] for a related procedure). These procedures tighten relaxations by progressively adding, in a controlled manner, all inequalities satisfied by integral solutions. They do this by first "lifting" the polytope for the linear relaxation to a higher dimensional space, adding inequalities to the high-dimensional polytope, and then projecting the resulting polytope back to the original space. Interest in these procedures comes from the following algorithmic property they enjoy: optimizing a linear function over the rth round relaxation produced by these procedures can be done in $O(n^r)$ time provided that the original relaxation has a weak separation oracle (for more on these methods see Lovász and Schrijver's original paper [13] introducing their methods or the exposition in Feige and Krauthgamer [8]).

Each of these procedures can be thought of as defining a natural restricted computation model. For example, in the computation model corresponding to the LS procedure, a problem is computed in n^r "time" if the problem can be computed by optimizing a linear function over a relaxation resulting from r rounds of LS tightening. These computation models seem quite powerful: It is known that all three procedures yield the integral hull after n rounds. Hence, all of **NP** is computable in "exponential time" even in the LS_0 model. On the other hand, many recent celebrated approximation algorithms such as the Goemans-Williamson [10] algorithm for MAX-CUT and the Arora-Rao-Vazirani [4] algorithm for SPARSEST-CUT can be derived within a constant number of LS_+ rounds and hence, are computable in "polynomial time" in the LS_+ model.

Thus, proving (unconditional) inapproximability results in these computation models may give evidence about a problem's true inapproximability. Indeed, one motivation for studying lift-and-project procedures comes from the remaining gaps for several optimization problems between the approximation ratios achieved by known algorithms and the approximation ratios ruled out using probabilistically checkable proofs (PCPs): strong lower bounds in the Lovász-Schrijver hierarchies may give evidence about the true inapproximability ratio for such problems.

One such problem which has attracted the attention of researchers studying lift-and-project methods is the minimum VERTEX COVER problem for graphs. Whereas the best al-

*Supported through Sanjeev Arora's NSF grants MSPA-MCS 0528414, CCF 0514993, ITR 0205594.

Proceedings of the 21st Annual IEEE Conference on Computational Complexity (CCC'06)
0-7695-2596-2/06 $20.00 © 2006 IEEE

gorithms known give $2 - o(1)$ approximations (even the trivial linear programming relaxation achieves this ratio), the best PCP-based hardness results only rule out 1.36 approximations (Dinur and Safra [7]). However, $2 - o(1)$ approximations are ruled out by Khot and Regev [12] assuming Khot's Unique Games conjecture [11]. Nevertheless, as long as the validity of Khot's conjecture remains unknown, proving strong (unconditional) integrality gaps for VERTEX COVER in the Lovász-Schrijver hierarchies may provide an alternative source of evidence about the true hardness of approximation for VERTEX COVER.

Proving lower bounds even in the weaker LS_0 and LS hierarchies for problems such as VERTEX COVER defined by 2-variable constraints has proved very difficult. This contrasts with problems defined by 3 (or more) variable constraints (e.g., MAX-3SAT, hypergraph VERTEX COVER) where strong inapproximability results are known even after $\Omega(n)$ rounds of LS_+ (see related work below).

The best previous inapproximability result known for VERTEX COVER in the Lovász-Schrijver hierarchies was due to Arora et al. [3] where they showed that the integrality gap remains $2 - \epsilon$ even after tightening the standard linear relaxation for VERTEX COVER with $\Omega(\log n)$ rounds of LS lift-and-project. Alternatively, their result can be viewed as saying that in the LS computation model there exists no non-trivial $o(n^{\log n})$ "time" approximation algorithm for VERTEX COVER. Unfortunately, their techniques can only prove lower bounds when the number of LS rounds is at most the girth of the input graph, which is $O(\log n)$ for graphs with large integrality gaps.

We break through this "girth barrier" and obtain integrality gaps for VERTEX COVER after even $\Omega(g^2)$ LS rounds when the girth of the input graph is g. Consequently we show that VERTEX COVER relaxations produced after $\Omega(\log^2 n)$ rounds of LS have integrality gaps of size $1.5-\epsilon$ for any $\epsilon > 0$. In other words, there exists no $1.5 - \epsilon$ approximation algorithm for VERTEX COVER in the LS computation model running in "time" $o(n^{\log^2 n})$.

While smaller than the ratio $2 - \epsilon$ ruled out by [3], our integrality gap is larger than the inapproximability ratio of 1.36 proved using PCP-based techniques [7]. In addition, we stress that our results are unconditional and do not rely on any "computational complexity" assumptions. As we discuss further in section 5, we conjecture that our techniques may yet yield integrality gaps after even linear rounds of LS.

1.1. Related Work

The work of Arora et al. [3] already mentioned extended earlier work by Arora et al. [2] showing that the integrality gap remained $2 - o(1)$ after $\Omega(\sqrt{\log n})$ rounds of LS. Related to these papers is a result by Tourlakis [16] showing that the integrality gap for hypergraph VERTEX COVER remains $k - o(1)$ even after $\Omega(\log \log n)$ rounds of LS.

While the above results are those most easily comparable to our main result, the techniques used in Buresh-Oppenheim et al. [5] and Alekhnovich et al. [1] are somewhat more related to those used here (this will be discussed further in the next section). Buresh-Oppenheim et al. show that the integrality gap for MAX-kSAT, $k \geq 5$, remains $(2^k - 1)/2^k - \epsilon$ even for relaxations produced after $\Omega(n)$ rounds of LS_+ tightenings. Alekhnovich et al. extend these results to MAX-3SAT. In addition, they prove integrality gaps of $(1 - \epsilon) \ln n$ for SET-COVER and $k - 1 - \epsilon$ for VERTEX COVER on k-hypergraphs even after $\Omega(n)$ rounds of LS_+.

Other recent papers studying Lovász-Schrijver liftings are [15, 6, 9, 8]. In the first three papers the emphasis is on proving lower bounds on the number of rounds needed to derive specific inequalities true for the integral hull; integrality gaps are not explored. While Feige and Krauthgamer [8] also do not explicitly consider integrality gaps, an easy corollary of their results shows that a large integrality gap remains for INDEPENDENT SET even after $\Omega(\log n)$ rounds of LS_+.

