

Semidefinite Optimization Approaches for Satisfiability and Maximum-Satisfiability Problems

Miguel F. Anjos

anjos@stanfordalumni.org

*Department of Management Sciences,
Faculty of Engineering,
University of Waterloo,
Waterloo, Ontario, Canada N2L 3G1*

Abstract

Semidefinite optimization, commonly referred to as semidefinite programming, has been a remarkably active area of research in optimization during the last decade. For combinatorial problems in particular, semidefinite programming has had a truly significant impact. This paper surveys some of the results obtained in the application of semidefinite programming to satisfiability and maximum-satisfiability problems. The approaches presented in some detail include the ground-breaking approximation algorithm of Goemans and Williamson for MAX-2-SAT, the Gap relaxation of de Klerk, van Maaren and Warners, and strengthenings of the Gap relaxation based on the Lasserre hierarchy of semidefinite liftings for polynomial optimization problems. We include theoretical and computational comparisons of the aforementioned semidefinite relaxations for the special case of 3-SAT, and conclude with a review of the most recent results in the application of semidefinite programming to SAT and MAX-SAT.

KEYWORDS: *satisfiability, maximum-satisfiability, semidefinite programming, semidefinite optimization, approximation algorithms, semidefinite relaxations, lifting procedures.*

Submitted February 2005; revised July 2005; published August 2005

1. Introduction

Semidefinite optimization, commonly referred to as semidefinite programming (SDP), has been a remarkably active area of research in optimization during the last decade. The rapidly growing interest is likely due to a convergence of several developments: the extension to SDP of efficient interior-point algorithms for linear programming; the richness of the underlying optimization theory; and the recognition that SDP problems arise in several areas of applications, such as linear control theory, signal processing, robust optimization, statistics, finance, polynomial programming, and combinatorial optimization. The handbook [68] 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 is particularly significant, including such breakthroughs as the theta number of Lovász for the maximum stable set problem [47], and the approximation algorithms of Goemans and Williamson for the maximum-cut and maximum-satisfiability problems [24].

The focus of this paper is on recent progress in the application of SDP to satisfiability (SAT) and maximum-satisfiability (MAX-SAT) problems. This is one of the latest developments in the long history of interplay between optimization and logical inference, which dates back at least to the pioneering work of Williams, Jeroslow, and others (see e.g. [12, 38, 66, 67]). The optimization perspective on propositional logic has mostly focussed on the formulation of SAT and MAX-SAT as 0-1 integer linear programming problems. This formulation 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, which is far easier to solve. For some types of problems, such as Horn formulas and their generalizations, close connections exist between the logic problem and its linear programming relaxation (see e.g. [13]). We refer the reader to the book of Chandru and Hooker [15] for an excellent coverage of results at the interface of logical inference and optimization.

This survey presents some of the approaches developed in recent years for obtaining SDP relaxations for SAT and MAX-SAT problems; the main theoretical properties of these relaxations; and their practical impact so far in the area of SAT. We discuss two types of contributions of SDP researchers to satisfiability, namely

- polynomial-time approximation algorithms for MAX-SAT problems; and
- polynomial-time computational proofs of unsatisfiability for SAT.

Our focus in this paper is on the construction, analysis, and computational application of SDP relaxations for SAT and MAX-SAT, and thus we do not discuss algorithms for solving SDP problems in any detail. However, all the work presented is motivated by the fact that SDP problems can be solved efficiently using one of the algorithms that have been implemented and benchmarked by researchers in the area. We refer the reader to Monteiro [51] for a survey of the state-of-the-art in SDP algorithms.

The remainder of this paper is organized as follows. After concluding this Introduction with some preliminaries and notation, we provide in Section 2 a short introduction to SDP: the definition of an SDP problem and examples of SDP relaxations for the maximum-cut (max-cut) and MAX-SAT problems; some basic properties of positive semidefinite matrices; a few results on the geometry of SDP for max-cut; the basic concepts of duality in SDP; and a few comments on the computational complexity of SDP. Most of these results are referred to in later sections, and the remainder are included for completeness of the presentation. Section 2 may be skipped without loss of continuity by a reader familiar with SDP and its application to combinatorial optimization problems.

Section 3 focuses on SDP-based approximation algorithms for the MAX-SAT problem. We begin by presenting the ground-breaking random hyperplane rounding algorithm of Goemans and Williamson for MAX-2-SAT [24]. We then provide an overview of subsequent improvements on their approach, including the vector rotation technique of Feige and Goemans [22], and the biased hyperplane algorithm of Matuura and Matsui [50]. For MAX-3-SAT, we present the algorithm of Karloff and Zwick whose performance ratio is optimal (unless $P=NP$). We also mention some of the proposed extensions of these ideas to general MAX-SAT.

In Section 4, we introduce the Gap relaxation of de Klerk and others [19, 20]. This SDP relaxation is based on the concept of elliptic approximations for SAT. We show how

this relaxation displays a deep connection between SAT and eigenvalue optimization. It also characterizes unsatisfiability for a class of covering problems, including mutilated chessboard and pigeonhole instances, in the sense that the Gap relaxation for this class is always infeasible. Since it is possible to compute a certificate of infeasibility for this type of SDP problem (see Section 2.3), this approach provides a proof of unsatisfiability in polynomial-time for these problems in a fully automated manner.

We then introduce in Section 5 the concept of higher semidefinite liftings for polynomial optimization problems. This concept was first applied to the max-cut problem in [9], and more generally to 0-1 optimization problems in [42]. The higher liftings approach generalizes the Gap relaxation, but the resulting SDP problems grow very rapidly in dimension. Hence, the concept of partial higher liftings for SAT was proposed and analyzed in [3, 5]. The objective here is the construction of SDP relaxations which are linearly-sized with respect to the size of the SAT instance, and are thus more amenable to practical computation than the entire higher liftings. The construction of such partial liftings for SAT is particularly interesting because the structure of the SAT instance directly specifies the structure of the SDP relaxation. The resulting SDP relaxations as well as some of their properties are presented.

Finally, in Section 6, we compare the feasible sets of four of the aforementioned SDP relaxations for the special case of 3-SAT, and in Section 7, we review the most recent results in the application of SDP to SAT and MAX-SAT.

1.1 Preliminaries and Notation

We consider the satisfiability (SAT) problem for instances in conjunctive normal form (CNF). Such instances are specified by a set of proposition letters p_1, \dots, p_n , and a propositional formula $\Phi = \bigwedge_{j=1}^m C_j$, with each clause C_j having the form $C_j = \bigvee_{k \in I_j} p_k \vee \bigvee_{k \in \bar{I}_j} \bar{p}_k$ where

$I_j, \bar{I}_j \subseteq \{1, \dots, n\}$, $I_j \cap \bar{I}_j = \emptyset$, and \bar{p}_i denotes the negation of p_i . Given such an instance, we consider the following two questions:

SAT Determine whether Φ has a model, that is, whether there is a truth assignment to the variables p_1, \dots, p_n such that Φ evaluates to TRUE;

MAX-SAT Find a truth assignment to the variables p_1, \dots, p_n which maximizes the number of clauses in Φ that are satisfied. This is the unweighted version of MAX-SAT. If each clause C_j has a weight w_j associated with it, then the weighted MAX-SAT problem seeks a truth assignment so that the total weight of the satisfied clauses is maximized.

It is clear that MAX-SAT subsumes SAT, and indeed it is likely a much harder problem. For $k \geq 2$, k -SAT and MAX- k -SAT refer to the instances of SAT and MAX-SAT respectively for which all the clauses have length at most k .

The optimization problems we consider here are generally represented by a constraint set \mathcal{F} and an objective (or cost) function f that maps the elements of the constraint set into real numbers. The set \mathcal{F} represents all possible alternatives x , and for each such x , the value $f(x)$ of the objective function is a scalar measure of the desirability of choosing alternative x . Thus, an optimal solution is an $x^* \in \mathcal{F}$ such that $f(x^*) \geq f(x)$ for all $x \in \mathcal{F}$.

MAX-SAT problems are clearly optimization problems: given any truth assignment, the objective function counts the number of satisfied clauses, and the constraints describe the 2^n possible truth assignments. On the other hand, from an optimization perspective, the SAT problem can be approached in two different ways:

1. we may convert it to a MAX-SAT instance with $w_j = 1$ for every clause, solve it, and determine that it is satisfiable (resp. unsatisfiable) if the optimal objective value is equal to (resp. strictly less than) the number of clauses; or
2. we may view it as a feasibility problem, that is, we look for a set of constraints \mathcal{F} that must be satisfied by every model, but not by every truth assignment, and thus we reduce the SAT problem to the problem of determining whether \mathcal{F} has a feasible solution (which will correspond to a model) or $\mathcal{F} = \emptyset$ (which means the SAT instance is unsatisfiable).

In this survey, we focus on the latter approach, but the former is briefly mentioned in Section 7.

2. A Brief Introduction to Semidefinite Programming

Semidefinite programming refers to the class of optimization problems where a linear function of a symmetric matrix variable X is optimized subject to linear constraints on the elements of X and the additional constraint that X must be positive semidefinite. This includes linear programming (LP) problems as a special case, namely when all the matrices involved are diagonal. A variety of polynomial-time interior-point algorithms for solving SDPs have been proposed in the literature, and several excellent solvers for SDP are now available. We refer the reader to the semidefinite programming webpage [33] as well as the books [18, 68] for a thorough coverage of the theory and algorithms in this area, as well as a discussion of several application areas where semidefinite programming researchers have made significant contributions. In particular, SDP has been successfully applied in the development of approximation algorithms for several classes of hard combinatorial optimization problems beyond the results that we present in this paper. The survey articles [8, 45] provide an excellent overview of the results in this area.

Like LP problems, SDP problems also come in pairs. One of the problems is referred to as the *primal* problem, and the second one is the *dual* problem. Either problem can be chosen as “primal”, since the two problems are dual to each other. In this paper, we choose the standard formulation of SDP as follows:

$$\begin{array}{ll}
 \text{(P)} \quad \max & C \bullet X \\
 \text{s.t.} & A_i \bullet X = b_i, i = 1, \dots, \ell \\
 & X \succeq 0
 \end{array}
 \qquad
 \begin{array}{ll}
 \text{(D)} \quad \min & b^T y \\
 \text{s.t.} & Z = \sum_{i=1}^{\ell} y_i A_i - C \\
 & Z \succeq 0
 \end{array}
 \quad (1)$$

where (P) denotes the primal problem, and (D) the dual problem; the variables X and Z are in \mathbb{S}^n , the space of $n \times n$ real symmetric matrices; $X \succeq 0$ denotes that the matrix X is positive semidefinite; the data matrices A_i and C may be assumed to be symmetric without

loss of generality; and $b \in \mathbb{R}^\ell$ and $y \in \mathbb{R}^\ell$ are column vectors. Furthermore, we use the scalar product between two matrices in \mathbb{S}^n defined as

$$R \bullet S := \text{Tr } R^T S = \sum_{i=1}^n \sum_{j=1}^n R_{i,j} S_{i,j}$$

where $\text{Tr } M$ denotes the trace of the square matrix M , which is the sum of the diagonal elements. In summary, SDP is the problem of optimizing a linear function subject to linear equality constraints and the requirement that the matrix variable be positive semidefinite. The set of positive semidefinite matrices has a surprisingly rich structure. In the next subsection, we present some of the properties of such matrices that will be relevant for the application of SDP to SAT and MAX-SAT.

Before proceeding, we provide two examples that illustrate how SDP can be applied to combinatorial optimization problems. The first example is a derivation of the basic SDP relaxation for the max-cut problem which, following the ground-breaking work of Goemans and Williamson [24], has become one of the flagship problems for studying applications of semidefinite programming to combinatorial optimization. The max-cut SDP relaxation is of interest in the context of SAT because it is also a relaxation of the so-called cut polytope, an important and well-known structure in the area of integer programming which is closely related to the SAT relaxations in this survey. The reader is referred to [21] for a wealth of results about the cut polytope. The second example below shows how the max-cut relaxation can be extended to a basic SDP relaxation for MAX-SAT.

