Journal on Satisfiability, Boolean Modeling and Computation 4 (2007) 15-31

# An Extended Semidefinite Relaxation for Satisfiability

**Miguel F. Anjos*** anjos@stanfordalumni.org
*Department of Management Sciences,*
*Faculty of Engineering,*
*University of Waterloo,*
*Waterloo, Ontario, Canada N2L 3G1*

## Abstract

This paper proposes a new semidefinite programming relaxation for the satisfiability problem. This relaxation is an extension of previous relaxations arising from the paradigm of partial semidefinite liftings for 0/1 optimization problems. The construction of the relaxation depends on a choice of permutations of the clauses, and different choices may lead to different relaxations. We then consider the Tseitin instances, a class of instances known to be hard for certain proof systems, and prove that for any choice of permutations, the proposed relaxation is exact for these instances, meaning that a Tseitin instance is unsatisfiable if and only if the corresponding semidefinite programming relaxation is infeasible.

Keywords: *satisfiability, semidefinite programming, discrete optimization, global optimization*

*Submitted August 2007; revised December 2007; published December 2007*

## 1. Introduction

This paper is about the application of semidefinite programming to the satisfiability (SAT) problem. Semidefinite programming (SDP) refers to the class of optimization problems where a linear function of a matrix variable $X$ is maximized (or minimized) subject to linear constraints on the elements of $X$ and the additional constraint that $X$ must be positive semidefinite. This includes linear programming problems as a special case, namely when all the matrices involved are diagonal. The handbook [41] provides an excellent coverage of SDP as well as an extensive bibliography covering the literature up to the year 2000. The impact of SDP in combinatorial optimization has been particularly significant, including such breakthroughs as the theta number of Lovász for the maximum stable set problem [31], and the approximation algorithms of Goemans and Williamson for the maximum-cut and maximum-satisfiability problems [21]. A variety of polynomial-time interior-point algorithms for solving SDPs (to within a fixed prescribed precision) have been proposed in the literature, and several excellent solvers for SDP are now available.

The fact that a propositional logic formula can be expressed in optimization terms as a feasibility problem involving polynomial equations and inequalities is well known, and dates back at least to the pioneering work of Williams [40] and Blair et al. [14]. Optimization-

* Research supported by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada.

based approaches to propositional logic originally considered formulations of SAT and maximum satisfiability (MAX-SAT) as 0/1 integer linear programming problems that can then be relaxed by allowing the 0/1 variables to take any real value between 0 and 1, thus yielding a linear programming relaxation that can be solved efficiently. For some types of problems, such as Horn formulas and their generalizations, deep connections have been established between the SAT problem and its linear programming relaxation (see e.g. [15]). The book of Chandru and Hooker [17] provides an excellent coverage of results linking logical inference and optimization.

One line of research made use of SDP to obtain the best known approximation algorithms for MAX-$k$-SAT problems[1]. Given a set of propositional clauses all of length at most $k$ in conjunctive normal form, the MAX-$k$-SAT problem is to determine the largest number of clauses that can be satisfied simultaneously. Seminal work in this direction was done by Goemans and Williamson [21] who proposed an SDP-based approximation algorithm for the MAX-2-SAT problem with a 0.87856-approximation guarantee. Improved guarantees for MAX-2-SAT were subsequently obtained by other researchers, and the SDP-based approach was extended to MAX-3-SAT by Karloff and Zwick [27] whose 0.875-approximation algorithm is the best possible (unless $P = NP$). Further extensions have been proposed by Zwick [42], Halperin and Zwick [26], and Asano and Williamson [11], providing the best known approximation guarantees for MAX-$k$-SAT problems. The survey paper [5] provides a thorough overview of these results.

Most recently, van Maaren and van Norden [38, 39] consider the application of Hilbert's Positivstellensatz to MAX-SAT. The idea is to formulate MAX-SAT as a global polynomial optimization problem in such a way that it can then be relaxed to a sum-of-squares (SOS) problem, and the latter can be solved using SDP (under certain assumptions). The construction of the SOS problem depends on the choice of a basis of monomials to express the SOS. Van Maaren and van Norden [39] consider several choices of bases, and present detailed comparisons between the resulting SOS relaxations and the relaxations of Goemans and Williamson and of Karloff and Zwick.

Another line of recent research has focused on analyzing the complexity of Nullstellensatz- and Positivstellensatz-based proofs, including cutting-plane and Lovász-Schrijver methods, and generalizations thereof (see e.g. [16, 24, 33]). The Nullstellensatz proof system, which uses only polynomial equalities, was first considered in [13]. By considering systems of polynomial inequalities (instead of equations only), much more powerful proof systems are obtained. The first proof system based on inequalities was the cutting plane system of Gomory [22, 23] which uses linear inequalities, while a more recent approach based on the Lovász-Schrijver systems [32] allows the use of quadratic inequalities. These two successful techniques arose in the area of integer linear programming, and in particular the Lovász-Schrijver approach is an example of the use of lift-and-project methods for 0/1 optimization [12, 32, 34]. Another recent development in this area is the Lasserre hierarchy of semidefinite liftings for polynomial optimization problems [29], which can be applied in particular to 0/1 programming problems [30]. Semidefinite constraints may also be employed in the Lovász-Schrijver lifting scheme, but in a different manner from that of the Lasserre con-

1. The notation $k$-SAT refers to the instances of SAT for which all the clauses have length at most $k$.

struction, which is based on a higher-liftings paradigm for constructing SDP relaxations of polynomial optimization problems.

The idea underlying the higher liftings is closely related to the sum-of-squares approach mentioned above, and can be summarized as follows. Suppose that we have a discrete optimization problem on $n$ binary variables. The SDP relaxation in the space of symmetric matrices with rows and columns indexed by the $n$ binary variables is called a first lifting. This first lifting for the specific case of SAT is the Gap relaxation of de Klerk et al. [19, 20]. De Klerk et al. show that the Gap relaxation characterizes unsatisfiability for a class of SAT problems that includes the mutilated chessboard and pigeonhole instances, in the sense that the Gap relaxation is infeasible if and only if the corresponding instance of SAT is unsatisfiable. Rounding schemes and approximation guarantees for the Gap relaxation, as well as its behavior on $(2+p)$-SAT problems, were studied in [19].

The Gap relaxation can be extended along the lines of the second lifting for maximum-cut proposed by Anjos and Wolkowicz [9], and its generalization independently proposed by Lasserre [28, 30]. Given any SAT instance, one could use the SDP relaxations $Q_{K-1}$ (as defined in [28]) for $K = 1, 2, \ldots, n$ where the matrix variable of $Q_{K-1}$ has rows and columns indexed by all the subsets of variables with cardinality at most $K$. Hence for $K = 1$, we obtain the matrix variable of the Gap relaxation. The results in [30] imply that for $K = n$, the resulting SDP relaxation characterizes unsatisfiability of the SAT instance. However, this SDP problem has dimension exponential in $n$. This limitation motivates the study of *partial higher liftings*, where we consider SDP relaxations that have a much smaller matrix variable, as well as fewer linear constraints.