2. Lovász-Schrijver liftings and proving round lower bounds

We use VERTEX COVER to explain relaxations and how they are tightened using LS-liftings. Recall that the integer programming (IP) characterization of VERTEX COVER for a graph $G = (V, E)$ is:

$$\begin{aligned} &\min \sum_{i \in V} v_i \\ \text{s.t.} \quad & v_i + v_j \geq 1 \qquad \forall \{i, j\} \in E \\ & v_i \in \{0, 1\} \qquad \forall i \in [n]. \end{aligned}$$

The *integer hull* I is the convex hull of all the IP solutions. The standard LP relaxation allows $0 \leq v_i \leq 1$. So the value of the LP is at most that of the IP. Let P be the convex hull of solutions to the LP. This LP is *tightened* by adding constraints that also hold for I. This results in a new tighter polytope P' where $I \subseteq P' \subseteq P$. The quality of a relaxation P is measured by the ratio $\frac{\text{optimum value over } I}{\text{optimum value over } P}$, called the *integrality gap*.

Lovász and Schrijver's [13] lift-and-project procedures systematically tighten relaxations of 0-1 IPs. They do this by taking the n-dimensional polytope for a relaxation and "lifting" it to n^2 dimension, adding new constraints to the high-dimensional polytope, and then projecting the resulting polytope back to n-space. We will not give a full technical exposition of their method here: to understand

our results the characterization of these lifting is given by Lemma 2.1 below will suffice.

Before giving this characterization we require definitions and notation. For technical reasons, the notation for LS liftings uses homogenized inequalities. In particular, given a linear relaxation we will introduce a new variable x_0 and replace each constraint $a^T x \geq b$ in the relaxation with the constraint $a^T x \geq b x_0$. For example, if G is a graph and we are considering the VERTEX COVER problem, then we will work with the convex cone $\mathrm{VC}(G)$ in $\mathcal{R}^{n+1}$ consisting of all vectors $(x_0, x_1, \ldots, x_n)$ such that

$$x_i + x_j \geq x_0 \qquad \forall \{i,j\} \in E, \tag{2.1}$$

$$0 \leq x_i \leq x_0 \qquad \forall i \in [n]. \tag{2.2}$$

All cones in what follows will be in $\mathcal{R}^{n+1}$ where the $n+1$ dimensions are indexed $0, 1, 2, \ldots, n$. We will be interested in the slice cut by the hyperplane $x_0 = 1$.

While this paper concentrates on LS liftings, we will also define the related LS_0 and LS_+ lift-and-project methods for completeness. Given a convex cone K, denote by $N_0^r(K)$, $N^r(K)$ and $N_+^r(K)$ the feasible cones of all (homogenized) inequalities obtained by applying r rounds of the LS_0, LS and LS_+ tightening procedures, respectively. We define $N_0^r(K) = N^r(K) = N_+^r(K) = K$.

The following lemma characterizes the effect of one round of these procedures:

Lemma 2.1 ([13]) *Let $K \subseteq \mathcal{R}^{n+1}$ be a convex cone. Then $x \in N_0(K)$ iff there exists a $(n+1) \times (n+1)$ matrix Y such that,*

1. *$Ye_0 = diag(Y) = x$,*
2. *For all $1 \leq i \leq n$, $Ye_i, Y(e_0 - e_i) \in P$.*

The vector $x \in N(K)$ iff in addition Y is symmetric. The vector $x \in N_+(K)$ iff in addition Y is positive semidefinite.

As suggested in [5] we will call the matrix Y in the above lemma a *protection matrix* since it "protects" the vector x for one round of tightening.

Given a protection matrix Y, the set

$$V(Y) = \{Ye_i, Y(e_0 - e_i) : 1 \leq i \leq n\}$$

is called the set of *protection vectors corresponding to* Y. The points x we protect will always have $x_0 = 1$. For such points, we will often consider the projections of the vectors in $V(Y)$ on to the hyperplane $x_0 = 1$. This set is called the set of *projected protection vectors corresponding to* Y and is defined as:

$$PV(Y) = \{Ye_i/x_i : i \in [n], 0 < x_i < 1\} \cup \{Y(e_0 - e_i)/(1 - x_i) : i \in [n], 0 < x_i < 1\}.$$

Corollary 2.2 *Let $K \subseteq \mathcal{R}^{n+1}$ be a convex cone and let $y \in K$, $y_0 = 1$. Suppose that Y is an $(n+1) \times (n+1)$ symmetric matrix such that,*

1. *$Ye_0 = diag(Y) = y$,*
2. *$PV(Y) \subseteq K|_{x_0=1}$.*

Then $y \in N(K)$.

To show that a large integrality gap remains for VERTEX COVER after r rounds of LS tightening it suffices to proceed as follows. First we begin with a graph G where the minimum vertex cover has size nearly n. We will then show that there exists some vector $w \in \mathrm{VC}(G)$, $w_0 = 1$, for which $\sum w_i$ is much smaller than n and such that $w \in N^r(\mathrm{VC}(G))$.

Corollary 2.2 suggests using inductive arguments to show that some vector w is in $N^r(\mathrm{VC}(G))$. As first suggested in [5], such an argument can be phrased as a Prover-Adversary game: The game maintains a vector x, initially w, and proceeds in rounds. Each round the following moves are made:

1. Given the current value for x, the Prover produces a candidate protection matrix Y_x supposedly showing that $x \in N(\mathrm{VC}(G))$.
2. The Adversary picks one vector y from $PV(Y_x)$ and sets x to y.