Example 1 (A basic SDP relaxation for max-cut). *The max-cut problem is a combinatorial optimization problem on undirected graphs with weights on the edges. Given such a graph $G = (V, E)$ with node set V and edge set E , the problem consists in finding a partition of V into two parts so as to maximize the sum of the weights on the edges that are cut by the partition. (An edge is said to be cut if it has exactly one end on each side of the partition.) We assume without loss of generality that G is a complete graph since non-edges can be added in with zero weight to complete the graph without changing the problem.*

Let the given graph G have node set $\{1, \dots, n\}$ and let it be described by its weighted adjacency matrix $W = (w_{ij})$. Let the vector $v \in \{-1, 1\}^n$ represent any cut in the graph via the interpretation that the sets $\{i : v_i = +1\}$ and $\{i : v_i = -1\}$ specify a partition of the node set of the graph. Then max-cut may be formulated as:

$$\begin{aligned} \max \quad & \sum_{i < j} w_{ij} \left(\frac{1 - v_i v_j}{2} \right) \\ \text{s.t.} \quad & v_i^2 = 1, \quad i = 1, \dots, n, \end{aligned} \tag{2}$$

so that the term multiplying w_{ij} in the sum equals one if the edge (i, j) is cut, and zero otherwise. Since $w_{ij} = w_{ji}$, $\sum_{i < j} w_{ij} \left(\frac{1 - v_i v_j}{2} \right) = \sum_{i, j} w_{ij} \left(\frac{1 - v_i v_j}{4} \right)$, and the objective function

can be expressed as:

$$\begin{aligned}
\frac{1}{4} \sum_{ij} (1 - v_i v_j) w_{ij} &= \frac{1}{4} \sum_i \left(\sum_j w_{ij} v_i^2 - \sum_j w_{ij} v_i v_j \right) \\
&= \frac{1}{4} \left(\sum_i \left(\sum_j w_{ij} \right) v_i v_i - \sum_{ij} v_i v_j \right) \\
&= \frac{1}{4} (v^T \text{Diag}(We)v - v^T Wv) \\
&= \frac{1}{4} v^T L v,
\end{aligned}$$

where $L := \text{Diag}(We) - W$ denotes the Laplacian matrix associated with the graph, e denotes the vector of all ones, and Diag denotes a diagonal matrix with its diagonal formed from the vector given as its argument. We can thus rewrite (2) as:

$$\begin{aligned}
\max \quad & v^T Q v \\
\text{s.t.} \quad & v_i^2 = 1, \quad i = 1, \dots, n,
\end{aligned}$$

where $Q := \frac{1}{4}L$.

To obtain an SDP relaxation, we now formulate max-cut in \mathbb{S}^n . Consider the change of variable $X = vv^T, v \in \{-1, 1\}^n$. Then $X \in \mathbb{S}^n$ and $v^T Q v = \text{Tr } v^T Q v = \text{Tr } Q X$ (using the fact that $\text{Tr } AB = \text{Tr } BA$), and it can be shown (see Theorem 8 below) that max-cut is equivalent to

$$\begin{aligned}
\max \quad & \text{Tr } Q X \\
\text{s.t.} \quad & X_{i,i} = 1, \quad i = 1, \dots, n \\
& \text{rank}(X) = 1 \\
& X \succeq 0, X \in \mathbb{S}^n
\end{aligned} \tag{3}$$

Removing the rank constraint (which is not convex) gives the basic SDP relaxation:

$$\begin{aligned}
\max \quad & \text{Tr } Q X \\
\text{s.t.} \quad & \text{diag}(X) = e \\
& X \succeq 0
\end{aligned} \tag{4}$$

where diag denotes a vector containing the diagonal entries of the matrix argument. The dual SDP problem is

$$\begin{aligned}
\min \quad & e^T y \\
\text{s.t.} \quad & Z = \text{Diag}(y) - Q \\
& Z \succeq 0
\end{aligned} \tag{5}$$

Example 2 (A basic SDP relaxation for MAX-SAT). Given an instance of MAX-SAT, we represent each boolean variable p_k by a ± 1 variable v_k , and each negation by \bar{p}_k by a ± 1 variable v_{n+k} , $k = 1, \dots, n$. Furthermore, introduce $v_0 \in \{-1, 1\}$ with the convention that p_k is TRUE if $v_k = -v_0$, and FALSE if $v_k = v_0$. Now the general MAX-SAT problem

can be formulated as:

$$\begin{aligned}
 \max \quad & \sum_{j=1}^m w_j z_j \\
 \text{s.t.} \quad & z_j \leq \sum_{i \in I_j} \frac{1-v_0 v_i}{2} + \sum_{i \in \bar{I}_j} \frac{1+v_0 v_i}{2}, \quad j = 1, \dots, m \\
 & 0 \leq z_j \leq 1, \quad j = 1, \dots, m \\
 & v_k v_{n+k} = -1, \quad k = 1, \dots, n \\
 & v_i \in \{-1, 1\}, \quad i = 0, 1, \dots, 2n,
 \end{aligned}$$

where $z_j = 1$ if and only if clause C_j is satisfied.

To relax this to an SDP, we replace the requirement that $v_i \in \{-1, 1\}$ by $v_i \in \mathbb{S}^n$, where \mathbb{S}^n denotes the unit sphere in \mathbb{R}^{n+1} . The matrix variable X for the SDP relaxation is now obtained by letting $X_{s,t} := v_s^T v_t$, $s, t = 0, 1, \dots, 2n$. This gives a $(2n+1) \times (2n+1)$ matrix and the SDP is:

$$\begin{aligned}
 \max \quad & \sum_{j=1}^m w_j z_j \\
 \text{s.t.} \quad & z_j \leq \sum_{i \in I_j} \frac{1-X_{0,i}}{2} + \sum_{i \in \bar{I}_j} \frac{1+X_{0,i}}{2}, \quad j = 1, \dots, m \\
 & 0 \leq z_j \leq 1, \quad j = 1, \dots, m \\
 & X_{k,n+k} = -1, \quad k = 1, \dots, n \\
 & \|v_i\|_2 = 1, \quad i = 1, \dots, 2n \\
 & X = V^T V
 \end{aligned} \tag{6}$$

where $V = (v_0, v_1, \dots, v_{2n})$. Since

$$X = V^T V, \|v_i\|_2 = 1 \quad \forall i \Leftrightarrow \text{diag}(X) = e, X \succeq 0$$

(see Theorem 5 below), this problem can be rewritten as an SDP problem closely related to the basic max-cut relaxation:

$$\begin{aligned}
 \max \quad & \sum_{j=1}^m w_j z_j \\
 \text{s.t.} \quad & z_j \leq \sum_{i \in I_j} \frac{1-X_{0,i}}{2} + \sum_{i \in \bar{I}_j} \frac{1+X_{0,i}}{2}, \quad j = 1, \dots, m \\
 & 0 \leq z_j \leq 1, \quad j = 1, \dots, m \\
 & X_{k,n+k} = -1, \quad k = 1, \dots, n \\
 & \text{diag}(X) = e \\
 & X \succeq 0
 \end{aligned} \tag{7}$$

In both examples, the SDP problem is a relaxation of the original problem. Thus, the optimal value of the SDP problem provides in each case a *global upper bound* on the true optimal value of the combinatorial problem.

Note also that these two examples illustrate two different, but equivalent, ways of deriving (and interpreting) SDP relaxations for combinatorial problems:

1. Embed the vector of binary variables into a rank-one matrix, formulate the combinatorial problem exactly using the entries of this matrix, and then remove the rank constraint to obtain an SDP; or

2. Replace the binary variables with real vectors of a suitably chosen length, and interpret their inner products as entries in a positive semidefinite matrix.

Both of these will be used in the sequel.

2.1 Positive Semidefinite Matrices

We begin with the definition of positive semidefiniteness.

Definition 1. A matrix $A \in \mathcal{S}^n$ is said to be positive semidefinite (psd) if

$$y^T A y = \sum_{i,j} A_{i,j} y_i y_j \geq 0 \text{ for all } y \in \mathfrak{R}^n.$$

When the condition holds with strict positivity for all $y \neq 0$, A is said to be positive definite (pd).

We use the notation $A \succeq 0$ for A positive semidefinite, and $A \succ 0$ for A positive definite.

We use \mathcal{S}_+^n (resp. \mathcal{S}_{++}^n) to denote the set of psd (resp. pd) matrices.

Example 3. The matrix $A = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix}$ is a rank-1 feasible solution for max-cut.

To prove this, observe that $A = vv^T$ with $v = (1 \ -1 \ 1)^T$. Hence, $y^T A y = (y^T v)^2 \geq 0$ for all $y \in \mathfrak{R}^n$.

The 3×3 identity matrix is also feasible, and it is easy to check that it is pd.

It follows immediately from Definition 1 that:

- Every non-negative linear combination of psd matrices is psd:

$$y^T \left(\sum_j \lambda_j X_j \right) y = \sum_j \lambda_j (y^T X_j y) \geq 0$$

whenever $\lambda_j \geq 0, X_j \succeq 0$ for all j .

- If $\lambda_j \geq 0, \sum_j \lambda_j > 0, X_j \succ 0$ for all j , then the linear combination $\sum_j \lambda_j X_j$ satisfies

$$y^T \left(\sum_j \lambda_j X_j \right) y = \sum_j \lambda_j (y^T X_j y) > 0$$

for all $y \neq 0$, and hence is pd.

- If $X, Y \in \mathcal{S}_+^n$, then $\lambda X + (1 - \lambda)Y \in \mathcal{S}_+^n$ for all $0 \leq \lambda \leq 1$. The same holds for \mathcal{S}_{++}^n , and thus both are convex sets. Note however that only \mathcal{S}_+^n is closed.

Consider the following definition:

Definition 2. A subset $\mathcal{K} \subset \mathbb{S}^n$ is called a cone if it is closed under positive scalar multiplication, i.e.

$$\lambda X \in \mathcal{K} \text{ whenever } X \in \mathcal{K} \text{ and } \lambda > 0.$$

Note that the origin may or may not be included. Clearly, both \mathbb{S}_+^n and \mathbb{S}_{++}^n are convex cones. The cone \mathbb{S}_+^n further possesses the following properties:

Definition 3. A cone \mathcal{K} is pointed if

$$\mathcal{K} \cap (-\mathcal{K}) = \{0\}.$$

It is proper if it has nonempty interior in \mathbb{S}^n and is closed, convex, and pointed.

For a proper cone \mathcal{K} , we define the dual cone \mathcal{K}^* as

$$\mathcal{K}^* := \{X \in \mathbb{S}^n : X \bullet Z \geq 0 \text{ for all } Z \in \mathcal{K}\}.$$

A proper cone \mathcal{K} is self-dual if $\mathcal{K}^* = \mathcal{K}$.

The self-duality of \mathbb{S}_+^n follows from the following theorem:

Theorem 1. [37, Corollary 7.5.4] $A \succeq 0 \iff A \bullet B \geq 0$ for all $B \succeq 0$.

Hence,

$$\begin{aligned} (\mathbb{S}_+^n)^* &= \{Y \in \mathbb{S}^n : X \bullet Y \geq 0 \ \forall X \in \mathbb{S}_+^n\} \\ &= \mathbb{S}_+^n \text{ by Theorem 1.} \end{aligned}$$

Theorem 1 above is only one of many properties of psd matrices. We present here a few more that will be of use in the sequel. First we have the following well-known theorem.

Theorem 2. [37, Theorem 4.1.5] (Spectral theorem) $A \in \mathbb{S}^n \iff$ there is a real matrix U s.t.

$$A = UDU^T$$

with $U^T U = U U^T = I$, and D a real diagonal matrix of the eigenvalues of A .

Corollary 1. All the eigenvalues of $A \in \mathbb{S}^n$ are real.