To obtain interesting partial liftings, there are two important requirements to satisfy. The first requirement is that the SDP problems arising from the partial liftings must have both the dimension of the matrix variable and the number of linear constraints depending linearly on the size of the SAT instance[2]. This is because the SDP problems arising from full higher liftings quickly become far too large for practical computation. For instance, even for second liftings (corresponding to $K = 2$) of maximum-cut problems, only instances with up to 27 binary variables were successfully solved in [1]. This requirement ensures that the partial liftings are amenable to practical computation. The second requirement is that the partial lifting construction should take advantage of the structure of the SAT instance. More specifically, we want to design general partial lifting procedures with the property that their structure directly follows from the structure of the SAT instance.

Partial lifting procedures for SAT that extend the Gap relaxation have been proposed by Anjos [2, 4, 6], and the closely related SOS approach for SAT was proposed by van Maaren and van Norden [38, 39]. All these constructions are defined explicitly rather than in terms of an iterative process increasingly tightening the relaxations. The explicitness of the resulting SDP problems is particularly advantageous from a computational point of view, and their practical performance has been analyzed in [2, 3, 38, 39].

This paper presents new progress in the study of partial semidefinite liftings for SAT. The contribution of this paper is twofold. First, we describe the construction of a new SDP relaxation for the general SAT problem. This relaxation is an extension of the SDP relaxations in [4, 39]. It is also related to the SDP relaxation in [6], but unlike that relaxation,

2. For an instance in conjunctive normal form, the size is basically determined by the number of clauses.

the relaxation proposed here is defined for any CNF formula. For this new SDP relaxation, the dimension of the matrix variable and the number of constraints are linear in the number of clauses in the instance, assuming that the length of the clauses is bounded from above by a given, fixed value.

Second, we show that the extended SDP relaxation is exact for the class of Tseitin instances of SAT. By being exact, we mean that the Tseitin instance is unsatisfiable if and only if the corresponding SDP problem is infeasible. Although this is not the first proof that Tseitin instances can be solved in polynomial-time (see e.g. [24]), and these instances are typically used as counter-examples to the effectiveness of proof systems, the theorem and its proof provide a greater understanding of how the SDP relaxation captures the global structure of these instances. We also note that the SDP approach not only establishes satisfiability or unsatisfiability, but also provides an explicit computational proof in polynomial-time. This proof comes in the form of a certificate of infeasibility consisting of a dual feasible solution which may be arbitrarily scaled while remaining feasible, so that the objective value of the dual problem becomes unbounded. Since the optimal value of the dual problem always provides a bound on the optimal value of the primal, it follows that the primal problem must be infeasible. One interesting open question is the design of an algorithm to extract the combinatorial information contained in the numerical certificate of infeasibility for the extended SDP relaxation corresponding to an unsatisfiable instance of SAT. This would likely provide a bridge between the numerical proof provided by an SDP solver, and more traditional approaches to proving unsatisfiability for SAT.

This paper is structured as follows. After introducing some notation in the remainder of this section, we recall some of the aforementioned SDP relaxations for SAT in Section 2. In Section 3, we present the construction of the new SDP relaxation. Then in Section 4 we recall the definition of the Tseitin instances of SAT, and in Section 5 we prove the main result of this paper, namely the characterization of unsatisfiability of the Tseitin instances by the new SDP relaxtion. Finally, Section 6 provides some closing comments, and outlines on-going and future research.

We consider the SAT problem for instances in conjunctive normal form (CNF). Such instances are specified by a set of variables $x_1, \ldots, x_n$ and a propositional formula $\Phi = \bigwedge_{j=1}^{m} C_j$, with each clause $C_j$ having the form $C_j = \bigvee_{i \in I_j} x_i \vee \bigvee_{k \in \bar{I}_j} \bar{x}_k$ where $I_j, \bar{I}_j \subseteq \{1, \ldots, n\}$, $I_j \cap \bar{I}_j = \emptyset$, and $\bar{x}_i$ denotes the negation of $x_i$. We assume without loss of generality that $|I_j \cup \bar{I}_j| \geq 2$ for every clause $C_j$. The SAT problem is: Given a satisfiability instance, is $\Phi$ satisfiable, that is, is there a truth assignment to the variables $x_1, \ldots, x_n$ such that $\Phi$ evaluates to TRUE? For $k \geq 2$, $k$-SAT refers to the instances of SAT for which all the clauses have length at most $k$.

We shall henceforth let TRUE be denoted by 1 and FALSE be denoted by $-1$. For clause $j$ and $i \in I_j \cup \bar{I}_j$, define

$$s_{j,i} := \begin{cases} 1, & \text{if} \quad i \in I_j \\ -1, & \text{if} \quad i \in \bar{I}_j \end{cases}$$

The SAT problem is now equivalent to the integer programming feasibility problem

$$\begin{array}{ll} \text{find} & x \in \{\pm 1\}^n \\ \text{s.t.} & \sum\limits_{i \in I_j \cup \bar{I}_j} s_{j,i} x_i \geq 2 - l(C_j), \quad j = 1, \ldots, m \end{array}$$

where $l(C_j) = |I_j \cup \bar{I}_j|$ denotes the number of literals in clause $C_j$. Clearly this problem is equivalent to the original SAT problem, and hence is in general NP-complete. Some special cases of SAT can be solved in polynomial-time using linear programming, see [18]. Special instances of SAT with certain constraints on the length of the clauses are often of interest, both theoretically and in practice; we refer the reader to the survey [25].

## 2. Previous SDP-Based Formulations and Relaxations for SAT

The initial study of the application of SDP to SAT was done by de Klerk, van Maaren, and Warners who introduced the Gap relaxation for SAT [19, 20]. The Gap relaxation for 3-SAT may be expressed as follows:

$$(R_1) \qquad \begin{array}{ll} \text{find} & X \in \mathcal{S}^{n+1} \\ \text{s.t.} & \\ & s_{j,i_1}\, s_{j,i_2}\, X_{i_1,i_2} - s_{j,i_1}\, X_{0,i_1} - s_{j,i_2}\, X_{0,i_2} + 1 = 0, \\ & \quad \text{where } \{i_1, i_2\} = I_j \cup \bar{I}_j, \ \text{if } l(C_j) = 2 \\ & s_{j,i_1}\, s_{j,i_2}\, X_{i_1,i_2} + s_{j,i_1}\, s_{j,i_3}\, X_{i_1,i_3} + s_{j,i_2}\, s_{j,i_3}\, X_{i_2,i_3} - s_{j,i_1}\, X_{0,i_1} \\ & \quad - s_{j,i_2}\, X_{0,i_2} - s_{j,i_3}\, X_{0,i_3} \leq 0, \\ & \quad \text{where } \{i_1, i_2, i_3\} = I_j \cup \bar{I}_j, \ \text{if } l(C_j) = 3 \\ & \operatorname{diag}(X) = e \\ & X \succeq 0 \end{array}$$

where $\mathcal{S}^n$ denotes the space of $n \times n$ square symmetric matrices, $\operatorname{diag}(X)$ represents a vector containing the diagonal elements of the matrix $X$, $e$ denotes the vector of all ones, and $X \succeq 0$ denotes that $X$ is positive semidefinite. This relaxation is based on the application of the elliptic approximations introduced in [37], and the linear constraints are obtained by expanding and linearizing the elliptic approximation for each clause. Using these elliptic approximations, it is straightforward to extend the Gap relaxation to SAT instances with any number of literals in each clause.