The game ends when x is no longer in P, i.e., when the Adversary forces the Prover into constructing an invalid candidate protection matrix. The following lemma follows immediately from Lemma 2.1 and the definition of the Prover-Adversary game, and was noted by Buresh-Oppenheim et al. [5] (see also [1]):

Lemma 2.3 *Suppose $w \in \mathrm{VC}(G)$. If there exists a strategy for the Prover such that the game lasts r rounds no matter what the Adversary does, then $x \in N^r(\mathrm{VC}(G))$.*

PROOF: The proof is by induction on r. The base case $r = 0$ is trivially true. So suppose the lemma holds for r and suppose the Prover has an $r+1$ round strategy for w. In the first round of the game the Prover produces a matrix Y_x according to its strategy. Since the Prover's strategy works for $r+1$ rounds, no matter which vector $y \in PV(Y_x)$ the Adversary selects, the game will last r further rounds. By induction, it follows that $PV(Y_x) \subseteq N^r(\mathrm{VC}(G))$. Corollary 2.2 then implies $x \in N^{r+1}(\mathrm{VC}(G))$. □

Let $conv(S)$ denote the convex hull of a set S. Note that if $x \in conv(S)$ and $S \subseteq N^r(\mathrm{VC}(G))$, it follows also that $x \in N^r(\mathrm{VC}(G))$. We can use this observation to modify the game rules as follows:

1. Given the current point x, the Prover produces a candidate protection matrix Y_x. The Prover then produces a set S_x of points such that $PV(Y_x) \subseteq conv(S_x)$.

2. The Adversary picks one vector y from S_x and sets x to y.

Modifying the proof of Lemma 2.3, it follows that if there exists a strategy for the Prover in this new game such that the game lasts r rounds no matter what the Adversary does, then $w \in N^r(\mathrm{VC}(G))$.

The intuition for introducing the rules of the revised game is that the vectors in S_x may have nicer structural properties than the vectors in $PV(Y_x)$ facilitating the Prover's strategy in future rounds of the game. Indeed, expressing the vectors in $PV(Y_x)$ using convex combination will be crucial for our "fence" method sketched in Section 2.1 below and described formally in Section 4.2.4.

In practice, when trying to show $w \in N^r(\mathrm{VC}(G))$, we will pick w so that it enjoys some nice structural properties. Hence, in our lower bound, the Prover will always construct S_x so that the difference between any vector in S_x and the current x is both minimal and predictable.

Remark 2.4 *If a coordinate is set to 0 or 1 in the game, then that coordinate will remain 0 or 1, respectively, for the remainder of the game: This follows from the definition of $PV(Y_x)$ and from the fact that the $PV(Y_x) \subseteq conv(S_x)$.*

We now note one way in which our approach will more resemble that taken in [5] and [1] rather than that in [3]. In [3] LP-duality was used to prove the existence of appropriate protection matrices needed for their lower bound; no explicit description for their protection matrices was obtained. While our protection matrices will be completely different than those used in [5, 1], nevertheless, as in those papers we will always give explicit descriptions for them. This is crucial since our arguments will require explicit descriptions for the sets $PV(Y_x)$.

2.1. The "Fence" trick

Good fences make good neighbours.
–Robert Frost (from "Mending Wall", 1914)

The key to our new lower bound is what we call the "Fence" method which we now roughly sketch. A more technical description will be given in Section 4.2.4.

As in [3] our lower bound will be proved for a graph G with girth $\Theta(\log n)$. To prove that a large integrality gap remains after $\Omega(\log^2 n)$ rounds of LS tightenings we start the Prover-Adversary game with a "bad" fractional solution vector w (i.e., the value of the objective function on w is far from the true integral optimum) and show that there is a $\Omega(\log^2 n)$ round strategy for the Prover against any Adversary. The vector w will be chosen so that it satisfies some "nice" structural properties.

In each round of the game, given the current vector x the Prover's strategy will be to design S_x (the set of vectors from which the Adversary chooses x for the next round) such that the difference between x and any vector in S_x is minimal. For technical reasons, vectors in S_x will always differ in at least a few coordinates from x. Hence, as more and more rounds of the game are played, the current vector x will differ more and more from the initial vector w.

Let C_x be the induced subgraph of G on those vertices (i.e., coordinates) that x differs from w. For technical reasons, our Prover strategy will always be successful against the Adversary *provided that C_x has no component with diameter greater than half the girth of G*.

Now, it is not too hard to tailor the Prover's strategy so that in the ith round of the game the sum of the diameters of all components in C_x is at most $O(i)$ (for instance, by adapting the arguments in [3]). So since G has girth $\Theta(\log n)$, the Prover can use such a strategy to play $\Omega(\log n)$ rounds of the game against any Adversary. However, this strategy will fail beyond $\Omega(\log n)$ rounds since C_x may then contain a component with diameter greater than half the girth.

To continue the game, the Prover then uses the "fence trick": If some vector y in S_x (the set from which the Adversary chooses x for the next round of the game) is such that C_y has a component A with diameter nearly half the girth, then the Prover will put a "fence" around this component to stop it from growing any larger and becoming "dangerous". The Prover does this by taking advantage of Remark 2.4 which implies that during the game we can ignore all nodes in G (i.e., remove the respective coordinates from x) that are set to 1 by x (this is made formal in section 4.1). So to put a "fence" around A, the Prover expresses y as a convex combination of vectors each of which sets some nodes surrounding A to 1, disconnecting it from the rest of G.

3. The Main Theorem

We prove our main result in this section: Given a graph G, let $\mathrm{VC}(G) \subseteq \mathcal{R}^{n+1}$ denote the convex cone of feasible solutions to constraints (2.1) and (2.2), the homogenized relaxed VERTEX COVER constraints for G.

Theorem 3.1 *For all $\epsilon > 0$ there exists a constant $\delta > 0$ and an integer n_0 such that for all $n \geq n_0$ there exists an n-vertex graph G for which $N^r(\mathrm{VC}(G))$ has an integrality gap of at least $1.5 - \epsilon$ for all $r \leq \delta \log^2 n$.*

The graphs used to prove Theorem 3.1 are high-girth sparse graphs with degree bounded by some constant $d \geq 3$. With non-zero probability, such graphs have a maximum in-

dependent set of size $O(\frac{n \log d}{d})$. In particular, we have the following theorem from standard graph theory:

Theorem 3.2 *There exist constants $\alpha, \beta > 0$ and integers n_0, d_0 such that for all $n \geq n_0$ and all $d \geq d_0$, there exists a graph $G(n,d)$ with n vertices, degree at most d, girth at least $\beta \log n$, and maximum independent set size at most $\frac{\alpha n \log d}{d}$.*

The graphs given by the above theorem will be used to prove the following theorem from which Theorem 3.1 will follow.