This leads to a first characterization of psd matrices.

Theorem 3. [37, Theorem 7.2.1] For $A \in \mathbb{S}^n$,

$$A \succeq 0 \iff \text{all the eigenvalues of } A \text{ are non-negative.}$$

Corollary 2. The trace and determinant of a psd matrix are non-negative.

Another useful characterization of psd matrices is formulated in terms of the principal minors of the given matrix.

Definition 4. Given $A \in \mathbb{S}^n$ and a subset $I \subseteq \{1 \dots n\}$, the principal submatrix of A corresponding to I is the submatrix with rows and columns indexed by I . Its determinant is called the principal minor.

Example 4. For $I = \{2, 3\}$ and

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 3 \\ 3 & 3 & 6 \end{pmatrix},$$

the principal submatrix is

$$A(I) = \begin{pmatrix} 4 & 3 \\ 3 & 6 \end{pmatrix}$$

and the principal minor is $\det A(I) = 15$.

The corresponding characterization of psd matrices is:

Theorem 4. [37, p.405] $A \succeq 0 \iff$ all principal minors are non-negative.

Hence, by considering $I = \{1, 3\}$, we can prove that the matrix in Example 4 is not psd.

A property which will be useful for the application to MAX-SAT is the existence of a square root of a psd matrix, which follows immediately from Theorems 2 and 3:

Theorem 5. $A \succeq 0 \iff \exists$ set of vectors $\{w_1, \dots, w_n\}$ such that $A_{i,j} = w_i^T w_j$, i.e., $A = W^T W$ where $W = (w_1, \dots, w_n)$.

Note that the matrix W is not unique. Indeed, for any orthogonal matrix $Q \in \mathbb{R}^n$, let $W_Q := QW$. Then $W_Q^T W_Q = W^T Q^T Q W = W^T W = A$, since $Q^T Q = I$ by the orthogonality of Q . A specific choice of W which is very useful from a computational point of view is the *Cholesky decomposition*:

$$A \succeq 0 \iff A = LL^T$$

where L is a lower triangular matrix. (If A is pd, then L is nonsingular with strictly positive entries on the diagonal.) The Cholesky decomposition can be computed efficiently (see for example [25]) and is useful in many practical algorithms for solving SDP problems.

Finally, there are two more properties which we will make use of:

Theorem 6. [37, Theorem 7.7.6] (Schur Complement theorem) If

$$M = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix},$$

and A is pd, then $M \succeq 0 \iff C - B^T A^{-1} B \succeq 0$.

Theorem 7. [37, Theorem 6.1.10] Let $A \in \mathbb{S}^n$ with strictly positive diagonal, and strictly diagonally dominant, i.e.

$$|A_{ii}| > \sum_{j \neq i, j=1}^n |A_{ij}| \quad \text{for } i = 1, \dots, n.$$

Then all the eigenvalues of A are strictly positive, i.e. A is pd.

2.2 The Geometry of SDP for Max-cut

We present here some results about the feasible set of the basic SDP relaxation of max-cut. We refer the reader to the excellent book of Deza and Laurent [21] which brings together the large body of results in this area, and focus here on results that will be useful in the sequel.

Definition 5. *The cut matrices in \mathcal{S}^n are the matrices of the form*

$$\begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T \text{ for some } x \in \{-1, 1\}^{n-1}.$$

The cut matrices are real, symmetric, psd, and rank-1. Furthermore, all their entries are ± 1 , and in particular $\text{diag}(X) = e$. The following theorem, based on results in [2], gives two characterizations of these matrices involving positive semidefiniteness.

Theorem 8. *Consider the following subsets of \mathcal{S}^n :*

$$\begin{aligned} \chi &:= \left\{ X \in \mathcal{S}^n : X = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T, x \in \{-1, 1\}^{n-1} \right\} \\ \chi_1 &:= \{X \in \mathcal{S}^n : \text{diag } X = e, \text{rank } X = 1, X \succeq 0\} \\ \chi_2 &:= \{X \in \mathcal{S}^n : X_{ij} \in \{-1, 1\}, X \succeq 0\}. \end{aligned}$$

Then $\chi = \chi_1 = \chi_2$.

Proof: The inclusions $\chi \subseteq \chi_1$ and $\chi \subseteq \chi_2$ are clear. It remains to prove the reverse inclusions.

Let $X \in \chi_1$. Then X is symmetric and rank-1, and thus it has the form $X = yy^T$. Since $\text{diag}(X) = e$, $y_i^2 = 1$ for every i , and we can choose y such that $y_1 = 1$ (replace y by $-y$ if necessary), we deduce that $X \in \chi$.

Let $X \in \chi_2$ and partition it as $\begin{pmatrix} 1 & x^T \\ x & \bar{X} \end{pmatrix}$, where the vector x is the first column of X minus the initial 1. Then by Theorem 6, $\bar{X} - xx^T \succeq 0$. Since $\text{diag}(xx^T) = e$, it follows that $\text{diag}(\bar{X} - xx^T) = 0$ and hence $\bar{X} - xx^T = 0$ (apply Theorem 4 to every 2×2 principal minor). Therefore, $X = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T$, so $X \in \chi$. \blacksquare

Note that χ consists of 2^{n-1} isolated points in \mathcal{S}^n . We now consider convex relaxations of χ , particularly some over which it is possible to optimize in polynomial-time.

The smallest convex set containing χ is the cut polytope, defined as the convex hull of the cut matrices:

$$C_n := \text{conv} \left\{ \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T : x \in \{-1, 1\}^{n-1} \right\}. \quad (8)$$

However, optimizing a linear function over C_n is equivalent to solving the max-cut problem, and is hence NP-hard. A fruitful approach is to approximate the cut polytope by a larger polytope containing it and over which we can optimize in polynomial time using LP. A

well-known relaxation is the metric polytope M_n defined as the set of all matrices satisfying the triangle inequalities:

$$\begin{aligned} M_n := \{X \in \mathcal{S}^n : \text{diag}(X) = e, X_{ij} + X_{ik} + X_{jk} \geq -1, \\ X_{ij} - X_{ik} - X_{jk} \geq -1, -X_{ij} + X_{ik} - X_{jk} \geq -1, \\ -X_{ij} - X_{ik} + X_{jk} \geq -1, \forall 1 \leq i < j < k \leq n\}. \end{aligned} \quad (9)$$

The triangle inequalities model the fact that for any assignment of ± 1 to the entries of x , the entries X_{ij}, X_{ik}, X_{jk} of the corresponding cut matrix must comprise an even number of negative ones.

Alternatively, it is possible to approximate the cut polytope with non-polyhedral convex sets. For instance, if we relax χ_1 by removing the rank constraint, we obtain another convex relaxation:

$$\mathcal{E}_n := \{X \in \mathcal{S}^n : \text{diag}(X) = e, X \succeq 0\}.$$

The set \mathcal{E}_n is precisely the feasible set of the basic SDP relaxation of max-cut (4). It is also known as the set of correlation matrices, and it has applications in several areas, including statistics, finance, and numerical analysis (see e.g. [17, 34] and the references therein).

For 3×3 principal submatrices of elements of \mathcal{E}_n , the following Lemma from [7] will be quite useful in the sequel.

Lemma 1. *Suppose*

$$\begin{pmatrix} 1 & a & b \\ a & 1 & c \\ b & c & 1 \end{pmatrix} \succeq 0.$$

Then

1. *If $a^2 = 1$ then $c = ab$;*
2. *If $b^2 = 1$ then $a = bc$;*
3. *If $c^2 = 1$ then $a = cb$.*

Proof: Using Theorems 4 and 6, we have:

$$\begin{pmatrix} 1 & a & b \\ a & 1 & c \\ b & c & 1 \end{pmatrix} \succeq 0 \Rightarrow \begin{pmatrix} 0 & c - ab \\ c - ab & 1 - b^2 \end{pmatrix} \succeq 0 \Rightarrow c - ab = 0$$

which proves the first claim.

The other two claims follow by similar arguments. ■

2.3 Duality in SDP

For (P) and (D) in the primal-dual pair (1) defined above, we have (as in LP):

Theorem 9. (Weak Duality) *If \tilde{X} is feasible for (P) and \tilde{y}, \tilde{Z} for (D), then $C \bullet \tilde{X} \leq b^T \tilde{y}$.*

Proof:

$$C \bullet \tilde{X} - \tilde{b}^T \tilde{y} = \left(\sum_{i=1}^m \tilde{y}_i A_i - \tilde{Z} \right) \bullet \tilde{X} - \sum_i \tilde{y}_i A_i \bullet \tilde{X} = -\tilde{Z} \bullet \tilde{X} \leq 0,$$

by Theorem 1. ■

However, because of the nonlinear psd constraint, SDP duality has some cases that do not occur in LP. We illustrate this with the following two examples from [68, pp. 71-72].

Positive Duality Gap In LP, if both (P) and (D) are feasible, then there is no duality gap. This may fail for SDP. For example,

$$\begin{array}{ll} \text{(P)} \quad \max & -ax_{11} \\ \text{s.t.} & x_{11} + 2x_{23} = 1 \\ & x_{22} = 0 \\ & X \succeq 0. \end{array} \quad \begin{array}{ll} \text{(D)} \quad \min & -y_2 \\ \text{s.t.} & \begin{pmatrix} y_2 - a & 0 & 0 \\ 0 & y_1 & y_2 \\ 0 & y_2 & 0 \end{pmatrix} \preceq 0 \end{array}$$

It is easy to see that (P) has optimal objective value $-a$, while (D) has 0.

Weak Infeasibility Even if there is no duality gap at optimality, the optimal value may not be attained for (P) and (D). Consider the primal-dual pair

$$\begin{array}{ll} \text{(P)} \quad \max & 2x_{12} \\ \text{s.t.} & \begin{pmatrix} 1 & x_{12} \\ x_{12} & 0 \end{pmatrix} \succeq 0 \end{array} \quad \begin{array}{ll} \text{(D)} \quad \min & y_1 \\ \text{s.t.} & \begin{pmatrix} y_1 & -1 \\ -1 & y_2 \end{pmatrix} \succeq 0 \end{array}$$

and observe that the optimal objective value 0 is attained for (P), but not for (D).

To avoid these difficulties, we require that the SDP pair satisfy a constraint qualification. This is a standard approach in nonlinear optimization. The purpose of a constraint qualification is to ensure the existence of Lagrange multipliers at optimality. These multipliers are an optimal solution to the dual problem, and thus the constraint qualification ensures that strong duality holds: it is possible to achieve primal and dual feasibility with no duality gap. For applications to combinatorial optimization problems such as SAT and MAX-SAT, *Slater's constraint qualification* is usually easy to verify: simply exhibit a feasible matrix which is pd (and not psd) for each of the SDP primal and dual problems.

Example 5 (Slater's Constraint Qualification for max-cut). *To illustrate Slater's constraint qualification, we show that it holds for the basic SDP relaxation of max-cut. Clearly, the $n \times n$ identity matrix is pd and feasible for the primal SDP (4). For the dual, choose y with entries sufficiently large such that $y_i > \sum_{j=1}^n Q_{ij}$ for each $i = 1, \dots, n$. Then the matrix $Z = \text{Diag}(y) - Q$ will be pd (by Theorem 7) and feasible for the dual SDP (5).*

Finally, we will use SDP to prove that certain instances of SAT are unsatisfiable. To prove that an SDP is infeasible, it suffices to compute a *certificate of infeasibility*, that is, a pair (y, Z) such that

$$b^T y = -1 \quad \text{and} \quad Z - \sum_{i=1}^{\ell} y_i A_i = 0.$$

The existence of such a pair implies that the dual SDP problem is unbounded below, and by weak duality (Theorem 9) the primal problem must be infeasible (since any primal feasible solution implies a lower bound on the optimal solution of the dual).