More recently, Anjos proposed two improved SDP relaxations that are able to detect unsatisfiability independently of the length of the clauses, and that inherit all the properties of the Gap relaxation. The general construction and analysis of these relaxations are presented in [2, 4]. We outline here the derivation of the stronger of the two relaxations, as it is the basis for the new relaxation in this paper.

By construction of the coefficients $s_{j,i}$, the clause is satisfied if and only if $s_{j,i}x_i$ equals 1 for at least one $i \in I_j \cup \bar{I}_j$, or equivalently, if $\prod\limits_{i \in I_j \cup \bar{I}_j} (1 - s_{j,i}x_i) = 0$. Applying [4, Proposition

1], we can formulate the satisfiability problem as follows:

$$
\begin{array}{ll}
\text{find} & x_1 \ldots, x_n \\
\text{s.t.} & \\
& \sum\limits_{t=1}^{l(C_j)} (-1)^{t-1} \left[ \sum\limits_{T \subseteq I_j \cup \bar{I}_j, |T|=t} \left( \prod\limits_{i \in T} s_{j,i} \right) \left( \prod\limits_{i \in T} x_i \right) \right] = 1, \quad j = 1, \ldots, m \\
& x_i^2 = 1, \quad i = 1, \ldots, n.
\end{array}
$$

Note that this formulation has one constraint per clause, corresponding to satisfiability of the clause, plus one constraint per variable, corresponding to integrality of the variable.

The next step is to formulate the problem in symmetric matrix space. Let $\mathcal{P}$ denote the set of nonempty sets $T \subseteq \{1, \ldots, n\}$ such that the term $\prod\limits_{i \in T} x_i$ appears in the above formulation. Also introduce new variables

$$
x_T := \prod_{i \in T} x_i,
$$

for each $T \in \mathcal{P}$, and thus define the rank-one matrix

$$
Y := \begin{pmatrix} 1 \\ x_{T_1} \\ \vdots \\ x_{T_{|\mathcal{P}|}} \end{pmatrix} \begin{pmatrix} 1 \\ x_{T_1} \\ \vdots \\ x_{T_{|\mathcal{P}|}} \end{pmatrix}^T,
$$

whose $|\mathcal{P}| + 1$ rows and columns are indexed by $\{\emptyset\} \cup \mathcal{P}$. By construction of $Y$, we have that $Y_{\emptyset,T} = x_T$ for all $T \in \mathcal{P}$. Using these new variables, and making use of the fact that the constraints

$$
\operatorname{diag}(Y) = e, \quad Y \succeq 0, \quad \operatorname{rank}(Y) = 1
$$

are equivalent to

$$
x_i^2 = 1, i = 1, \ldots, n \quad \text{and} \quad Y_{\emptyset,T} = \prod_{i \in T} x_i, \text{ for all } T
$$

(see e.g. [8]), we can formulate the SAT problem as:

$$
\begin{array}{ll}
\text{find} & Y \in \mathcal{S}^{1+|\mathcal{P}|} \\
\text{s.t.} & \\
& \sum\limits_{t=1}^{l(C_j)} (-1)^{t-1} \left[ \sum\limits_{T \subseteq I_j \cup \bar{I}_j, |T|=t} \left( \prod\limits_{i \in T} s_{j,i} \right) Y_{\emptyset,T} \right] = 1, \quad j = 1, \ldots, m \\
& \operatorname{diag}(Y) = e \\
& \operatorname{rank}(Y) = 1 \\
& Y \succeq 0.
\end{array} \tag{1}
$$

Relaxing this formulation by omitting the rank constraint would give an SDP relaxation for SAT.

However, we first add *redundant* constraints to this formulation[3]. To do this, observe that for every triple $T_1, T_2, T_3$ of subsets in $\mathcal{P}$ such that the symmetric difference of any two equals the third, the following three equations hold for every feasible $Y$ in (1):

$$Y_{\emptyset,T_1} = Y_{T_2,T_3}, \quad Y_{\emptyset,T_2} = Y_{T_1,T_3}, \quad \text{and} \quad Y_{\emptyset,T_3} = Y_{T_1,T_2}. \tag{2}$$

Since this is not necessarily true for the SDP relaxation (i.e., after the rank constraint is removed), we add some of these constraints explicitly to the SDP relaxation for the purpose of strengthening it. We choose to add the equations of the form (2) for all the triples $\{T_1, T_2, T_3\} \subseteq \mathcal{P}$ satisfying the symmetric difference condition and such that $(T_1 \cup T_2 \cup T_3) \subseteq (I_j \cup \bar{I}_j)$ for some clause $j$. The resulting SDP relaxation is:

$$\begin{array}{ll}
\text{find} & Y \in \mathcal{S}^{1+|\mathcal{P}|} \\
\text{s.t.} & \\
& \displaystyle\sum_{t=1}^{l(C_j)} (-1)^{t-1} \left[ \sum_{T \subseteq I_j \cup \bar{I}_j, |T|=t} \left( \prod_{i \in T} s_{j,i} \right) Y_{\emptyset,T} \right] = 1, \quad j = 1, \ldots, m \\
& Y_{\emptyset,T_1} = Y_{T_2,T_3}, \quad Y_{\emptyset,T_2} = Y_{T_1,T_3}, \text{ and } Y_{\emptyset,T_3} = Y_{T_1,T_2}, \ \forall \{T_1, T_2, T_3\} \subseteq \mathcal{P} \\
& \quad \text{such that } T_1 \Delta T_2 = T_3 \text{ and } (T_1 \cup T_2 \cup T_3) \subseteq (I_j \cup \bar{I}_j) \text{ for some clause } j \\
& \operatorname{diag}(Y) = e \\
& Y \succeq 0
\end{array} \tag{3}$$

where $T_i \Delta T_j$ denotes the symmetric difference of $T_i$ and $T_j$. For this SDP relaxation, the main theoretical result is:

**Theorem 1.** *[4] Given any propositional formula in CNF, consider the SDP relaxation (3). Then:*

- *If (3) is infeasible, then the formula is unsatisfiable.*
- *If (3) is feasible, and $Y$ is a feasible matrix such that* $\operatorname{rank} Y \leq 3$*, then a truth assignment satisfying the formula can be obtained from $Y$.* ■

If we had chosen $\mathcal{P}$ to contain all the subsets $I$ with $|I| \leq k$, where $k$ denotes the length of the longest clause in the SAT instance, and had added all the redundant constraints of the form $Y_{T_1,T_2} = Y_{T_3,T_4}$, where $\{T_1, T_2, T_3, T_4\} \subseteq \{\emptyset\} \cup \mathcal{P}$ and $T_1 \Delta T_2 = T_3 \Delta T_4$, then we would have obtained the Lasserre relaxation $Q_{k-1}$ for this problem. However, as mentioned earlier, the resulting SDP problem has a matrix variable of dimension $O(n^k)$, which is too large for computational purposes, even when $k = 2$. The objective is to strike a balance between using the full Lasserre relaxation, or a more reasonably sized relaxation that preserves as much as possible the strength of the full relaxation. For instance, whenever $k$ is bounded above by a small constant, the partial higher liftings approach yields an SDP relaxation with a much smaller matrix variable as well as fewer linear constraints corresponding to symmetric differences. Indeed, the matrix variable of (3) has dimension $O(m * 2^k) = O(m)$, the number of constraints is also $O(m)$, and although the SDP can have as many as $(\frac{1}{2}(2^k - 2)(2^k - 1) + 1)m$ linear constraints, the presence of common variables between different clauses means that it will typically have many fewer constraints.