Theorem 3.3 *Let $w = (1, \frac{2}{3}, \frac{2}{3}, \ldots, \frac{2}{3}) \in \mathcal{R}^{n+1}$. Let n_0, d_0 be the constants given by Theorem 3.2. Then for all $n \geq n_0$ and all $d \geq d_0$, the point w is in $N^r(VC(G(n,d))$ for $r = \Omega(\log^2 n)$.*

Section 4 below is devoted to proving Theorem 3.3.

PROOF:[Proof of Theorem 3.1] Theorem 3.3 shows that for large n and d there exist graphs G for which the integrality gap of $N^r(\mathrm{VC}(G))$ for $r = \Omega(\log^2 n)$ is $\frac{3}{2}(1 - \frac{\alpha \log d}{d})$. The latter quantity can be made arbitrarily close to 1.5 by taking d sufficiently large. □

4. Proof of Theorem 3.3

Fix $d \geq d_0$ and $n \geq n_0$, and let $G = G(n,d)$ be the graph given by Theorem 3.2 for these values. Let g denote the girth of G. Note that $g \geq \beta \log n$.

To prove Theorem 3.3 we will first require some definitions. Then we will describe the Prover's strategy for the vector w and graph G. Concurrently, we will show that the described strategy works for r rounds against any adversary, where $r = (g/28)^2/2$.

4.1. Round invariants and Components

We will define three properties/invariants that the Prover will ensure the current vector x for the game satisfies at the beginning of each round. The first is the following:

> **Property 1:** $x \in \mathrm{VC}(G)$ and $x \in \left\{0, 1, \frac{1}{3}, \frac{2}{3}\right\}^{n+1}$ (of course, $x_0 = 1$ always).

This invariant allows us to make several crucial definitions. First we make some observations.

Note that the constraints for any edge in G incident to a vertex i where $x_i = 1$ are trivially satisfied. Moreover, since $x \in \mathrm{VC}(G)$, if $x_i = 0$ then x_j must be 1 for all vertices j adjacent to i. Hence, since vertices that are 0/1 valued will never change their values in subsequent rounds (see Remark 2.4), when analyzing the effect of one round of N it will suffice to only consider the subgraph G_x of G induced by those vertices with value in $\left\{\frac{1}{3}, \frac{2}{3}\right\}$ under x. We will say that *vertex j has value a in G_x* if $x_j = a$.

Next we define the concept of a *simple component* in G_x. Intuitively, a simple component in G_x is any connected component in G_x such that all edges with both vertices in the component have one vertex with value $\frac{1}{3}$ and the other with value $\frac{2}{3}$. We now make this precise.

Given $x \in \mathrm{VC}(G)$, let G'_x be the subgraph of G_x induced by all edges (i,j) in G_x such that one of x_i or x_j is $\frac{1}{3}$ (note that since $x \in \mathrm{VC}(G)$, at most one vertex in each edge has value $\frac{1}{3}$).

Definition 4.1 *A (vertex induced) subgraph C of G_x is called a* simple component *if it is a maximal connected component of G'_x (i.e., adding any vertex of G'_x to C results in an unconnected (vertex induced) subgraph of G'_x).*

We will ensure (see Property 2 below) throughout the Prover-Adversary game that all simple components in G_x have diameter much smaller than half the girth of G. Hence, adjacent vertices in a given simple component cannot both have value $\frac{2}{3}$ under x. In particular, this ensures that the following definition is consistent:

Definition 4.2 *A node i belongs to the* boundary *of a simple component C in G_x if (a) $i \in C$, (b) $x_i = \frac{2}{3}$, and (c) there exists j such that $x_j = \frac{2}{3}$ and $(i,j) \in E$. (Note that since C has diameter less than $g/2$, j cannot be in C.)*

The edge distance $d_E(C, C')$ in G_x *between two simple components C and C' is the length (i.e., number of edges) of the shortest path in G_x connecting some boundary node of C with some boundary node of C'. The* distance $d(C, C')$ *between the two simple components is equal to $d_E(C, C') - 1$. If no paths exists between the components, then $d_E(C, C')$ and $d(C, C')$ are defined to be infinite.*

Consider the following procedure: Let D be a set whose items will be simple components of G_x, and suppose D initially contains only one simple component C. Repeat the following procedure until D no longer grows in size: For every simple component C' in D, add to D all simple components C'' that are within distance 2 of C' in G_x. The final set from this procedure is called the *closure of C*.

Definition 4.3 *A vertex induced subgraph C of G_x is a* complete component *if there exists a simple component C' in G_x such that the vertices of C are precisely the vertices of all simple components in the closure of C'.*

Intuitively, a complete component is formed by adding to a simple component all simple components that are "near" it, and then adding all simple components that are "near" the resulting component, and so on.

Definition 4.4 *A node in G_x is* untouched *if it belongs to no complete component. Otherwise, the node is* touched*. Note that untouched nodes always have value $\frac{2}{3}$.*

Distance between two complete components is defined in the same way it was defined for simple components. Note that by definition the distance between two components is at least 3. This fact is important, so we make special note of it:

> **Observation 1:** The distance between two complete components is at least 3. Hence, along any path connecting two complete components in G_x there are at least 3 consecutive untouched nodes.

The second invariant the Prover will ensure is that at the beginning of each round all complete (and hence all simple) components in G_x have diameter substantially smaller than g. In particular, it will ensure that the following property holds for x at the beginning of each round:

> **Property 2:** No complete component in G_x has diameter greater than γ, where $\gamma = g/28$.

If at some point in the game some complete component's diameter gets too large, then the Prover will "remove" the complete component from G_x. Intuitively, this will be done by altering x so that the values of nodes immediately around the too-large component are 0/1. Hence the complete component will be "fenced off" from the rest of the graph and can be ignored for the rest of the game. The details of how this is done are in Section 4.2.4.

Finally, the following Property will also be ensured:

> **Property 3:** At the beginning of round i the sum of the diameters of all complete components in G_x is at most $7i$.