2.4 Computational Complexity and SDP Relaxations

When considering optimization approaches, we must assess the computational complexity of these problems. It is well known that SAT was the first problem shown to be NP-complete [16], although several important special cases can be solved in polynomial time, and that the MAX-SAT and MAX- k -SAT problems are known to be NP-hard. The hardness of these problems motivates the study of optimization problems which are not exact formulations of the SAT and MAX-SAT problems, but rather relaxations that can be solved in polynomial-time. These relaxations then lead to either approximation algorithms with polynomial-time complexity and provable approximation guarantees, or branch-and-bound (enumerative) approaches which solve the problem exactly, but have no guaranteed polynomial-time complexity. For completeness, we summarize here (based on [18, Section 1.9] and [45, Section 2.3]) a few facts about the complexity of SDP.

The fact that SDP problems can be solved in polynomial-time to within a given accuracy follows from the complexity analysis of the ellipsoid algorithm (see [28]). More specifically for our purposes, consider the SDP problem (P) defined in (1) with integer data, a given rational $\epsilon > 0$, and a given integer $R > 0$ such that either (P) is infeasible or $\|X\| \leq R$ for some feasible X . Then it is possible to find in polynomial-time either a matrix X^* at distance at most ϵ from the feasible set of (P) such that $C \bullet X^* - p^* \leq \epsilon$, or a certificate that the feasible set of (P) does not contain a ball of radius ϵ . The complexity of the algorithm is polynomial in n , ℓ , $\log(R)$, $\log(\frac{1}{\epsilon})$, and the bit length of the input data.

It is worth pointing out that, in contrast to LP, some peculiar situations may occur in SDP. First, there are SDP problems with no rational optimal solution. For instance, the pair of constraints (taken from [45])

$$\begin{pmatrix} 1 & x \\ x & 2 \end{pmatrix} \succeq 0 \quad \text{and} \quad \begin{pmatrix} 2x & 2 \\ 2 & x \end{pmatrix} \succeq 0$$

have $x = \sqrt{2}$ as the unique feasible solution (apply Corollary 2).

Another situation that may occur in SDP is that all feasible solutions are doubly exponential. Consider the set of constraints (taken from [54])

$$\begin{aligned} x_1 &\geq 2 \\ \begin{pmatrix} 1 & x_{i-1} \\ x_{i-1} & x_i \end{pmatrix} &\succeq 0, i = 2, \dots, n. \end{aligned}$$

Then any feasible solution must satisfy $x_i \geq 2^{2^i - 1}$, $i = 1, \dots, n$, which means that every rational feasible solution has exponential bitlength.

3. Approximation Algorithms for MAX-SAT

A ρ -approximation algorithm for MAX-SAT is a polynomial-time algorithm that computes a truth assignment such that at least a proportion ρ of the clauses in the MAX-SAT instance

are satisfied. The number ρ is the approximation ratio or guarantee. Håstad [31] proved that for any $\epsilon > 0$, there is no $(\frac{21}{22} + \epsilon)$ -approximation algorithm for MAX-2-SAT, and no $(\frac{7}{8} + \epsilon)$ -approximation algorithm for MAX-SAT (unless P=NP). This section, based on the excellent presentation by Laurent and Rendl in [45, Section 6.6], presents an overview of the approximation algorithms in the literature, and a detailed description of the groundbreaking algorithm of Goemans and Williamson.

The first approximation algorithm for MAX-SAT is a $\frac{1}{2}$ -approximation algorithm due to Johnson [39]. Given n values $\pi_i \in [0, 1]$, $i = 1, \dots, n$, the algorithm sets each variable p_i to TRUE independently and randomly with probability π_i . Therefore, the probability that clause $C_j = \bigvee_{k \in I_j} p_k \vee \bigvee_{k \in \bar{I}_j} \bar{p}_k$ is satisfied equals

$$1 - \prod_{k \in I_j} (1 - \pi_k) \prod_{k \in \bar{I}_j} \pi_k.$$

If we choose $\pi_i = 1/2$ for all $i = 1, \dots, n$, this probability is $1 - 2^{-l(C_j)}$, and thus the total expected weight of the satisfied clauses is

$$\sum_j w_j \left(1 - 2^{-l(C_j)}\right) \geq \frac{1}{2} \sum_j w_j.$$

This gives a randomized $\frac{1}{2}$ -approximation algorithm for MAX-SAT.

Improved approximation algorithms have since been obtained. The first $\frac{3}{4}$ -approximation algorithm for MAX-SAT was proposed by Yannakakis [69] and makes use of solutions to maximum flow problems, and subsequently Goemans and Williamson [23] presented another $\frac{3}{4}$ -approximation algorithm using LP. However, the best-known approximation guarantees for MAX-SAT problems make use of SDP relaxations and appropriate randomized rounding schemes.

3.1 SDP-based Approximation Algorithms for MAX-2-SAT

The breakthrough was achieved by Goemans and Williamson [24] who proposed an SDP-based approximation algorithm for the MAX-2-SAT problem with a 0.87856 guarantee. Their algorithm uses an extension of the SDP relaxation (6) in Example 2 above, obtained by adding for each 2-clause the appropriate inequality as follows:

$$\begin{aligned} p_i \vee p_j &: z_j \leq 1 - \left(\frac{1+v_0 v_i}{2}\right) \left(\frac{1+v_0 v_j}{2}\right) = \frac{3-v_0 v_i - v_0 v_j - v_i v_j}{4} \\ \bar{p}_i \vee p_j &: z_j \leq 1 - \left(\frac{1-v_0 v_i}{2}\right) \left(\frac{1+v_0 v_j}{2}\right) = \frac{3+v_0 v_i - v_0 v_j + v_i v_j}{4} \\ p_i \vee \bar{p}_j &: z_j \leq 1 - \left(\frac{1+v_0 v_i}{2}\right) \left(\frac{1-v_0 v_j}{2}\right) = \frac{3-v_0 v_i + v_0 v_j + v_i v_j}{4} \\ \bar{p}_i \vee \bar{p}_j &: z_j \leq 1 - \left(\frac{1-v_0 v_i}{2}\right) \left(\frac{1-v_0 v_j}{2}\right) = \frac{3+v_0 v_i + v_0 v_j - v_i v_j}{4} \end{aligned}$$

For clarity of presentation, we assume for the remainder of this section that $\bar{I}_j = \emptyset$ for all clauses. It is straightforward to modify the relaxations and the analysis to account for negated variables.

The Goemans-Williamson SDP relaxation for MAX-2-SAT is thus

$$\begin{aligned}
 \max \quad & \sum_{j=1}^m w_j z_j \\
 \text{s.t.} \quad & z_j \leq \sum_{i \in I_j} \frac{1-X_{0,i}}{2}, \quad j = 1, \dots, m \\
 & z_j \leq \frac{3-X_{0,i_1}-X_{0,i_2}-X_{i_1,i_2}}{4} \\
 & \quad \text{for } j \text{ such that } \{i_1, i_2\} = I_j \cup \bar{I}_j, j = 1, \dots, m \\
 & 0 \leq z_j \leq 1, \quad j = 1, \dots, m \\
 & X_{k,n+k} = -1, \quad k = 1, \dots, n \\
 & \text{diag } X = e \\
 & X \succeq 0
 \end{aligned} \tag{10}$$

Furthermore, it follows from Lemma 1 that $X_{k,n+k} = -1 \Rightarrow X_{s,n+k} = -X_{s,k}$ for $s = 0, 1, \dots, 2n$, i.e. the columns k and $n+k$ of X are the negative of each other, which is consistent with the definition of the vectors v_i in Example 2.

The algorithm of Goemans and Williamson proceeds as follows:

Step 1 Solve the SDP relaxation (10), obtaining an optimal value θ_{GW}^* and a corresponding optimal solution X^* .

Step 2 Compute a Gram decomposition of X^* using, for instance, the Cholesky decomposition (Theorem 5).

Thus obtain a set of $2n+1$ vectors $v_0, v_1, v_2, \dots, v_n, v_{n+1}, \dots, v_{2n}$ with the property that $X_{ij}^* = v_i^T v_j$ and $v_{n+k} = -v_k$ for $k = 1, \dots, n$.

Step 3 Randomly generate a vector $\eta \in \mathbb{S}^n$ and let H_η denote the hyperplane with normal η .

Step 4 For $i = 1, \dots, n$, let

$$p_i = \begin{cases} \text{TRUE,} & \text{if } H_\eta \text{ separates the vectors } v_0 \text{ and } v_i \\ \text{FALSE,} & \text{otherwise.} \end{cases}$$

We now sketch the analysis of the algorithm's performance. Let $\xi_{ij} := \arccos(v_i^T v_j)$ denote the angle between v_i and v_j , $p(i)$ denote the probability that the clause x_i is satisfied, and $p(i, j)$ denote the probability that the clause $x_i \vee x_j$ is satisfied. Our goal is to establish a relationship between θ_{GW}^* and the expected number of satisfied clauses.

First, the probability that H_η separates v_0 and v_i is equal to $\frac{\xi_{0i}}{\pi}$. Therefore, $p(i) = \frac{\xi_{0i}}{\pi}$, which implies that

$$\frac{p(i)}{z_i} = \frac{2}{\pi} \frac{\xi_{0i}}{1 - \cos \xi_{0i}} \geq \alpha_0$$

where

$$\alpha_0 := \min_{0 \leq \xi \leq \pi} \frac{2}{\pi} \frac{\xi}{1 - \cos \xi} \approx 0.87856. \tag{11}$$

Thus, $p(i) \geq \alpha_0 z_i$ for every clause of length 1.

For clauses of length 2, $p(i, j)$ is equal to the probability that at least one of the vectors v_i, v_j is separated from v_0 by H_η . Let $\bar{p}(i, j)$ denote the probability that v_0, v_i and v_j all lie on the same side of H_η , so that $p(i, j) = 1 - \bar{p}(i, j)$. A straightforward way to calculate this probability is to observe that it is equal to

$$\frac{1}{2} \left(\frac{\xi_{0i}}{\pi} + \frac{\xi_{0j}}{\pi} + \frac{\xi_{ij}}{\pi} \right)$$

and hence,

$$\frac{p(i, j)}{z(C_j)} \geq \frac{\frac{1}{2\pi} (\xi_{0i} + \xi_{0j} + \xi_{ij})}{\frac{3 - X_{0i} - X_{0j} - X_{ij}}{4}} = \frac{2}{\pi} \frac{\xi_{0i} + \xi_{0j} + \xi_{ij}}{(1 - \cos \xi_{0i}) + (1 - \cos \xi_{0j}) + (1 - \cos \xi_{ij})} \geq \alpha_0.$$

Hence, the expected total weight of the satisfied clauses is at least α_0 times the optimal value of the SDP relaxation.

Theorem 10. *For an instance of MAX-2-SAT, the Goemans-Williamson algorithm as described above provides a truth assignment for which*

$$(\text{expected number of satisfied clauses}) \geq \alpha_0 \theta_{GW}^*,$$

where θ_{GW}^* is the optimal value of (10), and α_0 is as defined in (11).

Since the optimal value of the MAX-2-SAT problem is at least the expected value of the randomized truth assignment, this proves that the algorithm is an α_0 -approximation algorithm for MAX-2-SAT. The randomized hyperplane rounding procedures can be formally derandomized using the techniques in [49]. This MAX-2-SAT algorithm led to an improved 0.7584-approximation algorithm for general MAX-SAT in [23].

For MAX-2-SAT, a further significant improvement was achieved by Feige and Goemans [22] who proposed a 0.931-approximation algorithm for MAX-2-SAT. They start with the SDP relaxation (10) augmented by the $\binom{n}{3}$ triangle inequalities (9). From the optimal solution X^* of this strengthened SDP relaxation, they obtain a set of $n + 1$ vectors v_0, v_1, \dots, v_n as in step 2 of the algorithm of Goemans and Williamson above. However, instead of applying the random hyperplane rounding technique to these vectors directly, Feige and Goemans use them to generate a set of rotated vectors to which they then apply the hyperplane rounding.