3. This approach of adding redundant constraints to the problem formulation so as to tighten the resulting SDP relaxation is discussed in detail in [8].

## 3. The New Extended Semidefinite Relaxation

The relaxation (3) has essentially one constraint per clause, and the connections between the clauses are solely provided by the positive semidefiniteness constraint on the matrix of linearized terms. The idea is to extend the relaxation by adding rows and columns to the matrix variable in such a way that more connections are made between the clauses. In doing so, however, we still want to control the growth in the size of the SDP relaxation.

For this purpose, we choose one representative term per clause, namely

$$\prod_{i \in I_j \cup \bar{I}_j} x_i$$

for each clause $j$. Let $\tilde{m}$ denote the number of such terms. It is clear that $\tilde{m} \leq m$: since two or more clauses may be formed using exactly the same variables, $\tilde{m} < m$ may occur. These terms are already included in $\mathcal{P}$, therefore what we do is augment $\mathcal{P}$ by adding sets of variables representing pairwise products of these $\tilde{m}$ terms, so as to better capture the interactions between clauses. One idea would be to add all pairwise products, i.e., $\binom{\tilde{m}}{2}$ terms. However, the result would be a matrix with dimension quadratic in $\tilde{m}$, and hence potentially superlinear in $m$ as well.

To restrict the size of the matrix, we proceed as follows: Let $\mathcal{C}_0 = \{S | S = I_j \cup \bar{I}_j \text{ for some } j\}$; clearly $\mathcal{C}_0 \subset \mathcal{P}$ and $|\mathcal{C}_0| = \tilde{m}$. We wish to consider arbitrary pairings of the $\tilde{m}$ elements of $\mathcal{C}_0$, therefore we fix an ordering of the elements of $\mathcal{C}_0$, say: $S_0^{(1)}, S_0^{(2)}, \ldots, S_0^{(\tilde{m})}$. We denote the pairing using a permutation $\pi_0$ of $\{1, \ldots, \tilde{m}\}$ with the interpretation that the first two elements in the permutation are paired, then the next two, and so on. There may be a non-paired element at the end of the permutation if $\tilde{m}$ is odd.

Let $\pi_0$ thus represent a given pairing of the elements of $\mathcal{C}_0$. Using $\pi_0$, we define

$$\mathcal{C}_1 = \left\{ S_1^{(\lambda)} = S_0^{(\pi_0(2\lambda-1))} \Delta S_0^{(\pi_0(2\lambda))} | \lambda = 1, 2, \ldots, \lfloor \frac{\tilde{m}}{2} \rfloor \right\} \cup \left\{ S_0^{(\pi_0(\tilde{m}))} | \tilde{m} \text{ is odd} \right\}.$$

Clearly $|\mathcal{C}_1| = \lceil \frac{\tilde{m}}{2} \rceil$. Define $\mathcal{C}_2$ in a similar way using an arbitrary pairing of the elements of $\mathcal{C}_1$, and so on, until reaching $\mathcal{C}_L$ with only one set. Note that $L \leq \log_2 \tilde{m} \leq \log_2 m$.

Using the set of column indices $\mathcal{C} := \mathcal{P} \cup \bigcup_{\sigma=1}^{L} \mathcal{C}_\sigma$, we formulate the problem in symmetric matrix space by proceeding as above. Introducing a variable

$$x_T := \prod_{i \in T} x_i,$$

for each $T \in \mathcal{C}$, we define a rank-one matrix with $|\mathcal{C}| + 1$ rows and columns indexed by $\{\emptyset\} \cup \mathcal{C}$. Note that $|\mathcal{C}| \leq 2^{k+1} m$, where $k$ is again the length of the longest clause, and hence for fixed $k$ the size of the matrix variable remains linear in the dimension of the instance. Using these new variables, we can obtain a new relaxation of the SAT problem. As in Section 2, we tighten the resulting SDP relaxation by adding redundant constraints. We choose to add the equations of the form (2) for all the triples $\{T_1, T_2, T_3\} \subseteq \mathcal{P}$ satisfying the symmetric difference condition and such that $(T_1 \cup T_2 \cup T_3) \subseteq (I_j \cup \bar{I}_j)$ for some clause $j$, plus all the triples $\{S_\mu^{(\pi(2\lambda-1))}, S_\mu^{(\pi(2\lambda))}, S_{\mu+1}^{(\lambda)},\} \subseteq \bigcup_{\sigma=1}^{L} \mathcal{C}_\sigma$ such that $S_{\mu+1}^{(\lambda)} = S_\mu^{(\pi(2\lambda-1))} \Delta S_\mu^{(\pi(2\lambda))}$ for some $\mu$ and $\lambda$. The resulting SDP relaxation is:

$$
\begin{array}{ll}
\text{find} & Z \in \mathcal{S}^{1+|\mathcal{C}|} \\
\text{s.t.} & \\
& \displaystyle\sum_{t=1}^{l(C_j)} (-1)^{t-1} \left[ \sum_{T \subseteq I_j \cup \bar{I}_j, |T|=t} \left( \prod_{i \in T} s_{j,i} \right) Z_{\emptyset,T} \right] = 1, \quad j = 1, \ldots, m \\
& Z_{\emptyset,T_1} = Z_{T_2,T_3}, \quad Z_{\emptyset,T_2} = Z_{T_1,T_3}, \text{ and } Z_{\emptyset,T_3} = Z_{T_1,T_2}, \forall \{T_1, T_2, T_3\} \subseteq \mathcal{P} \\
& \quad \text{such that } T_1 \Delta T_2 = T_3 \text{ and } (T_1 \cup T_2 \cup T_3) \subseteq (I_j \cup \bar{I}_j) \text{ for some clause } j \\
& Z_{\emptyset, S_\mu^{(\pi_\mu(2\lambda-1))}} = Z_{S_\mu^{(\pi_\mu(2\lambda))}, S_{\mu+1}^{(\lambda)}}, \quad Z_{\emptyset, S_\mu^{(\pi_\mu(2\lambda))}} = Z_{S_\mu^{(\pi_\mu(2\lambda-1))}, S_{\mu+1}^{(\lambda)}}, \\
& \quad \text{and } Z_{\emptyset, S_{\mu+1}^{(\lambda)}} = Z_{S_\mu^{(\pi_\mu(2\lambda-1))}, S_\mu^{(\pi_\mu(2\lambda))}}, \forall \{S_\mu^{(\pi_\mu(2\lambda-1))}, S_\mu^{(\pi_\mu(2\lambda))}, S_{\mu+1}^{(\lambda)}\} \subseteq \mathcal{P} \\
& \quad \text{such that } S_{\mu+1}^{(\lambda)} = S_\mu^{(\pi_\mu(2\lambda-1))} \Delta S_\mu^{(\pi_\mu(2\lambda))} \text{ for some } \mu \text{ and } \lambda \\
& \operatorname{diag}(Z) = e \\
& Z \succeq 0.
\end{array}
\tag{4}
$$

We use a simple example to illustrate the construction of the relaxation, and the fact that different choices of permutations $\pi_\mu$ may lead to different relaxations.