In the next section we describe the Prover's strategy in round $i \leq r = \gamma^2/2 = \Omega(\log^2 n)$. While describing the strategy we will prove that if $i < r$, then the Prover's strategy will guarantee that Properties 1–3 will hold at the start of the following round. Hence, Theorem 3.1 will follow.

4.2. The Prover's strategy for x in round i

4.2.1 "High-level" description

The Prover's strategy can be described as follows: By induction, the Prover can assume that the Properties 1–3 hold for x at the start of the round; they of course hold for the base case vector $x = w$. The Prover will then construct the "obvious" protection matrix Y_x for x where "obvious" will be made precise below; the Round Invariants will be crucial for this (Section 4.2.2). The set $PV(Y_x)$ may not be contained in $x \in \left\{0, 1, \frac{1}{3}, \frac{2}{3}\right\}^{n+1}$ (in particular, some vectors may have entries that are $\frac{1}{2}$), i.e., these vectors may not satisfy Property 1. The Prover will then construct a new set $S'_x \subseteq \left\{0, 1, \frac{1}{3}, \frac{2}{3}\right\}^{n+1}$ of vectors such that $PV(Y_x) \subseteq conv(S'_x)$ (Section 4.2.3). The vectors in S'_x may not satisfy Property 2. The "fence" trick will be used by the prover to construct a new set S_x satisfying all the invariants and such that $S'_x \subseteq conv(S_x)$ (Section 4.2.4). The details now follow.

4.2.2 The "obvious" protection matrix Y_x for x

To simplify notation, we will write Y for Y_x. Since Y is supposed to be symmetric, we will often use the notation Y_{ij} instead of $Y_{i,j}$ or $Y_{j,i}$.

The protection matrix Y for x will depend on the structure and properties enjoyed by complete components in G_x. In particular, we will use the fact that Property 2 implies all complete (and simple) components are trees. Moreover, we will also use Observation 1 which says that there is a "buffer" between complete components. This "buffer" will ensure that entries of Y corresponding to one complete component are independent of those for other complete components. In particular, whenever two vertices i and j are not in the same complete component this will generally allow us to set $Y_{ij} = x_i x_j$. Only for i, j in the same complete component (and some isolated other cases), will there be a "non-trivial" value for Y_{ij}. The formal definition for Y now follows.

We will define Y by defining Ye_i for all i, $1 \leq i \leq n$, and then arguing that our definition is symmetrical. Fix a node i in G. If $x_i = 0$, then Ye_i is simply $\vec{0}$. If $x_i = 1$, then $Ye_i = x$. So assume $0 < x_i < 1$. As required by the definition of a protection matrix, we have that $Y_{ii} = x_i$. There are now two cases depending on whether i is contained in a complete component or not.

Case 1: i is not in a complete component

Consider i's neighbours. These may or may not belong to complete components (in particular, they may or may not lie on the boundary of some simple component). Note that it follows from Observation 1 that at most one neighbour of i belongs to a complete component.

We define Ye_i as follows: for each neighbour i_j of i let $Y_{i_j,i} = \frac{1}{3}$. For all remaining vertices $\ell \in G$, we set $Y_{\ell,i} = x_\ell x_i$. If there is a neighbour, say i_1, that belongs to a complete component, we must make some adjustments to Ye_i: Let C' be the simple component in which i_j belongs, and set $Y_{k,i} = \frac{1}{3}$ for all $k \in C'$.

Case 2: i is in some complete component C

There are two subcases depending on whether vertex i is in a simple component or not.

First consider the case that i is in some simple component C'. Either $x_i = \frac{1}{3}$, or $x_i = \frac{2}{3}$. Suppose $x_i = \frac{1}{3}$. Then for all vertices $k \in C'$, $Y_{k,i} = \frac{1}{3}$ if $x_k = \frac{1}{3}$ (i.e., since C'

is a tree, k has even distance from i in C'), while $Y_{k,i} = 0$ otherwise (i.e., if the distance is odd). For all simple components D in C distance 0 from C' in G_x, and for all nodes $k \in D$, let $Y_{k,i} = \frac{1}{3}$ if $x_k = \frac{2}{3}$, and let $Y_{k,i} = 0$ otherwise. For all nodes $k \in G_x$ that do not belong to any complete component and such that k is adjacent to a boundary node of C', let $Y_{k,i} = \frac{1}{3}$. Finally, for all remaining nodes $\ell \in G$, set $Y_{\ell,i} = x_\ell x_i$.

Now suppose instead that $x_i = \frac{2}{3}$. Then for all vertices $k \in C'$, $Y_{k,i} = \frac{2}{3}$ if $x_k = \frac{2}{3}$, while $Y_{k,i} = 0$ otherwise. For all simple components D in C distance 0 from C' in G_x, $Y_{k,i} = \frac{1}{3}$ for all nodes $k \in D$. For all nodes $k \in G$ that do not belong to any component and such that k is adjacent to a boundary node of C', we have that $Y_{k,i} = \frac{1}{3}$. Finally, for all remaining nodes $\ell \in G$, we set $Y_{\ell,i} = x_\ell x_i$.

Now we consider the case that i is not in any simple component. We must then have $x_i = \frac{2}{3}$. Moreover, every vertex adjacent to i must also have value $\frac{2}{3}$ in G_x. We define Ye_i as follows. For each vertex k adjacent to i we set $Y_{k,i} = frac13$. In addition, for each simple component D in C adjacent to i (note that there must be at least one such simple component), we set $Y_{k,i} = \frac{1}{3}$ for all vertices $k \in D$. Finally, for all remaining nodes $\ell \in G$, we set $Y_{\ell,i} = x_\ell x_i$.

It remains now to argue that (1) the definition of Y is indeed symmetric, and that (2) the protection vectors defined by Y are in $\mathrm{VC}(G)$.

Symmetry follows straightforwardly from the definition and a complete proof will not be given here. We will only note as an illustrative example how when i, j are both in the same simple component, then the definition of Y_{ij} always depends only on whether the distance between i and j is odd or even. Moreover, this is well-defined since all simple components are trees.