The general idea is as follows. We define a rotation function to be a continuous function $r : [0, \pi] \rightarrow [0, \pi]$ such that $r(0) = 0$ and $r(\pi - \xi) = \pi - r(\xi)$. Given such a rotation function, and the vectors v_0, v_1, \dots, v_n , then for $i = 1, \dots, n$, we define v_i^r , the rotated version of v_i , to be the vector obtained by rotating v_i in the plane spanned by v_0 and v_i until the angle between v_0 and v_i equals $r(\xi_{0i})$. As in the analysis above, it now follows that $p^r(i, j)$, the probability that at least one of the vectors v_i^r, v_j^r is separated from v_0 by H_η , is

$$\frac{r(\xi_{0i}) + r(\xi_{0j}) + r(\xi_{ij})}{2\pi}$$

and therefore

$$\frac{p^r(i, j)}{z(C_j)} \geq \frac{2}{\pi} \frac{r(\xi_{0i}) + r(\xi_{0j}) + r(\xi_{ij})}{(1 - \cos \xi_{0i}) + (1 - \cos \xi_{0j}) + (1 - \cos \xi_{ij})}.$$

The goal now is to choose the rotation function r so that the performance ratio of the algorithm is as high as possible in the worst case, and knowing that the quantities $\cos \xi_{0i}$, $\cos \xi_{0j}$, and $\cos \xi_{ij}$ satisfy the triangle inequalities (9).

Feige and Goemans [22] consider a family of rotation functions of the form

$$r_\lambda(\xi) = (1 - \lambda)\xi + \lambda\frac{\pi}{2}(1 - \cos \xi)$$

and claim a performance ratio of at least 0.93109 for the rotation function defined by $\lambda = 0.806765$. A precise performance ratio of 0.931091 for this algorithm was subsequently proved rigorously by Zwick [70].

Matuura and Matsui [50] obtained a higher performance ratio with an approximation algorithm which, like that of Feige and Goemans, uses the SDP relaxation (10) with the triangle inequalities (9) added. Matuura and Matsui fix $v_0 = (1, 0, \dots, 0)^T$, and the remaining vectors v_1, \dots, v_n are obtained from X^* as usual. Note that the restriction on v_0 can be easily handled via an appropriate orthogonal matrix (see Theorem 5 and the discussion thereafter). Instead of rotating the set of vectors, Matuura and Matsui change the way of choosing the random hyperplane. Instead of using a uniform distribution, they select the random hyperplane using a distribution function on the sphere which is skewed towards v_0 , and uniform in every direction orthogonal to v_0 . A judicious choice of the distribution function yields a 0.935 performance ratio for their algorithm.

Finally, Lewin, Livnat and Zwick [46] proposed a combination of the Feige-Goemans and Matuura-Matsui approaches, namely the rotation of the set of vectors and the skewed hyperplane rounding, and obtained a 0.940-approximation algorithm for MAX-2-SAT.

3.2 SDP-based Approximation Algorithms for MAX-3-SAT

The MAX-2-SAT approach of Goemans and Williamson was extended to MAX-3-SAT by Karloff and Zwick [40]. Karloff and Zwick use the following SDP relaxation:

$$\begin{aligned} \max \quad & \sum_{j=1}^m w_j z_j \\ \text{s.t.} \quad & z_j \leq \frac{1 - X_{0,i_1}}{2} \quad \text{for } j \text{ such that } I_j \cup \bar{I}_j = \{i_1\} \\ & z_j \leq \frac{3 - X_{0,i_1} - X_{0,i_2} - X_{i_1,i_2}}{4} \quad \text{for } j \text{ such that } I_j \cup \bar{I}_j = \{i_1, i_2\} \\ & z_j \leq 1 - \frac{X_{0,i_2} + X_{0,i_3} + X_{i_1,i_2} + X_{i_1,i_3}}{4}, \quad z_j \leq 1 - \frac{X_{0,i_1} + X_{0,i_3} + X_{i_1,i_2} + X_{i_2,i_3}}{4}, \\ & \quad z_j \leq 1 - \frac{X_{0,i_1} + X_{0,i_2} + X_{i_1,i_3} + X_{i_2,i_3}}{4}, \quad \text{for } j \text{ such that } I_j \cup \bar{I}_j = \{i_1, i_2, i_3\} \\ & 0 \leq z_j \leq 1 \quad j = 1, \dots, m \\ & X_{k,n+k} = -1, \quad k = 1, \dots, n \\ & \text{diag}(X) = e \\ & X \succeq 0 \end{aligned} \tag{12}$$

and extend the analysis of Goemans and Williamson to obtain an optimal (unless P=NP) $\frac{7}{8}$ -approximation algorithm for MAX-3-SAT [40, 70].

We use the same notation as above, and further let θ_{KZ}^* denote the optimal value of the SDP relaxation (12), and $p(i, j, k)$ denote the probability that the clause $x_i \vee x_j \vee x_k$ is satisfied. Our goal is to establish a relationship between θ_{KZ}^* and the expected number

of satisfied clauses. Since $\alpha_0 > \frac{7}{8}$, the analysis of Goemans and Williamson for clauses of lengths 1 and 2 gives the desired performance ratio.

For clauses of length 3, however, a more careful analysis is required. Using the random hyperplane rounding technique, we still have that $p(i, j, k)$ is equal to the probability that the random hyperplane H_η separates v_0 from at least one of the vectors v_i, v_j, v_k . Therefore, $p(i, j, k) = 1 - \bar{p}(i, j, k)$, where $\bar{p}(i, j, k)$ is the probability that v_0, v_i, v_j and v_k all lie on the same side of H_η . By symmetry, it follows that $\bar{p}(i, j, k)$ equals two times the probability that $\eta^T v_0, \eta^T v_i, \eta^T v_j$, and $\eta^T v_k$ are all non-negative. Denote this probability by $\bar{q}(i, j, k)$. The estimation of $\bar{q}(i, j, k)$ is the main step of the performance analysis.

Since we are only interested in the four inner products above, we can restrict ourselves to the 4-dimensional space spanned by v_0, v_i, v_j , and v_k , with the vector η replaced by its normalized projection onto this space, which is uniformly distributed on the unit sphere \mathbb{S}^3 . Hence, we may assume without loss of generality that we are working in \mathbb{R}^4 and that all five vectors of interest lie in \mathbb{S}^3 . If we define

$$T(0, i, j, k) := \{\nu \in \mathbb{S}^3 : \nu^T v_t \geq 0, \forall t = 0, i, j, k\},$$

then $\bar{q}(i, j, k) = \frac{\text{volume}(T(0, i, j, k))}{\text{volume}(\mathbb{S}^3)}$, where $\text{volume}(\cdot)$ denotes the 3-dimensional spherical volume. Since $\text{volume}(\mathbb{S}^3) = 2\pi^2$, it follows that

$$p(i, j, k) = 1 - \frac{\text{volume}(T(0, i, j, k))}{\pi^2}.$$

It remains to estimate the spherical volume of $T(0, i, j, k)$. When the vectors v_0, v_i, v_j , and v_k are linearly independent, $T(0, i, j, k)$ is a spherical tetrahedron. However, there is no known closed-form formula for this volume, and it is possible that none exists.

Karloff and Zwick [40] proved that if the instance of MAX-3-SAT is satisfiable, that is, if $\theta_{KZ}^* = 1$, then $p(i, j, k) \geq \frac{7}{8}$. Zwick [70] proved rigorously the performance ratio of $\frac{7}{8}$ for general MAX-3-SAT. It is worth noting that both proofs were computer assisted: the first result involved one computation carried out using Mathematica with 50 digits of precision, and the second result was obtained using Zwick's RealSearch system, which makes use of interval arithmetic rather than floating point arithmetic, thus providing a rigorous proof.

3.3 Further Extensions of the SDP-based Approach to MAX-SAT

Karloff and Zwick [40] also proposed a general construction of SDP relaxations for MAX- k -SAT. For MAX-4-SAT specifically, Halperin and Zwick [30] proposed an SDP relaxation, studied several rounding schemes, and obtained approximation algorithms that almost attain the theoretical upper bound of $\frac{7}{8}$. Halperin and Zwick also consider strengthened SDP relaxations for MAX- k -SAT. Most recently, Asano and Williamson [10] have combined ideas from several of the aforementioned approaches and obtained a 0.7846-approximation algorithm for general MAX-SAT.

4. First Lifting: The Gap Relaxation

We now turn to the application of SDP to SAT in terms of a feasibility problem. The initial work in this area is due to de Klerk, van Maaren, and Warners [19, 20] who introduced

the Gap relaxation for SAT. This SDP relaxation was motivated by the concept of *elliptic approximations* for SAT instances. Elliptic approximations were first proposed in [60] and were applied to obtain effective branching rules as well as to recognize certain polynomially solvable classes of SAT instances [59, 62, 63, 64, 65].

We let TRUE be denoted by 1 and FALSE by -1 , and for clause j and $k \in I_j \cup \bar{I}_j$, define

$$s_{j,k} := \begin{cases} 1, & \text{if } k \in I_j \\ -1, & \text{if } k \in \bar{I}_j \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

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

$$\begin{aligned} \text{find } & x \in \{-1, 1\}^n \\ \text{s.t. } & \sum_{k \in I_j \cup \bar{I}_j} s_{j,k} x_k \geq 2 - l(C_j), \quad j = 1, \dots, m \end{aligned} \quad (14)$$

where $l(C_j) = |I_j \cup \bar{I}_j|$ denotes the number of literals in clause C_j .

If every $x_k = \pm 1$, then the corresponding truth assignment satisfies C_j precisely when

$$l(C_j) \geq \sum_{k \in I_j \cup \bar{I}_j} s_{j,k} x_k \geq 2 - l(C_j) \Leftrightarrow \left(1 - \sum_{k \in I_j \cup \bar{I}_j} s_{j,k} x_k \right)^2 \leq (l(C_j) - 1)^2$$

This motivates the definition of the elliptic approximation of C_j , denoted \mathbb{E}_j :

$$\mathbb{E}_j := \left\{ x \in \mathbb{R}^n : \left(1 - \sum_{k \in I_j \cup \bar{I}_j} s_{j,k} x_k \right)^2 \leq (l(C_j) - 1)^2 \right\} \quad (15)$$

This is called elliptic because the set of points in \mathbb{R}^n contained in \mathbb{E}_j forms an ellipsoid. Using these approximations, we can reformulate SAT as the problem of finding a ± 1 vector x in the intersection of the m ellipsoids. However, it is difficult to work directly with intersections of ellipsoids, but we can use these ellipsoids to obtain an SDP relaxation of this problem.

At this point, there are two ways to use the concept of elliptic approximation for constructing an SDP relaxation for SAT, both of which lead to the Gap relaxation. The first derivation we present shows a deep connection between SAT and eigenvalue optimization, and justifies the name of the relaxation. The second derivation is more direct, and sets the stage for the subsequent development of tighter SDP relaxations.

4.1 First Derivation of the Gap Relaxation

This derivation is motivated by the transformation of the SAT problem into one involving the optimization of eigenvalues, which can be reformulated as an SDP problem. Our presentation is based on the derivation in [20].

First, we consider the aggregation of this information into a single ellipsoid by taking the sum of the m ellipsoids. This again yields an ellipsoid. Furthermore, rather than giving

each ellipsoid equal weight in the sum, we can associate a non-negative weight w_j with each clause C_j , and thus define the *weighted elliptic approximation* of the given SAT formula Φ :

$$\mathbb{E}(\Phi, w) := \left\{ x \in \mathfrak{R}^n : \sum_{j=1}^m w_j \left(1 - \sum_{k \in I_j \cup \bar{I}_j} s_{j,k} x_k \right)^2 \leq \sum_{j=1}^m w_j (l(C_j) - 1)^2 \right\} \quad (16)$$

It is straightforward to prove that

Theorem 11. [20, Theorem 2.1] *Let Φ be a CNF formula with associated parameters $s_{j,k}$ and weighted elliptic approximation $\mathbb{E}(\Phi, w)$. If $x \in \{-1, 1\}^n$ is a satisfying truth assignment of Φ , then $x \in \mathbb{E}(\Phi, w)$ for any choice of $w \geq 0$.*

The contrapositive of Theorem 11 gives a sufficient condition for proving unsatisfiability.