**Example 1.** *Suppose we are given the CNF formula*

$$x_1 \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_1 \vee x_3) \wedge (x_2 \vee \bar{x}_3 \vee x_4) \wedge (x_3 \vee \bar{x}_4 \vee \bar{x}_5).$$

*We construct the new SDP relaxation as follows. For the first clause, we have the variable* $x_1$[4]. *For the second clause, we add the variables* $x_2$ *and* $x_{12}$. *Continuing in this fashion, we obtain*

$$\mathcal{P} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{1,2\}, \{1,3\}, \{2,3\}, \{2,4\}, \{3,4\}, \{3,5\}, \{4,5\}, \{2,3,4\}, \{3,4,5\}\}$$

*and*

$$\mathcal{C}_0 = \{\{1\}, \{1,2\}, \{1,3\}, \{2,3,4\}, \{3,4,5\}\}.$$

*If* $\pi_0 = (5\,3\,1\,4\,2)$, $\pi_1 = (3\,1\,2)$, *and* $\pi_2 = (1\,2)$, *then* $\mathcal{C}_1 = \{\{1,4,5\}, \{1,2,3,4\}, \{1,2\}\}$, $\mathcal{C}_2 = \{\{2,4,5\}, \{1,2,3,4\}\}$, *and* $\mathcal{C}_3 = \{\{1,3,5\}\}$. *Thus,* $\mathcal{C} = \mathcal{P} \cup \{\{1,4,5\}, \{1,2,3,4\}, \{2,4,5\}, \{1,3,5\}\}$, *and* $|\mathcal{C}| = 18$. *Therefore, the matrix variable* $Z$ *has dimension 19. As for the linear constraints, for each clause in the formula, we have one equality constraint:*

$$
\begin{array}{rc}
x_1 \Rightarrow & Z_{\emptyset,x_1} = 1; \\
(\bar{x}_1 \vee x_2) \Rightarrow & -Z_{\emptyset,x_1} + Z_{\emptyset,x_2} + Z_{\emptyset,x_{12}} = 1; \\
(\bar{x}_1 \vee x_3) \Rightarrow & -Z_{\emptyset,x_1} + Z_{\emptyset,x_3} + Z_{\emptyset,x_{13}} = 1; \\
(x_2 \vee \bar{x}_3 \vee x_4) \Rightarrow & Z_{\emptyset,x_2} - Z_{\emptyset,x_3} + Z_{\emptyset,x_4} + Z_{\emptyset,x_{23}} - Z_{\emptyset,x_{24}} + Z_{\emptyset,x_{34}} - Z_{\emptyset,x_{234}} = 1; \\
(x_3 \vee \bar{x}_4 \vee \bar{x}_5) \Rightarrow & Z_{\emptyset,x_3} - Z_{\emptyset,x_4} + Z_{\emptyset,x_5} + Z_{\emptyset,x_{34}} + Z_{\emptyset,x_{35}} - Z_{\emptyset,x_{45}} + Z_{\emptyset,x_{345}} = 1.
\end{array}
$$

*The other linear constraints enforce the structure of* $Z$ *as depicted in Figure 1. The elements of* $Z$ *denoted by asterisks in Figure 1 are not involved in any of the linear equality constraints,*

$$
\begin{pmatrix}
1 & x_1 & x_2 & x_3 & x_4 & x_5 & x_{12} & x_{13} & x_{23} & x_{24} & x_{34} & x_{35} & x_{45} & x_{234} & x_{345} & x_{145} & x_{1234} & x_{245} & x_{135} \\
 & 1 & x_{12} & x_{13} & * & * & x_2 & x_3 & * & * & * & x_{135} & x_{145} & x_{1234} & * & x_{45} & x_{234} & * & x_{35} \\
 & & 1 & x_{23} & x_{24} & * & x_1 & * & x_3 & x_4 & x_{234} & * & x_{245} & x_{34} & * & * & * & x_{45} & * \\
 & & & 1 & x_{34} & x_{35} & * & x_1 & x_2 & x_{234} & x_4 & x_5 & x_{345} & x_{24} & x_{45} & * & * & * & * \\
 & & & & 1 & x_{45} & * & * & x_{234} & x_2 & x_3 & x_{345} & x_5 & x_{23} & x_{35} & * & * & * & * \\
 & & & & & 1 & * & * & * & * & x_{345} & x_3 & x_4 & * & x_{34} & * & * & x_{24} & x_{13} \\
 & & & & & & 1 & x_{23} & x_{13} & * & x_{1234} & * & * & * & * & x_{245} & x_{34} & x_{145} & * \\
 & & & & & & & 1 & x_{12} & x_{1234} & * & * & * & * & x_{145} & x_{345} & x_{24} & * & x_5 \\
 & & & & & & & & 1 & x_{34} & x_{24} & * & * & x_4 & x_{245} & * & * & x_{345} & * \\
 & & & & & & & & & 1 & x_{23} & * & * & x_3 & * & * & x_{13} & x_5 & * \\
 & & & & & & & & & & 1 & x_{45} & x_{35} & x_2 & x_5 & x_{135} & x_{12} & * & x_{145} \\
 & & & & & & & & & & & 1 & x_{34} & x_{245} & x_4 & * & * & x_{234} & x_1 \\
 & & & & & & & & & & & & 1 & * & x_3 & x_1 & * & x_2 & * \\
 & & & & & & & & & & & & & 1 & * & * & x_1 & x_{35} & * \\
 & & & & & & & & & & & & & & 1 & x_{13} & * & x_{23} & * \\
 & & & & & & & & & & & & & & & 1 & * & x_{12} & x_{34} \\
 & & & & & & & & & & & & & & & & 1 & x_{135} & x_{245} \\
 & & & & & & & & & & & & & & & & & 1 & x_{1234} \\
 & & & & & & & & & & & & & & & & & & 1
\end{pmatrix}
$$