Now let us argue that the vectors in $PV(Y)$ are in $\mathrm{VC}(G)$. Fix i such that $0 < i < 1$ and consider $y = Ye_i/x_i$. Note that y is identical to x in all coordinates except:

1. Coordinate i which has $y_i = 1$,

2. If i was not in a simple component of G_x, then all (non 0/1) neighbours of i in G are now $\frac{1}{2}$, and all nodes in simple components of G_x that touched i are also $\frac{1}{2}$.

3. If i was in a simple component C', then all nodes at odd distance from i in C' are now 0 under y; and all nodes at even distance from i in C' are now 1 under y. Moreover, if $x_i = \frac{2}{3}$, then all nodes in those simple components (in G_x) that touched C' are now $\frac{1}{2}$. Free nodes touching C' are also set to $\frac{1}{2}$ under y. Finally, if instead $x_i = \frac{1}{3}$, then all nodes in a simple components D (in G_x) that touched C' are 1 under y if the node's distance (through D) to C' is even, and 0 under y if the distance to C' is odd. Again, in this case ($x_i = \frac{1}{3}$), free nodes touching C' are now set to 1 under y.

Since $x \in \mathrm{VC}(G)$ it follows that the VERTEX COVER constraints are satisfied by all edges whose nodes have the same value under y as under x. So let's concentrate on those nodes which changed as described above. Clearly all edges in those simple components whose values are affected as described in (2) and (3) above still satisfy the VERTEX COVER constraints under y. Moreover, using the fact there is a "buffer" between complete components (Observation 1), it follows from the definition of Y that the edges between affected components and unaffected nodes also satisfy the VERTEX COVER constraints. So y satisfies the VERTEX COVER constraints.

To show that $Y(e_0 - e_i)/(1 - x_i) \in \mathrm{VC}(G)$ uses similar arguments.

4.2.3 Constructing $S'_x \subseteq \left\{0, 1, \frac{1}{3}, \frac{2}{3}\right\}^{n+1}$ such that $PV(Y_x) \subseteq conv(S'_x)$

The vectors in $PV(Y_x)$ may not be in $\left\{0, 1, \frac{1}{3}, \frac{2}{3}\right\}^{n+1}$. Indeed they may contain $\frac{1}{2}$'s. However, since $PV(Y_x) \subseteq \mathrm{VC}(G)$, it follows that for all vectors $y \in PV(Y_x)$, the following is true: In the graph G_y, all nodes j with $y_j = \frac{1}{2}$ are adjacent to nodes whose values are either $\frac{1}{2}$ or $\frac{2}{3}$. Hence it is easy to see then that for each vector $y \in PV(Y_x)$ there exist vectors y^1 and y^2 such that y is the average of y^1 and y^2, and such that y^1 and y^2 are equal to y everywhere with the exception that for every node k such that $y_k = \frac{1}{2}$, then k is $\frac{1}{3}$ in one of y^1 or y^2, and $\frac{2}{3}$ in the other.

We then define S'_x to be the set containing precisely the vectors y^1 and y^2 corresponding to each vector $y \in PV(Y_x)$.

4.2.4 The "Fence" trick: Ensuring components have small diameter

The vectors in S'_x may not satisfy Property 2. To fix this, for each vector $y \in S'_x$ that does not satisfy Property 2, the Prover will construct a set B_y of vectors that will satisfy all three properties in the next round and moreover, $y \in conv(B_y)$. Essentially, the vectors in B_y will isolate all components that do not satisfy Property 2 by ensuring such vectors have a "fence" of 1's around such components. The set S_x will then be $\cup_{y \in S'_x} B_y$ (where $B_y = \{y\}$ if y does satisfy Property 2).

Before we describe how the Prover finds these "fences", let us consider how the graph G_y for some $y \in S'_x$ compares to G_x. Essentially, the only possible differences are:

1. Some free node is set to 0/1 which results in (a) either a new simple component of diameter 3 being created, or (b) in some previously existing simple component having its diameter increased by at most 3.

2. Some free node is set to 0/1 which results in at most d complete components being "merged". The resulting complete component will have diameter bounded by at most $3 + \gamma_1 + \gamma_2$ where γ_1 and γ_2 are the diameters of the two largest component involved in the merge.

3. Some node in a simple component C' is set to 0/1 which results in each adjacent simple components having its vertices either (a) set to 0/1 or (b) the pattern of $\frac{1}{3}$-$\frac{2}{3}$ for the values of the nodes in the component is reversed (i.e., nodes that were $\frac{1}{3}$ are now $\frac{2}{3}$ and vice versa). In addition, free nodes adjacent to the affected simple components may be altered. Let C be the complete component containing C'. In both cases (a) and (b), C may have some formerly untouched nodes added to it. In either case, the diameter of C increases by at most 2. This may result in C being closer than distance 3 to other complete components in which case these components merge to form a new complete component. Again, it is not hard to see that this new complete component will have diameter bounded by at most $6 + \gamma + \gamma_1 + \gamma_2$ where γ is the diameter of the complete component containing C' and γ_1 and γ_2 are the diameter of the two largest other component involved in the merge.

Note that in all cases, the sum of the diameters of all complete components in G_y increases by at most 6. Moreover, note that any complete component in G_y with diameter greater than γ will still have diameter bounded by 3γ.

We can now give a high-level description of the algorithm the Prover will use when putting up "fences" for a vector $y \in S'_x$ that violates Property 2. First the Prover will "group" all complete components that are "near" to each other in an effort to divide all complete components into sets of "super-components" of diameter at least $\gamma/2$ and at most 7γ. Note that any super-components that cannot be "grown" to a super-component of this diameter must be "far" from all other super-components; the Prover will ignore them. Super-components of diameter at least $\gamma/2$ are called *full-size*. The set of full-size components will be denoted by $\mathcal{C}$.

The Prover will then "isolate" all super-components by defining a set A of vectors such that for each super-component C (a) every path in G_y of length 2 adjacent to a boundary node of C has at least one vertex with value 1 in all vectors in S_1 and (b) $y \in conv(A)$. Hence, for all $z \in A$, in the graph G_z (a) each super-component is disconnected from the rest of the graph and (b) no complete component has diameter greater than $\gamma/2$. This isolation step is where the "fences" are put up.