Corollary 3. *Let Φ be a CNF formula. If there exists $w \geq 0$ such that $x \notin \mathbb{E}(\Phi, w)$ for all $x \in \{-1, 1\}^n$, then Φ is unsatisfiable.*

Since $x \in \{-1, 1\}^n$ implies $x^T x = n$, another sufficient condition for unsatisfiability is:

Corollary 4. *Let Φ be a CNF formula. If there exists $w \geq 0$ such that $x \notin \mathbb{E}(\Phi, w)$ whenever $x^T x = n$, then Φ is unsatisfiable.*

Next, we rewrite condition (16) as

$$x^T S^T W S x - 2w^T S x \leq r^T w$$

where $W = \text{Diag}(w)$, $S = (s_{j,k})$ is an $m \times n$ matrix, and r is an m -vector with $r_j = l(C_j)(l(C_j) - 2)$. Furthermore, we introduce an extra boolean variable x_{n+1} to obtain a homogeneous quadratic inequality:

$$x^T S^T W S x - 2x_{n+1} w^T S x - r^T w \leq 0. \quad (17)$$

Note that Theorem 11 and Corollaries 3 and 4 still hold for this homogenized inequality. With $\tilde{x} := (x_1, x_2, \dots, x_n, x_{n+1})^T$ and the $(n+1) \times (n+1)$ matrix

$$Q(w) := \begin{pmatrix} S^T W S - \frac{r^T w}{n} I & -S^T w \\ -w^T S & 0 \end{pmatrix}$$

and since $x^T x = n$, we can rewrite (17) as $\tilde{x}^T Q(w) \tilde{x} \leq 0$. Finally, we add a correcting vector $u \in \mathfrak{R}^n$ to obtain

$$\tilde{Q}(w, u) := \begin{pmatrix} S^T W S - \frac{r^T w}{n} I - \text{Diag}(u) & -S^T w \\ -w^T S & e^T u \end{pmatrix}$$

and it is easy to check that $\tilde{x}^T \tilde{Q}(w, u) \tilde{x} = \tilde{x}^T Q(w) \tilde{x}$.

The upshot of this derivation is that the given formula Φ is unsatisfiable if we can find a pair (w, u) with $w \geq 0$ for which $\tilde{x}^T \tilde{Q}(w, u) \tilde{x} > 0$ for all $\tilde{x} \in \{-1, 1\}^{n+1}$. Thus, we consider the problem

$$\begin{aligned} \max \quad & (n+1)\lambda \\ \text{s.t.} \quad & \tilde{Q}(w, u) \succeq \lambda I \\ & w \geq 0 \end{aligned} \quad (18)$$

The optimal value of problem (18) is the *gap of formula* Φ . We have the following result:

Corollary 5. *If the formula Φ has a positive gap, then it is unsatisfiable.*

Proof: If the optimal value of the problem (18) is positive, then there exists $\lambda^* > 0$ and a pair (w, u) with $w \geq 0$ such that $\tilde{Q}(w, u)$ is pd, since $\tilde{Q}(w, u) \succeq \lambda^* I \succ 0$ holds. By Corollary 4, Φ must be unsatisfiable. \blacksquare

Note that problem (18) is an SDP. To write down its dual, we rewrite the matrix $\tilde{Q}(w, u)$ as follows:

$$\tilde{Q}(w, u) = \sum_{j=1}^m w_j \begin{pmatrix} s_j s_j^T - \frac{r_j}{n} I & -s_j \\ -s_j^T & 0 \end{pmatrix} + \sum_{k=1}^n u_k \begin{pmatrix} -e_k e_k^T & 0 \\ 0 & 1 \end{pmatrix}$$

where s_j denotes the j^{th} column of S^T , and e_j denotes the $n \times n$ unit vector with a 1 in the j^{th} position and zeros elsewhere. Rewriting problem (18) as

$$\begin{aligned} & -\min \quad -(n+1)\lambda \\ \text{s.t.} \quad & \sum_{j=1}^m w_j \begin{pmatrix} s_j s_j^T - \frac{r_j}{n} I & -s_j & 0 \\ -s_j^T & 0 & 0 \\ 0 & 0 & e_j e_j^T \end{pmatrix} + \sum_{k=1}^n u_k \begin{pmatrix} -e_k e_k^T & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ & + \lambda \begin{pmatrix} -I & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \succeq 0 \end{aligned}$$

we have a problem in the form (D) of Section 2. Therefore, the corresponding dual is the problem of the form

$$\begin{aligned} & \max \quad 0 \\ \text{s.t.} \quad & \mathbf{Tr}(s_j s_j^T Y^{(1)}) - \frac{r_j}{n} \mathbf{Tr}(Y^{(1)}) - 2s_j^T y + Y_{j,j}^{(2)} = 0, \quad j = 1, \dots, m \\ & -Y_{k,k}^{(1)} + \bar{y} = 0, \quad k = 1, \dots, n \\ & -\sum_{k=1}^n Y_{k,k}^{(1)} - \bar{y} = -(n+1) \\ & \begin{pmatrix} Y_{n \times n}^{(1)} & y & 0 \\ y^T & \bar{y} & 0 \\ 0 & 0 & Y_{m \times m}^{(2)} \end{pmatrix} \succeq 0 \end{aligned}$$

We make the following observations:

1. Since the objective function is constant, this is equivalent to a feasibility problem.
2. $Y_{m \times m}^{(2)} \succeq 0$ implies $Y_{j,j}^{(2)} \geq 0$ for all j , hence the first set of constraints is equivalent to $s_j s_j^T Y^{(1)} - \frac{r_j}{n} Y^{(1)} - 2s_j^T y \leq 0$.
3. The constraints $-\sum_{k=1}^n Y_{k,k}^{(1)} - \bar{y} = -(n+1)$ and $-Y_{k,k}^{(1)} + \bar{y} = 0, k = 1, \dots, n$ are precisely equivalent to the constraints $\text{diag}(Y^{(1)}) = e$ and $\bar{y} = 1$.

4. Hence, $\mathbf{Tr}(Y^{(1)}) = n$ for every feasible solution.
 Furthermore, $\mathbf{Tr}(s_j s_j^T Y^{(1)}) = \mathbf{Tr}(s_j^T Y^{(1)} s_j) = s_j^T Y^{(1)} s_j$.

Hence, the dual problem is equivalent to

$$\begin{aligned} \text{find } & Y \in \mathcal{S}^{n+1} \\ \text{s.t. } & s_j^T Y^{(1)} s_j - 2s_j^T y \leq r_j, \quad j = 1, \dots, m \\ & \text{diag}(Y) = e \\ & Y = \begin{pmatrix} 1 & y^T \\ y & Y^{(1)} \end{pmatrix} \\ & Y \succeq 0. \end{aligned}$$

4.2 Second Derivation of the Gap Relaxation

Recall the inequality defining the ellipsoid approximation \mathbb{E}_j of clause C_j :

$$\left(1 - \sum_{k \in I_j \cup \bar{I}_j} s_{j,k} x_k\right)^2 \leq (l(C_j) - 1)^2\}$$

Expanding it, we obtain:

$$\sum_{k \in I_j \cup \bar{I}_j} \sum_{k' \in I_j \cup \bar{I}_j} s_{j,k} s_{j,k'} x_k x_{k'} - 2 \sum_{k \in I_j \cup \bar{I}_j} s_{j,k} x_k \leq l(C_j)^2 - 2l(C_j)$$

and since every term in the double sum with $k = k'$ equals 1 (for ± 1 solutions), we have

$$\sum_{k \in I_j \cup \bar{I}_j} \sum_{k' \in I_j \cup \bar{I}_j, k' \neq k} s_{j,k} s_{j,k'} x_k x_{k'} - 2 \sum_{k \in I_j \cup \bar{I}_j} s_{j,k} x_k \leq l(C_j)^2 - 3l(C_j)$$

Letting $Y = xx^T$ and applying Theorem 8, we obtain the formulation

$$\begin{aligned} \text{find } & Y \in \mathcal{S}^{n+1} \\ \text{s.t. } & \sum_{k \in I_j \cup \bar{I}_j} \sum_{k' \in I_j \cup \bar{I}_j, k' \neq k} s_{j,k} s_{j,k'} Y_{k,k'} - 2 \sum_{k \in I_j \cup \bar{I}_j} s_{j,k} Y_{0,k} \leq l(C_j)^2 - 3l(C_j) \\ & \text{diag}(Y) = e \\ & \text{rank } Y = 1 \\ & Y \succeq 0 \end{aligned}$$

where we refer to the first row (and column) of the matrix variable $Y = \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix}$ as the 0th row (and column), so that Y has rows and columns indexed by $\{0, 1, \dots, n\}$. Omitting the rank constraint, we obtain an SDP relaxation. It is straightforward to check that this is the same SDP as that obtained using the first derivation.

4.3 Properties of the Gap Relaxation

The Gap relaxation characterizes unsatisfiability for 2-SAT problems (see Theorem 12 below). More interestingly, it also characterizes satisfiability for a class of covering problems, such as mutilated chessboard and pigeonhole instances. Rounding schemes and approximation guarantees for the Gap relaxation, as well as its behaviour on so-called $(2 + p)$ -SAT problems, are studied in [19]. We present here some details about the first two properties of the Gap relaxation, and refer the reader to the cited papers for more details.

First we note that for each clause of length 1 or 2, the elliptic approximation inequality can be set to equality. Therefore we henceforth consider the Gap relaxation in the following form:

$$\begin{aligned}
& \text{find } Y \in \mathcal{S}^{n+1} \\
& \text{s.t.} \\
& \quad \sum_{k \in I_j \cup \bar{I}_j} \sum_{k' \in I_j \cup \bar{I}_j} s_{j,k} s_{j,k'} Y_{k,k'} - \sum_{k \in I_j \cup \bar{I}_j} s_{j,k} Y_{0,k} = -1, \\
& \quad \quad \quad \text{for } j \text{ such that } l(C_j) = 2 \\
& \quad \sum_{k \in I_j \cup \bar{I}_j} \sum_{k' \in I_j \cup \bar{I}_j, k' \neq k} s_{j,k} s_{j,k'} Y_{k,k'} - 2 \sum_{k \in I_j \cup \bar{I}_j} s_{j,k} Y_{0,k} \leq l(C_j)^2 - 3l(C_j), \\
& \quad \quad \quad \text{for } j \text{ such that } l(C_j) \geq 3 \\
& \quad \text{diag}(Y) = e \\
& \quad Y \succeq 0
\end{aligned} \tag{19}$$

A first result about the Gap relaxation is:

Theorem 12. [20, Theorem 5.1] *The Gap relaxation for a 2-SAT formula Φ has a positive gap if and only if Φ is unsatisfiable.*

Therefore, the Gap relaxation is feasible if and only if Φ is satisfiable. Furthermore, given a feasible solution Y for the relaxation (19), it is straightforward to extract a truth assignment satisfying the 2-SAT formula by the following algorithm:

1. For every k such that $Y_{0,k} \neq 0$, set $x_k = \text{sign}(Y_{0,k})$;
2. For every remaining constraint j that remains unsatisfied, we have the constraint's two variables k and k' such that $Y_{0,k} = 0$ and $Y_{0,k'} = 0$. Hence, $Y_{k,k'} = -s_{j,k} s_{j,k'} = \pm 1$. Considering all these ± 1 elements, the matrix can be completed to a rank-1 matrix that is feasible for (19).

See [20, 22] for more details. We will make use of this result in the sequel.