**Figure 1.** Upper triangle of the symmetric matrix $Z$ for Example 1

*although they are of course constrained by the positive semidefiniteness constraint. The resulting SDP relaxation is:*

$$
\begin{array}{ll}
\text{find} & Z \in \mathcal{S}^{19} \\
\text{s.t.} & \\
 & Z_{\emptyset,x_1} = 1 \\
 & -Z_{\emptyset,x_1} + Z_{\emptyset,x_2} + Z_{\emptyset,x_{12}} = 1 \\
 & -Z_{\emptyset,x_1} + Z_{\emptyset,x_3} + Z_{\emptyset,x_{13}} = 1 \\
 & Z_{\emptyset,x_2} - Z_{\emptyset,x_3} + Z_{\emptyset,x_4} + Z_{\emptyset,x_{23}} - Z_{\emptyset,x_{24}} + Z_{\emptyset,x_{34}} - Z_{\emptyset,x_{234}} = 1 \\
 & Z_{\emptyset,x_3} - Z_{\emptyset,x_4} + Z_{\emptyset,x_5} + Z_{\emptyset,x_{34}} + Z_{\emptyset,x_{35}} - Z_{\emptyset,x_{45}} + Z_{\emptyset,x_{345}} = 1 \\
 & Z \text{ as in Figure } 1 \\
 & Z \succeq 0.
\end{array}
$$

*If we instead used $\pi_0 = (1\,3\,2\,4\,5)$, $\pi_1 = (1\,3\,2)$, and $\pi_2 = (1\,2)$, then $\mathcal{C}_1 = \{\{3\}, \{1,3,4\}, \{3,4,5\}\}$, $\mathcal{C}_2 = \{\{4,5\}, \{1,3,4\}\}$, and $\mathcal{C}_3 = \{\{1,3,5\}\}$, and $\mathcal{C} = \mathcal{P} \cup \{\{1,3,4\}, \{1,3,5\}\}$. Thus, $|\mathcal{C}| = 16$ and the matrix variable $Z$ would have dimension 17.*

Before concluding this section, we point out that while the matrix variable in (3) has the same dimensions as that of the SOS relaxation obtained using $M_{pt}$, the largest monomial basis considered in [39]), the number and structure of the linear constraints differs. Van Maaren and van Norden [39] present computational results supporting the claim that (3) is generally solved more quickly, but is weaker than the SOS relaxation using $M_{pt}$. Since

4. We use $x_1$ as shorthand notation for $x_{\{1\}}$, $x_{12}$ as shorthand for $x_{\{1,2\}}$, and so on.

the relaxation proposed here is a strengthening of (3), it is not immediately clear how its strength compares with that of the SOS relaxation using $M_{pt}$. A careful comparison will be the subject of future research.

Our objective here is to demonstrate the improvement of this new relaxation over (3). For this purpose, we recall in the next section the well-known class of SAT instances known as Tseitin instances. We then prove in Section 5 that this new relaxation characterizes unsatifiability for these instances, for any choice of permutations $\pi_\mu$.

## 4. Definition of the Tseitin Instances

First, we define the parity problem. Such a problem consists of a collection of statements about the parity of a given set of Boolean variables. Each parity statement has the form

$$x_1 \oplus x_2 \oplus \ldots \oplus x_p = r, \tag{5}$$

where $\oplus$ denotes exclusive or. Therefore, $r = 0$ denotes that an even number of the variables involved is TRUE, while $r = 1$ denotes that an odd number of them is TRUE. The problem is to determine if all the statements can be satisfied simultaneously.

Each parity statement (5) is equivalent to a conjunction of $2^{p-1}$ disjunctive clauses. The structure of the clauses depends on the value of $r$:

- if $r = 0$ then the conjunction consists of all possible clauses on the $p$ variables with an odd number of negated variables; and

- if $r = 1$ then the conjunction consists of all possible clauses on the $p$ variables with an even number of negated variables.

To build a Tseitin instance of SAT, we fix a connected graph $G = (V, E)$ with each vertex $v_i \in V$ labelled with a value $t(v_i) \in \{0, 1\}$. Then we introduce a Boolean variable $x_{i,j}$ for each $(i, j) \in E$, and let each $v_i \in V$ give rise to the conjunction of $2^{\deg(v_i)-1}$ clauses corresponding to the parity statement

$$\bigoplus_{\nu \in N(v_i)} x_{v_i,\nu} = t(v_i),$$

where $N(v_i) \subset V$ denotes the vertices connected to $v_i$ by an edge. It is straightforward to check that the SAT instance obtained from the conjunction of all these clauses is satisfiable if and only if $\sum_{v_i \in V} t(v_i)$ is even.

If $G$ is a toroidal grid graph, the new SDP relaxation is very closely related to, but not the same as, the SDP relaxation in [6]. Tseitin [35] showed that proving unsatisfiability of such instances using regular resolution requires a superpolynomial number of resolution steps. Tseitin's result was extended to general resolution in [36] using so-called expander graphs, a more general class of graphs. Tseitin instances are now typically defined in terms of expander graphs (see e.g. [24]). The main result in the next section makes no assumption on the structure of $G$, and hence applies to expander graphs as a special case.

## 5. Proof of Characterizability of Tseitin Instances by the SDP Relaxation

For a Tseitin instance of SAT, each vertex $v_i$ is associated with the set $VN(v_i) := \{x_{v_i,\nu} | \nu \in N(v_i)\}$ of cardinality $\deg(v_i)$, and contributes $2^{\deg(v_i)-1}$ clauses to the SAT instance, with the structure of the clauses being determined by the value of $t(v_i)$:

- if $t(v_i) = 0$ then $v_i$ contributes all possible clauses of length $\deg(v_i)$ on the variables in $VN(v_i)$ with an odd number of negated variables; and
- if $t(v_i) = 1$ then $v_i$ contributes all possible clauses of length $\deg(v_i)$ on the variables in $VN(v_i)$ with an even number of negated variables.

We denote the $2^{\deg(v_i)-1}$ clauses thus obtained by $C_\tau(v_i), \tau = 1, \ldots, 2^{\deg(v_i)-1}$. Hence there are $\sum_{i=1}^{|V|} 2^{\deg(v_i)-1}$ clauses in the SAT instance. Let $J_\tau(v_i) \subseteq VN(v_i)$ denote the set of variables negated in clause $\tau$ corresponding to vertex $v_i$. Then by construction,

$$|J_\tau(v_i)| \text{ is odd for every } \tau = 1, \ldots, 2^{\deg(v_i)-1} \text{ if } t(v_i) = 0,$$
$$\text{and}$$
$$|J_\tau(v_i)| \text{ is even for every } \tau = 1, \ldots, 2^{\deg(v_i)-1} \text{ if } t(v_i) = 1.$$

For clause $C_\tau(v_i)$ and variable $x_{v_i,\nu}$, define

$$s_\tau(v_i, \nu) := \begin{cases} 1, & \text{if } x_{v_i,\nu} \notin J_\tau(_i) \\ -1, & \text{if } x_{v_i,\nu} \in J_\tau(v_i) \end{cases}$$

From the structure of the clauses, it is easy to verify that

$$\prod_{\nu \in N(v_i)} s_\tau(v_i, \nu) = (-1)^{t(v_i)+1}, \text{ for each } \tau, \text{ and } i = 1, \ldots, |V|,$$

and that

$$\sum_{\tau=1}^{2^{\deg(v_i)-1}} \left( \prod_{\nu \in T} s_\tau(v_i, \nu) \right) = 0 \tag{6}$$

for each subset $T \neq \emptyset$, $T \subsetneq N(v_i)$, and for each $i = 1, \ldots, |V|$.