Finally we will argue that this means that there exists yet another set B_y of vectors that are 0/1 on all super-components, will satisfy Properties 1–3 in the next round, and such that $A \subseteq conv(B_y)$. Thus the Prover effectively "removes" all super-components from the graph, and in particular, removes all complete components of diameter greater than γ from the graph.

We now describe formally how the Prover does "grouping", "isolating", and "removing". The arguments showing that these procedures produce a final set B_y whose vectors all satisfy Properties 1–3—and hence allow the Prover to successfully play at least one more round against the Adversary—will crucially rely on the fact that $i < r = \gamma^2/2$.

Grouping: The two rules for grouping complete components into super-components are as follows: (1) Two complete components can be put in the same super-component if the distance between them is at most 8; and (2) The diameter of a super-component cannot exceed 7γ. Note that all complete components in G_y have diameter at most 3γ. Moreover, by definition all complete components not in $\mathcal{C}$ have diameter at most γ.

Do an initial grouping of all components as follows: Let the first group g_1 initially contain some complete component C of G_y. Now keep adding to g_1 any complete component that is within distance 8 of some complete component already in g_1. Do this until no more such complete components can be found. For the second group g_2, let it initially contain some complete component C' not in g_1. Again add to g_2 all components that are within distance 8 of all components already in g_2. Groups g_3, g_4, and so on, are then formed in the same way until all complete components are in some group.

Note that the distance between two groups is at least 9. Assume there are ℓ groups, and assume without loss of generality that the first k groups have diameter less than $\gamma/2$. These super-components will be called *small*. Focus attention on groups $g_{k+1}, \ldots, g_\ell$. Note that some of these groups may have diameter greater than 7γ. However, since no complete component has diameter greater than 3γ, there is a partition of these groups into new groups $g'_1, \ldots, g'_{k'}$ such that each of these groups has diameter between $\gamma/2$ and 6γ.

Groups $g_1, \ldots, g_k$ and $g'_1, \ldots, g'_{k'}$ together define the super-components the Prover will use. Note that the small super-components given by $g_1, \ldots, g_k$ each have distance at least 9 from any other super-component. The super-components $g'_1, \ldots, g'_{k'}$ are the *full-size components*. Note that by Property 3, there can be at most $r/(\gamma/2) \leq \gamma$ full-size components. Note also that this is the only part of the proof that will use the fact that $r \leq \gamma^2/2$.

It is not hard to see that when the partitioning is done to form the full-size components, it can be done so that the following property holds: For each g'_i, there exists a spanning tree for the super-component g'_i (where a spanning trees for a super-components is defined in the obvious way) such that the distance between any two of these spanning trees is at

least 3 (this can be argued by appealing to Observation 1). Fix such a spanning tree for each full-size component.

Isolating: We will show how to handle the case where no two full-size components have distance exactly 4 (i.e., distance is either 3 or at least 5); we leave out the case where some full-size components have distance 4 which involves similar reasoning but requires a somewhat more technical case-by-case analysis.

Define a node to be *on the boundary of a full-size component* if it is a boundary node of a simple component contained in the full-size component. We will now define two fractional vertex covers y^1 and y^2 such that $y = \frac{1}{3}y^1 + \frac{2}{3}y^2$. For every point i that is a boundary node of some simple component C' in a full-size component C we do the following: For every path (i, j_1, j_2) of length 2 (i.e., 2 edges) starting from i such that neither of j_1 or j_2 are in the spanning tree for C (such a path is called a *2-path coming out of a boundary node*) we will have the following assignments in y^1 and y^2: Either (a) the values of j_1 and j_2 are 1 and 0, respectively, in y^1 and $\frac{1}{2}$ and 1, respectively in y^2, or (b) the values of j_1 and j_2 are $\frac{2}{3}$ and 1, respectively, in y^1 and $\frac{2}{3}$ and $\frac{1}{2}$, respectively in y^2. Thus every path of length 2 out of a boundary node has two possible assignments in y^1 and y^2. For all remaining coordinates, have y^1 and y^2 be identical to y.

Lemma 4.5 *There exists a consistent way to decide what type of assignment to give to all 2-paths coming out of boundary nodes in full-sized components.*

PROOF: We make the following observation that will be crucial below: Since the diameter of a super-component is less than 7γ, no two 2-paths from a single component are adjacent.

Suppose no such consistent assignment existed. In particular, suppose that for every possible assignment there exists some boundary node i in a full-size component C and two 2-paths $p_1 = (i, j_1, j_2)$ and $p_2 = (i, j_1, j_3)$ coming out of i such that the following holds: If p_1 has an assignment of type (a), then p_2 must have an assignment of type (b), and vice versa.

For this causal relationship to hold, the following must be the case: If p_1 has an assignment of type (a), it forces the assignment of some adjacent 2-path (i', j_1', j_2') where i' is a boundary node in some other full-size component (Note: it cannot be from C by the above observation). In turn this forces the assignment of some 2-path adjacent to the previous 2-path coming out of yet another full-size component (it must be different by girth considerations). In turn this forces the assignment of some 2-path adjacent to the previous 2-path coming out of yet another full-size component, different from all full-size components involved so far. This chain of dependencies continues in this way. However, this chain can only continue for $k' \leq \gamma = g/28$ more steps (where k' is the number of full-size components) since after that there are no more full-size components to continue the chain. But then, this causal chain cannot reach p_2, contradicting the fact that p_1's assignment type influences p_2's assignment. □

It follows that y^1 and y^2 can be consistently defined.

Note that each full-size components has no edge to the rest of the graph in G_{y^1}; hence all full-size components are isolated in G_{y^1}. However, this is not necessarily true in G_{y^2}. To fix this we will define four vectors y^3, y^4, y^5 and y^6 such that y^2 is in the convex hull of these vectors and such that the full-size components are isolated the corresponding graphs for these vectors. These vectors are defined as follows: For every 2-path (i, j_1, j_2) with assignment of type (a), j_1 and j_2 are $\frac{1}{3}$ and 1, resp., in y^3, and are 1 and 1, resp., in y^4; for every 2-path (i, j_1, j_2) with assignment of type (a), j_1 and j_2 are $\frac{1}{3}$ and 1, resp., in y^5, and are 1 and 0, resp., in y^6. It can be verified that these vectors have the required properties and moreover, satisfy the VERTEX COVER constraints. Let $A = \{y^1, y^3, y^4, y^5, y^6\}$. By construction, $y \in conv(A)$.