The next result about the Gap relaxation is concerned with a class of covering instances. Let $\{V_1, \dots, V_q\}$ and $\{V_{q+1}, \dots, V_{q+t}\}$ be two partitions of the set of variables $\{p_1, \dots, p_n\}$ such that $t < q$. We consider CNF formulas Φ of the form:

$$\bigwedge_{j=1}^q \left(\bigvee_{k \in V_j} p_k \right) \wedge \bigwedge_{j=q+1}^{q+t} \left(\bigwedge_{k_1, k_2 \in V_j, k_1 \neq k_2} (\bar{p}_{k_1} \vee \bar{p}_{k_2}) \right). \tag{20}$$

We show that the SDP approach, using the Gap relaxation, can prove such formulas to be unsatisfiable in polynomial-time in a fully automated manner, i.e. without using any

problem-specific information. For these instances, the Gap relaxation can be written as

$$\begin{aligned}
 &\text{find } Y \in \mathcal{S}^{n+1} \\
 &\text{s.t.} \\
 &\quad \sum_{k \in V_j} \sum_{k' \in V_j, k' \neq k} Y_{k,k'} - 2 \sum_{k \in V_j} Y_{0,k} \leq |V_j|(|V_j| - 3), \quad \text{for } j = 1, \dots, q \\
 &\quad Y_{k,k'} + Y_{0,k} + Y_{0,k'} = -1, \quad \text{for } k, k' \in V_j, k > k', j = q + 1, \dots, q + t \\
 &\quad \text{diag}(Y) = e \\
 &\quad Y \succeq 0
 \end{aligned} \tag{21}$$

We prove the following theorem.

Theorem 13. [20, Theorem 6.1] *The SDP relaxation (21) is infeasible if and only if $t < q$ (i.e. the corresponding instance Φ is unsatisfiable).*

Proof: If $t \geq q$, then the formula is satisfiable, and clearly the SDP relaxation is feasible. Suppose now that $t < q$. Consider the following partition of Y feasible for (21):

$$Y = \begin{pmatrix} 1 & y^T \\ y & \tilde{Y} \end{pmatrix}$$

where \tilde{Y} is $n \times n$ and $\text{diag}(\tilde{Y}) = e$. By Theorem 6 and the positive semidefiniteness of Y ,

$$\begin{pmatrix} 1 & y^T \\ y & \tilde{Y} \end{pmatrix} \succeq 0 \Rightarrow \tilde{Y} - yy^T \succeq 0 \Rightarrow s_j^T \tilde{Y} s_j - s_j^T yy^T s_j \geq 0 \Rightarrow s_j^T \tilde{Y} s_j \geq (s_j^T y)^2$$

for every vector s_j . Hence, for each of the first q constraints,

$$\left(\sum_{k \in V_j} Y_{0,k} \right)^2 \leq \sum_{k \in V_j} \sum_{k' \in V_j} Y_{k,k'} = |V_j| + \sum_{k \in V_j} \sum_{k' \in V_j, k' \neq k} Y_{k,k'}$$

and therefore

$$\begin{aligned}
 \left(\sum_{k \in V_j} Y_{0,k} \right)^2 - 2 \sum_{k \in V_j} Y_{0,k} &\leq |V_j| + \sum_{k \in V_j} \sum_{k' \in V_j, k' \neq k} Y_{k,k'} - 2 \sum_{k \in V_j} Y_{0,k} \\
 &\leq |V_j| + |V_j|(|V_j| - 3) = |V_j|(|V_j| - 2)
 \end{aligned}$$

which implies that

$$2 - |V_j| \leq \sum_{k \in V_j} Y_{0,k} \leq |V_j|.$$

Summing these inequalities over $j = 1, \dots, q$, we obtain

$$2q - n \leq \sum_{k=1}^n Y_{0,k} \leq n \tag{22}$$

Now consider the remaining t constraints. For $j = q + 1, \dots, t$, adding up all the corresponding constraints yields

$$\sum_{k \in V_j} \sum_{k' \in V_j, k' > k} Y_{k,k'} + (|V_j| - 1) \sum_{k \in V_j} Y_{0,k} = -\binom{|V_j|}{2}$$

which is equivalent to

$$\sum_{k \in V_j} \sum_{k' \in V_j, k' \neq k} Y_{k,k'} + 2(|V_j| - 1) \sum_{k \in V_j} Y_{0,k} = -|V_j|(|V_j| - 1)$$

and therefore

$$\left(\sum_{k \in V_j} Y_{0,k} \right)^2 - |V_j| + 2(|V_j| - 1) \sum_{k \in V_j} Y_{0,k} \leq -|V_j|(|V_j| - 1)$$

which is equivalent to

$$\left(\sum_{k \in V_j} Y_{0,k} \right)^2 + 2(|V_j| - 1) \sum_{k \in V_j} Y_{0,k} \leq -|V_j|(|V_j| - 1) + |V_j| = -|V_j|(|V_j| - 2)$$

which implies that

$$-|V_j| \leq \sum_{k \in V_j} Y_{0,k} \leq 2 - |V_j|, \text{ for } j = q + 1, \dots, t.$$

Summing these inequalities over $j = q + 1, \dots, t$, we obtain

$$-n \leq \sum_{k=1}^n Y_{0,k} \leq 2t - n \tag{23}$$

Equations (22) and (23) together imply

$$2q \leq \sum_{k=1}^n Y_{0,k} \leq 2t$$

which is a contradiction since $t < q$. ■

Theorem 13 shows that for the class of instances (20), the SDP approach not only proves that infeasibility can be shown in polynomial-time, but also provides a certificate of infeasibility in polynomial-time without making explicit use of any additional knowledge about the instance.

On the other hand, for any instance of SAT in which all the clauses have length three or more, the identity matrix is always feasible for the Gap relaxation. Therefore, this relaxation cannot prove unsatisfiability for any such instances. The work presented in the following section was partly motivated by the search for an SDP relaxation which can be used to prove that a given SAT formula is unsatisfiable independently of the lengths of the clauses in the instance.

5. Strengthened SDP Relaxations via Higher Liftings

The SDP relaxations presented in this Section are strengthenings of the Gap relaxation, and they inherit many of its properties. They are constructed using ideas from a “higher liftings” paradigm for constructing SDP relaxations of discrete optimization problems.

The concept of lifting has been proposed by several researchers and has led to different general purpose hierarchical frameworks for solving 0-1 optimization problems. The idea of applying SDP relaxations to 0-1 optimization dates back at least to Lovász’s introduction of the so-called theta function as a bound for the stability number of a graph [47]. Hierarchies based on LP relaxations include the lift-and-project method of Balas, Ceria and Cornuéjols [11], the reformulation-linearization technique of Sherali and Adams [55], and the matrix-cuts approach of Lovász and Schrijver [48]. Researchers in the SAT community have studied the complexity of applying some of these techniques, and generalizations thereof, to specific classes of SAT problems (see the recent papers [14, 27, 26]). We note that semidefinite constraints may also be employed in the Lovász-Schrijver matrix-cuts approach, but in a different manner from that of the lifting paradigm we consider in this section.

The higher-liftings paradigm we consider is based on the Lasserre hierarchy of SDP relaxations of polynomial optimization problems. The idea behind this paradigm can be summarized as follows. Suppose that we have a discrete optimization problem on n binary variables. The SDP relaxation in the space of $(n + 1) \times (n + 1)$ symmetric matrices is called a first lifting. (For SAT, this corresponds to the Gap relaxation.) Note that, except for the first row, the rows and columns of the matrix variable in this relaxation are indexed by the binary variables themselves. To generalize this operation, we allow the rows and columns of the SDP relaxations to be indexed by *subsets* of the discrete variables in the formulation. These larger matrices can be interpreted as higher liftings, in the spirit of the second lifting for max-cut proposed by Anjos and Wolkowicz [9], and its generalization for 0-1 optimization independently proposed by Lasserre [41, 42]. A detailed analysis of the connections between the Sherali-Adams, Lovász-Schrijver, and Lasserre frameworks was done by Laurent [43]. In particular, Laurent showed that the Lasserre framework is the tightest among the three. This fact motivates the study of its application to the SAT problem.

For higher liftings of the max-cut problem, one of the theoretical questions that has been considered is to prove conditions on the rank of an optimal matrix for the SDP relaxation which ensure that the optimal value of the SDP is actually the optimal value of the underlying discrete problem. This question was settled for liftings of max-cut as follows. First, the rank-1 case is obvious since the optimal solution of the SDP is then a cut matrix (Theorem 8). For second liftings, a rank-2 guarantee of optimality was proved by Anjos and Wolkowicz [7]. This result was extended to the whole of the Lasserre hierarchy by Laurent [44] who showed that as the higher liftings for max-cut in the Lasserre hierarchy increase in dimension, correspondingly increasing rank values become sufficient for optimality. These rank-based conditions for optimality can be interpreted as a measure of the relative strength of the relaxations.

Applying directly the Lasserre approach to SAT, we would use the SDP relaxations Q_{K-1} (as defined in [41]) for $K = 1, 2, \dots, n$ where the matrix variable of Q_{K-1} has rows and columns indexed by all the subsets I with $|I| \leq K$. (Hence for $K = 1$, we obtain the matrix variable of the Gap relaxation.) The results in [42] imply that for $K = n$,

the feasible set of the resulting SDP relaxation is precisely the cut polytope. However, this SDP has dimension exponential in n . Indeed, the SDPs that must be solved when using this approach quickly become far too large for practical computation. For instance, even for second liftings (corresponding to $K = 2$) of max-cut problems, only problems with up to 27 binary variables were successfully solved in [2]. This limitation motivated the study of *partial higher liftings*, where we consider SDP relaxations which have a much smaller matrix variable, as well as fewer linear constraints. The objective of this approach is the construction of SDP relaxations which are linearly-sized with respect to the size of the SAT instance, and are thus more amenable to practical computation than the entire higher liftings. The construction of such partial liftings for SAT is particularly interesting because we can let the structure of the SAT instance specify exactly the structure of the SDP relaxation. We now outline the construction of these relaxations.

5.1 Derivation of Two SDP Relaxations

The strengthened SDP relaxations are based on the following proposition.

Proposition 1. *For $l(C_j) \geq 2$, clause $C_j = \bigvee_{k \in I_j} x_k \vee \bigvee_{k \in \bar{I}_j} \bar{x}_k$ is satisfied by $x_k = \pm 1, i \in I_j \cup \bar{I}_j$, if and only if*

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

where the coefficients $s_{j,k}$ are as defined in (13) above.

Proof: By construction of the coefficients $s_{j,k}$, the clause is satisfied if and only if $s_{j,k}x_k$ equals 1 for at least one $k \in I_j \cup \bar{I}_j$, or equivalently, if $\prod_{k \in I_j \cup \bar{I}_j} (1 - s_{j,k}x_k) = 0$. Expanding the product, we have

$$1 + \sum_{t=1}^{l(C_j)} (-1)^t \left[\sum_{T \subseteq I_j \cup \bar{I}_j, |T|=t} \left(\prod_{k \in T} s_{j,k}x_k \right) \right] = 0.$$

The result follows. ■

Using Proposition 1, we formulate the satisfiability problem as follows:

find x_1, \dots, x_n
s.t.

$$\sum_{t=1}^{l(C_j)} (-1)^{t-1} \left[\sum_{T \subseteq I_j \cup \bar{I}_j, |T|=t} \left(\prod_{k \in T} s_{j,k} \right) \left(\prod_{k \in T} x_k \right) \right] = 1, \quad j = 1, \dots, m$$

$$x_k^2 = 1, \quad k = 1, \dots, n$$

The next step is to formulate the problem in symmetric matrix space. Let \mathcal{P} denote the set of nonempty sets $I \subseteq \{1, \dots, n\}$ such that the term $\prod_{k \in I} x_k$ appears in the above formulation.