We will make use of the following two lemmata.

**Lemma 1.** *If $Z$ is feasible for the SDP relaxation (4) of a Tseitin instance, then*

$$Z_{\emptyset, VN(v_i)} = (-1)^{|N(v_i)|+t(v_i)}, \quad i = 1, \ldots, |V|.$$

**Proof:** We consider the $2^{\deg(v_i)-1}$ constraints in the SDP relaxation of the form

$$\sum_{t=1}^{l(C_j)} (-1)^{t-1} \left[ \sum_{T \subseteq I_j \cup \bar{I}_j, |T|=t} \left( \prod_{i \in T} s_{j,i} \right) Z_{\emptyset,T} \right] = 1$$

for fixed $i$. It is clear that $l(C_j) = |N(v_i)|$ and $I_j \cup \bar{I}_j = VN(v_i)$ for each of these constraints. Indexing the constraints over $\tau$ instead of $j$, and summing over $\tau$, we obtain:

$$\sum_{\tau=1}^{2^{\deg(v_i)-1}} \left( \sum_{t=1}^{|N(v_i)|} (-1)^{t-1} \left[ \sum_{T \subseteq VN(v_i), |T|=t} \left( \prod_{\nu \in T} s_\tau(v_i, \nu) \right) Z_{\emptyset,T} \right] \right) = 2^{\deg(v_i)-1}$$

which implies

$$\sum_{t=1}^{|N(v_i)|} (-1)^{t-1} \left[ \sum_{T \subseteq VN(v_i), |T|=t} \left( \sum_{\tau=1}^{2^{\deg(v_i)-1}} \left( \prod_{\nu \in T} s_\tau(v_i, \nu) \right) Z_{\emptyset,T} \right) \right] = 2^{\deg(v_i)-1}.$$

But by (6) above, the terms $\sum_{\tau=1}^{2^{\deg(v_i)-1}} \left( \prod_{\nu \in T} s_\tau(v_i, \nu) \right)$ all equal zero, except when $T = VN(v_i)$. Therefore, we have

$$(-1)^{|N(v_i)|-1} \left( 2^{\deg(v_i)-1} (-1)^{t(v_i)+1} Z_{\emptyset, VN(v_i)} \right) = 2^{\deg(v_i)-1}$$

and hence

$$Z_{\emptyset,T} = (-1)^{|N(v_i)|+t(v_i)}.$$

■

**Lemma 2.** *[7, Lemma 3.9] Suppose* $\begin{pmatrix} 1 & a & b \\ a & 1 & c \\ b & c & 1 \end{pmatrix} \succeq 0$. *Then* $a^2 = 1 \Rightarrow c = a\,b$.

We now prove the main theoretical result in this paper.

**Theorem 2.** *A Tseitin instance of SAT is unsatisfiable if and only if the corresponding SDP relaxation (4) is infeasible for any choice of permutations $\pi_\mu$.*

**Proof:** Sufficiency is clear, since any model for the SAT instance yields a feasible (rank-one) matrix.

We prove necessity by contradiction. Suppose that the SAT instance is unsatisfiable, and that $Z$ is feasible for the SDP problem. For a Tseitin instance, it is clear that $\tilde{m} = |V|$, and by construction of the SDP relaxation, $\mathcal{C}_0 = \{VN(v_i) | v_i \in V\}$, and $3 \leq |\mathcal{C}_{L-1}| \leq 4$.

Without loss of generality, we assume that $S_0^{(\lambda)} = VN(v_\lambda), \lambda = 1, \ldots, |V|$. By Lemma 1, $Z_{\emptyset, S_0^{(\lambda)}} = (-1)^{|N(v_\lambda)|+t(v_\lambda)}$ for every $S_0^{(\lambda)} \in \mathcal{C}_0$, $\lambda = 1, \ldots, |V|$. Now,

$$\begin{aligned}
Z_{\emptyset, S_1^{(\lambda)}} &= Z_{S_0^{(\pi_0(2\lambda-1))}, S_0^{(\pi_0(2\lambda))}}, \text{ by construction of the SDP relaxation} \\
&= Z_{\emptyset, S_0^{(\pi_0(2\lambda-1))}} Z_{\emptyset, S_0^{(\pi_0(2\lambda))}}, \text{ by Lemma 2} \\
&= (-1)^{|N(v_{\pi_0(2\lambda-1)})|+t(v_{\pi_0(2\lambda-1)})} (-1)^{|N(v_{\pi_0(2\lambda)})|+t(v_{\pi_0(2\lambda)})} \\
&= (-1)^{\deg(v_{\pi_0(2\lambda-1)})+\deg(v_{\pi_0(2\lambda)})+t(v_{\pi_0(2\lambda-1)})+t(v_{\pi_0(2\lambda)})}
\end{aligned}$$

and similarly for $Z_{\emptyset,S_2^{(\lambda)}}, \ldots, Z_{\emptyset,S_{L-1}^{(\lambda)}}$, for all applicable $\lambda$.

We now have two cases. If $|\mathcal{C}_{L-1}| = 3$, then

$$Z_{\emptyset,S_L} = Z_{S_{L-1}^{(1)},S_{L-1}^{(2)}} = Z_{\emptyset,S_{L-1}^{(1)}} Z_{\emptyset,S_{L-1}^{(2)}}$$

and also

$$Z_{\emptyset,S_L} = Z_{\emptyset,S_{L-1}^{(3)}}$$

therefore

$$Z_{\emptyset,S_{L-1}^{(1)}} Z_{\emptyset,S_{L-1}^{(2)}} = Z_{\emptyset,S_{L-1}^{(3)}}$$

which implies

$$(-1)^{\sum_{i=1}^{|V|} \deg(v_i) + \sum_{i=1}^{|V|} t(v_i)} = 1.$$

But $\sum_{i=1}^{|V|} \deg(v_i) = 2|E|$, and since $\sum_{i=1}^{|V|} t(v_i)$ is odd by assumption, we have a contradiction.

If $|\mathcal{C}_{L-1}| = 4$, then we similarly deduce that

$$Z_{\emptyset,S_{L-1}^{(1)}} Z_{\emptyset,S_{L-1}^{(2)}} = Z_{\emptyset,S_{L-1}^{(3)}} Z_{\emptyset,S_{L-1}^{(4)}}$$

which also implies

$$(-1)^{\sum_{i=1}^{|V|} \deg(v_i) + \sum_{i=1}^{|V|} t(v_i)} = 1,$$

and also results in a contradiction.

Hence $Z$ cannot exist, and the SDP problem must be infeasible. ∎

## 6. Concluding Remarks and Future Research

From a theoretical point of view, an intriguing open question is how to extract the combinatorial information contained in the SDP certificate of infeasibility for the extended SDP relaxation corresponding to an unsatisfiable instance of SAT. This would likely provide a bridge between the numerical proof provided by an SDP solver, and more conventional means of proving unsatisfiability for SAT.