It is straightforward to verify that the vectors in A satisfy Properties 1 and 3 for the next round.

Removing: As noted above, all components with diameter larger than $\gamma/2$ are disconnected in the graphs corresponding to the vectors in A. Fix a vector $z \in A$ and consider the subgraph G_C of G_z induced by some isolated component C. Since this isolated component has diameter less than the the girth g of G, it follows that C is a tree. In particular, C has an independent set of size $|C|/2$. It follows that z_C (i.e., z restricted to those coordinates indexed by nodes in C) is in the integral hull of the VERTEX COVER polytope for G_C. Indeed, this is true for all vectors in A and all isolated components.

But then, there exists a set Z of vectors which are (a) 0-1 on the subgraphs of the isolated components, (b) identical to some vector in A outside those components, and (c) $A \subseteq conv(Z)$. The Prover then lets $B_y = Z$.

5. Discussion

We feel that our methods should extend to proving that the integrality gap for VERTEX COVER relaxations is in fact $2 - o(1)$ after $\Omega(\log^2 n)$ rounds of LS tightening. Indeed, the use of the all-$\frac{2}{3}$ vector to prove our integrality gap of $1.5 - o(1)$ was motivated by our desire to keep our protection matrices and hence our girth correction strategies as simple as possible. An analogous proof using the all-$(\frac{1}{2} + \gamma)$ vector for some small $\gamma > 0$ (and hence yielding an integrality gap of $2/(1 + 2\gamma) - o(1)$) may also be possible; however, coming up with a strategy for putting up "fences"

will become much more complicated and likely be very difficult to analyze.

We also conjecture that a variation of our "fence" method might be able to yield integrality gaps for up to $\Omega(n^{1-\delta})$ rounds of LS for some $\delta > 0$. The current proof fails after $O(\log^2 n)$ rounds since Lemma 4.5 only works if there are $\Omega(\log n)$ super-components; this is not true after $\log^2 n$ rounds using the current Prover strategy. A different argument may be able to get around this deficiency. Note that under Khot's Unique Games conjecture [11], VERTEX COVER has no $2 - o(1)$ approximations [12]. In particular, Khot's conjecture implies that there must remain a large integrality gap even after n^δ rounds of LS tightenings for some $\delta > 0$.

Our lower bound argument does not yield any integrality gaps for LS_+ tightenings for VERTEX COVER (recall that unlike LS, the LS_+ procedure also includes a positive semi-definiteness constraint). Proving such lower bounds for VERTEX COVER remains a difficult open problem. Moreover, such lower bounds would arguably provide much stronger evidence about the true inapproximability of VERTEX COVER.

It would be interesting to see if our techniques can be used to prove integrality gaps for UNIQUE LABEL COVER (the problem from Khot's Unique Games conjecture) in the LS or LS_+ hierarchies. Such results could further support Khot's Unique Games conjecture, in turn providing further evidence about the true inapproximability of VERTEX COVER.

6. Acknowledgements

I would like to thank Sanjeev Arora for many helpful discussions on this topic as well as for his advice while preparing this paper. In addition, I would also like to thank Joshua Buresh-Oppenheim, Avner Magen and Toniann Pitassi for many helpful discussions.

I would also like to thank the anonymous referees for several helpful comments as well as for suggesting the inclusion of the Frost quotation in Section 2.1.

References

[1] M. Alekhnovich, S. Arora, and I. Tourlakis. Towards strong nonapproximability results in the Lovász-Schrijver hierarchy. In *Proceedings of the 37th Annual Symposium on the Theory of Computing*, New York, May 2005. ACM Press.

[2] S. Arora, B. Bollobás, and L. Lovász. Proving integrality gaps without knowing the linear program. In *Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS-02)*, pages 313–322, Los Alamitos, Nov. 16–19 2002.

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

[4] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proceedings of the thirty-sixth annual ACM Symposium on Theory of Computing (STOC-04)*, pages 222–231, New York, June 13–15 2004. ACM Press.

[5] J. Buresh-Oppenheim, N. Galesi, S. Hoory, A. Magen, and T. Pitassi. Rank bounds and integrality gaps for cutting planes procedures. In *Proceedings: 44th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2003, 11–14 October 2003, Cambridge, Massachusetts*, pages 318–327. pub-IEEE, 2003.

[6] W. Cook and S. Dash. On the matrix-cut rank of polyhedra. *Mathematics of Operations Research*, 26(1):19–30, 2001.

[7] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1), 2005.

[8] U. Feige and R. Krauthgamer. The probable value of the Lovász–Schrijver relaxations for maximum independent set. *SIAM Journal on Computing*, 32(2):345–370, 2003.

[9] M. X. Goemans and L. Tunçel. When does the positive semidefiniteness constraint help in lifting procedures? *Mathematics of Operations Research*, 26(4):796–815, 2001.

[10] M. X. Goemans and D. P. Williamson. .878-approximation algorithms for MAX CUT and MAX 2SAT. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing, STOC'94 (Montréal, Québec, Canada, May 23-25, 1994)*, pages 422–431, New York, 1994. ACM Press.

[11] S. Khot. On the power of unique 2-Prover 1-Round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC-02)*, pages 767–775, New York, May 19–21 2002. ACM Press.

[12] S. Khot and O. Regev. Vertex cover might be hard to approximate to within $2 - \varepsilon$. In *18th IEEE Conference on Computational Complexity*, page 379, 2003.

[13] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, May 1991.

[14] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, Aug. 1990.

[15] T. Stephen and L. Tunçel. On a representation of the matching polytope via semidefinite liftings. *Mathematics of Operations Research*, 24(1):1–7, 1999.

[16] I. Tourlakis. Towards optimal integrality gaps for hypergraph vertex cover in the Lovász-Schrijver hierarchy. In *8th. International Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 9th International Workshop on Randomization and Computation*, pages 233–244, 2005.