From a computational point of view, we are currently studying the computational properties of the extended SDP relaxation for a variety of instances of SAT. It seems likely that the choice of permutations $\pi_\mu$ for specific instances will have an impact on the practical performance of the SDP relaxation. The ultimate objective of this research is an efficient SDP-based algorithm for the solution of large satisfiability instances.

## Acknowledgments

The author wishes to thank two anonymous referees for their constructive comments that helped improve the paper.

## References


[1] M.F. Anjos. *New Convex Relaxations for the Maximum Cut and VLSI Layout Problems*. PhD thesis, University of Waterloo, 2001. Published online at http://etd.uwaterloo.ca/etd/manjos2001.pdf.

[2] M.F. Anjos. On semidefinite programming relaxations for the satisfiability problem. *Math. Meth. Oper. Res.*, **60**(3), 2004.

[3] M.F. Anjos. Proofs of unsatisfiability via semidefinite programming. In *Operations Research Proceedings 2003*, pages 308–315, Springer, Berlin, 2004.

[4] M.F. Anjos. An improved semidefinite programming relaxation for the satisfiability problem. *Math. Program.*, **102**(3):589–608, 2005.

[5] M.F. Anjos. Semidefinite optimization approaches for satisfiability and maximum-satisfiability problems. *J. on Satisfiability, Boolean Modeling and Computation*, **1**(1):1–47, 2005.

[6] M.F. Anjos. An explicit semidefinite characterization of satisfiability for Tseitin instances on toroidal grid graphs. *Ann. Math. Artif. Intell.*, **47**(3-4):1–14, 2006.

[7] M.F. Anjos and H. Wolkowicz. Geometry of semidefinite max-cut relaxations via matrix ranks. *J. Comb. Optim.*, **6**(3):237–270, 2002.

[8] M.F. Anjos and H. Wolkowicz. Semidefinite programming for discrete optimization and matrix completion problems. *Discrete Appl. Math.*, **123**(1–2):513–577, 2002.

[9] M.F. Anjos and H. Wolkowicz. Strengthened semidefinite relaxations via a second lifting for the max-cut problem. *Discrete Appl. Math.*, **119**(1–2):79–106, 2002.

[10] B. Apswall, M.F. Plass, and R.E. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Inform. Process. Lett.*, **8**(3):121–123, 1979.

[11] T. Asano and D.P. Williamson. Improved approximation algorithms for MAX SAT. *J. Algorithms*, **42**(1):173–202, 2002.

[12] E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Math. Program.*, **58**(3, Ser. A):295–324, 1993.

[13] P. Beame, R. Impagliazzo, J. Krajíček, T. Pitassi, and P. Pudlák. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. *Proc. London Math. Soc. (3)*, **73**(1):1–26, 1996.

[14] C. E. Blair, R. G. Jeroslow, and J. K. Lowe. Some results and experiments in programming techniques for propositional logic. *Comput. Oper. Res.*, **13**(5):633–645, 1986.

[15] E. Boros, P.L. Hammer, and X. Sun. Recognition of $q$-Horn formulae in linear time. *Discrete Appl. Math.*, **55**(1):1–13, 1994.

[16] J. Buresh-Oppenheim, N. Galesi, S. Hoory, A. Magen, and T. Pitassi. Rank bounds and integrality gaps for cutting plane procedures. In *Proc. FOCS 2003*, pages 318–327, October 2003.

[17] V. Chandru and J. Hooker. *Optimization Methods for Logical Inference.* Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., New York, 1999.

[18] M. Conforti and G. Cornuéjols. A class of logic problems solvable by linear programming. *J. ACM*, **42**(5):1107–1113, 1995.

[19] E. de Klerk and H. van Maaren. On semidefinite programming relaxations of $(2+p)$-SAT. *Ann. Math. Artif. Intell.*, **37**(3):285–305, 2003.

[20] E. de Klerk, H. van Maaren, and J.P. Warners. Relaxations of the satisfiability problem using semidefinite programming. *J. Automat. Reason.*, **24**(1-2):37–65, 2000.

[21] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.*, **42**(6):1115–1145, 1995.

[22] R.E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bull. Amer. Math. Soc.*, **64**:275–278, 1958.

[23] R.E. Gomory. An algorithm for integer solutions to linear programs. In *Recent advances in mathematical programming*, pages 269–302. McGraw-Hill, New York, 1963.

[24] D. Grigoriev, E.A. Hirsch, and D.V. Pasechnik. Complexity of semialgebraic proofs. *Moscow Math. J.*, **2**(4):647–679, 2002.

[25] J. Gu, P.W. Purdom, J. Franco, and B.W. Wah. Algorithms for the satisfiability (SAT) problem: a survey. In *Satisfiability problem: theory and applications (Piscataway, NJ, 1996)*, pages 19–151. Amer. Math. Soc., Providence, RI, 1997.

[26] E. Halperin and U. Zwick. Approximation algorithms for MAX 4-SAT and rounding procedures for semidefinite programs. *J. Algorithms*, **40**(2):184–211, 2001.

[27] H. Karloff and U. Zwick. A 7/8-approximation algorithm for MAX 3SAT? In *Proc. of 38th FOCS*, pages 406–415, 1997.

[28] J.B. Lasserre. Optimality conditions and LMI relaxations for 0-1 programs. Technical report, LAAS-CNRS, Toulouse, France, 2000.

[29] J.B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, **11**(3):796–817 (electronic), 2000/01.

[30] J.B. Lasserre. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM J. Optim.*, **12**(3):756–769 (electronic), 2002.

[31] L. Lovász. On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory*, **25**(1):1–7, 1979.

[32] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, **1**(2):166–190, 1991.

[33] P.A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Math. Program.*, **96**(2, Ser. B):293–320, 2003.

[34] H.D. Sherali and W.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.

[35] G.S. Tseitin. On the complexity of derivation in propositional calculus. In A.O. Slisenko, editor, *Structures in Constructive Mathematics and Mathematical Logic, Part II*, Seminars in Mathematics (translated from Russian), pages 115–125. Steklov Mathematical Institute, 1968.

[36] A. Urquhart. Hard examples for resolution. *J. Assoc. Comput. Mach.*, **34**(1):209–219, 1987.

[37] H. van Maaren. Elliptic approximations of propositional formulae. *Discrete Appl. Math.*, **96/97**:223–244, 1999.

[38] H. van Maaren and L. van Norden. Sums of squares, satisfiability and maximum satisfiability. In *SAT 2005*, volume **3569** of *Lecture Notes in Comput. Sci.*, pages 294–308. Springer-Verlag, Berlin, 2005.

[39] H. van Maaren, L. van Norden, and M.J.H. Heule.
Sums of squares based approximation algorithms for MAX-SAT.
*JSAT Addendum*, http://jsat.ewi.tudelft.nl/addendum.html, July 2007.

[40] H.P. Williams. Logical problems and integer programming. *Bull. Inst. Math. Appl.*, **13**(1):18–20, 1977.

[41] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming.* Kluwer Academic Publishers, Boston, MA, 2000.

[42] U. Zwick. Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems. In *Proceedings of 31st STOC*, pages 679–687, 1999.