Also introduce new variables

$$x_I := \prod_{k \in I} x_k,$$

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

$$Y := \begin{pmatrix} 1 \\ x_{I_1} \\ \vdots \\ x_{I_{|\mathcal{P}|}} \end{pmatrix} \begin{pmatrix} 1 \\ x_{I_1} \\ \vdots \\ x_{I_{|\mathcal{P}|}} \end{pmatrix}^T,$$

whose $|\mathcal{P}| + 1$ rows and columns are indexed by $\{\emptyset\} \cup \mathcal{P}$. By construction of Y , we have that $Y_{\emptyset, I} = x_I$ for all $I \in \mathcal{P}$. Using these new variables, we can formulate the SAT problem as:

$$\begin{aligned} \text{find } & Y \in \mathcal{S}^{1+|\mathcal{P}|} \\ \text{s.t. } & \\ & \sum_{t=1}^{l(C_j)} (-1)^{t-1} \left[\sum_{T \subseteq I_j \cup \bar{I}_j, |T|=t} \left(\prod_{k \in T} s_{j,k} \right) Y_{\emptyset, T} \right] = 1, \quad j = 1, \dots, m \\ & \text{diag}(Y) = e \\ & \text{rank}(Y) = 1 \\ & Y \succeq 0 \end{aligned} \tag{24}$$

Relaxing this formulation by omitting the rank constraint would give an SDP relaxation for SAT. However, in order to tighten the resulting SDP relaxation, we first add *redundant* constraints to this formulation. This approach of adding redundant constraints to the problem formulation so as to tighten the resulting SDP relaxation is discussed in detail for the max-cut problem in [8].

The constraint $\text{rank}(Y) = 1$ implies that for every triple I_1, I_2, I_3 of subsets of indices in \mathcal{P} such that the symmetric difference of any two equals the third, the following three equations hold:

$$Y_{\emptyset, I_1} = Y_{I_2, I_3}, \quad Y_{\emptyset, I_2} = Y_{I_1, I_3}, \quad \text{and} \quad Y_{\emptyset, I_3} = Y_{I_1, I_2}. \tag{25}$$

Hence we can add some or all of these redundant constraints to formulation (24) without affecting its validity. We choose to add the equations of the form (25) for all the triples $\{I_1, I_2, I_3\} \subseteq \mathcal{P}$ satisfying the symmetric difference condition and such that $(I_1 \cup I_2 \cup I_3) \subseteq (I_j \cup \bar{I}_j)$ for some clause j . Beyond the fact that they tighten the SDP relaxation, this particular subset of redundant constraints was chosen because it suffices for proving the main theoretical result (Theorem 14 below). We refer to the resulting SDP relaxation as

the R_3 relaxation:

$$\begin{aligned}
& \text{find } Y \in \mathcal{S}^{1+|\mathcal{P}|} \\
& \text{s.t.} \\
(R_3) \quad & \sum_{t=1}^{l(C_j)} (-1)^{t-1} \left[\sum_{T \subseteq I_j \cup \bar{I}_j, |T|=t} \left(\prod_{k \in T} s_{j,k} \right) Y_{\emptyset, T} \right] = 1, \quad j = 1, \dots, m \\
& Y_{\emptyset, I_1} = Y_{I_2, I_3}, \quad Y_{\emptyset, I_2} = Y_{I_1, I_3}, \quad \text{and } Y_{\emptyset, I_3} = Y_{I_1, I_2}, \quad \forall \{I_1, I_2, I_3\} \subseteq \mathcal{P} \\
& \quad \text{such that } I_1 \Delta I_2 = I_3 \text{ and } (I_1 \cup I_2 \cup I_3) \subseteq (I_j \cup \bar{I}_j) \text{ for some clause } j \\
& \text{diag}(Y) = e \\
& Y \succeq 0
\end{aligned}$$

where $I_i \Delta I_j$ denotes the symmetric difference of I_i and I_j . The R_3 terminology relates to the fact that a rank-3 guarantee holds for this SDP relaxation (see Theorem 14).

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 $Y_{I_1, I_2} = Y_{I_3, I_4}$, where $\{I_1, I_2, I_3, I_4\} \subseteq \{\emptyset\} \cup \mathcal{P}$ and $I_1 \Delta I_2 = I_3 \Delta I_4$, then we would have obtained the Lasserre relaxation Q_{K-1} for this problem. However, as mentioned earlier, the resulting SDP has a matrix variable of dimension $|\mathcal{P}| + 1 = O(n^K)$, which is too large for practical computational purposes, even when $K = 2$. In contrast, 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. The matrix variable of R_3 has dimension $O(m * 2^K) = O(m)$, since for practical SAT instances K is a very small constant. 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.

The computational performance of R_3 was studied in [5] and it was observed that when used in a branching algorithm, the SDP relaxation is still impractical for solving SAT problems with more than about 100 clauses, unless the solution can be obtained without resorting to branching (see Sections 5.3 and 6.2 for more details on the computational performance of R_3). Therefore, a more compact semidefinite relaxation, denoted R_2 , was proposed in [3]. This relaxation is also a strengthening of the Gap relaxation, and is computationally superior to R_3 because of significant reductions in the dimension of the matrix variable and in the number of linear constraints. The matrix variable of the compact SDP relaxation is a principal submatrix of the matrix variable in R_3 , and it was shown in [3] that although the SDP relaxation R_2 does not retain the rank-3 guarantee, it has a rank-2 guarantee. Hence, it is a compromise relaxation between the Gap and R_3 , and it completes a trio of linearly sized semidefinite relaxations with correspondingly stronger rank guarantees.

To obtain R_2 , we replace \mathcal{P} by a smaller set of column indices, namely

$$\mathcal{O} := \{I \mid I \subseteq (I_j \cup \bar{I}_j) \text{ for some } j, \ |I| \bmod 2 = 1\}.$$

The set \mathcal{O} consists of the sets of odd cardinality in \mathcal{P} . It is clear that the sets in \mathcal{P} of even cardinality corresponding to terms appearing in the above formulation are all generated as symmetric differences of the sets in \mathcal{O} . Having chosen our set of column indices, we

introduce new variables

$$x_I := \prod_{k \in I} x_k,$$

for each $I \in \mathcal{O}$, and define the rank-1 matrix

$$Y := \begin{pmatrix} 1 \\ x_{I_1} \\ \vdots \\ x_{I_{|\mathcal{O}|}} \end{pmatrix} \begin{pmatrix} 1 \\ x_{I_1} \\ \vdots \\ x_{I_{|\mathcal{O}|}} \end{pmatrix}^T,$$

whose $|\mathcal{O}| + 1$ rows and columns are indexed by $\emptyset \cup \mathcal{O}$. By construction, we have $Y_{\emptyset, I} = x_I$ for all $I \in \mathcal{O}$ and $Y_{\{\min(I)\}, I \Delta \{\min(I)\}} = x_I$ for all $I \in \mathcal{P} \setminus \mathcal{O}$. (Note that $T \Delta \{\min(T)\}$ is an element of $\mathcal{P} \setminus \mathcal{O}$ when $|T|$ is even.) This means that the new variables corresponding to subsets of logical variables of odd cardinality appear exactly once in the first row of Y , and the new variables corresponding to subsets of even cardinality have the “representative” matrix entries $Y_{\{\min(T)\}, T \Delta \{\min(T)\}}$. By the same approach as for R_3 above, we obtain the R_2 relaxation:

$$\begin{aligned} & \text{find } Y \in \mathcal{S}^{|\mathcal{O}|+1} \\ & \text{s.t.} \\ (R_2) \quad & \sum_{t=1}^{l(C_j)} (-1)^{t-1} \left[\sum_{T \subseteq I_j \cup \bar{I}_j, |T|=t} \left(\prod_{k \in T} s_{j,k} \right) Y(T) \right] = 1, \quad j = 1, \dots, m \\ & Y(I_1) = Y_{I_2 \cap I_3, I_2 \cup I_3}, \quad \forall \{I_1, I_2, I_3\} \subseteq \mathcal{P} \setminus \mathcal{O} \text{ such that } I_1 = I_2 \Delta I_3 \\ & \quad \text{and } (I_1 \cup I_2 \cup I_3) \subseteq (I_j \cup \bar{I}_j) \text{ for some clause } j \\ & \text{diag}(Y) = e \\ & Y \succeq 0 \end{aligned}$$

where

$$Y(T) = \begin{cases} Y_{\emptyset, T}, & |T| \text{ odd;} \\ Y_{\{\min(T)\}, T \Delta \{\min(T)\}}, & |T| \text{ even.} \end{cases}$$

Note that for 2-SAT, R_2 is precisely the Gap relaxation.

5.2 Theoretical Properties of the Strengthened SDP Relaxations

It is clear that if the propositional formula Φ is satisfiable, then using any model it is straightforward to construct a rank-one matrix Y feasible for every SDP relaxation. The contrapositive of this statement gives a sufficient condition for proving unsatisfiability using the SDP relaxations.

Lemma 2. *Given a propositional formula in CNF, if any one of the SDP relaxations is infeasible, then the CNF formula is unsatisfiable.*

Several results are known for conditions on the rank of a feasible matrix Y which guarantee that a model can be obtained from Y . Such rank conditions for an SDP relaxation are significant for two reasons. Firstly, from a theoretical point of view, the rank value can be viewed as a measure of the strength of the relaxation. For general instances of SAT,

the Gap relaxation will prove satisfiability only if a feasible matrix of rank 1 is obtained, whereas for our relaxation any matrix with rank 1, 2, or 3 immediately proves satisfiability. Furthermore, the higher rank value also reflects the relaxation's greater ability to detect unsatisfiability, compared to the Gap relaxation. Secondly, from a practical point of view, the rank conditions are helpful because of the inevitable occurrence of SDP solutions with high rank when there are multiple optimal solutions to the original binary problem. This happens because interior-point algorithms typically converge to a matrix in the interior of the optimal face of \mathcal{S}_+^n , and in the presence of multiple solutions this face contains matrices of rank higher than one. Therefore, the ability to detect optimality for as high a rank value as possible may allow an enumerative algorithm to avoid further branching steps and potentially yield a significant reduction in computational time.

The results on the SDP relaxations are summarized in the following theorem:

Theorem 14. *Given any propositional formula in CNF, consider the SDP relaxations presented. Then*

- *If at least one of the Gap, R_2 , or R_3 relaxations is infeasible, then the formula is unsatisfiable.*
- *If Y is feasible for the Gap and $\text{rank } Y = 1$, then a truth assignment satisfying the formula can be obtained from Y .*
- *If Y is feasible for R_2 and $\text{rank } Y \leq 2$, then a truth assignment satisfying the formula can be obtained from Y .*
- *If Y is feasible for R_3 and $\text{rank } Y \leq 3$, then a truth assignment satisfying the formula can be obtained from Y .*

For any instance of SAT, the SDP relaxations Gap, R_2 , and R_3 form a trio of linearly sized SDP relaxations for SAT with correspondingly stronger rank guarantees. If we use R_1 to refer to the Gap relaxation, then the names of the relaxations reflect their increasing strength in the following sense: For $k = 1, 2, 3$, any feasible solution to the relaxation R_k with rank at most k proves satisfiability of the corresponding 3-SAT instance. Furthermore, the increasing values of k also reflect an improving ability to detect unsatisfiability, and an increasing computational time for solving the relaxation. Nonetheless, the dimensions of the relaxations grow only linearly with n and m . If we consider the Lasserre relaxations Q_{K-1} for $K \geq 3$, results such as Theorem 14 (and stronger ones) would follow from [44, Theorem 21]. The important point here is that Theorem 14 holds for the significantly smaller relaxations obtained as partial liftings.

To conclude this Section, we sketch the proof that if Y is feasible for the R_2 relaxation and $\text{rank } Y \leq 2$, then Y yields a truth assignment proving that the SAT instance is satisfiable. The result follows from the following lemmata.

Lemma 3. [7, Lemma 3.11] *Suppose that the matrix*

$$\begin{pmatrix} 1 & a & b & c \\ a & 1 & c & b \\ b & c & 1 & a \\ c & b & a & 1 \end{pmatrix}$$

is positive semidefinite and has rank at most two. Then at least one of a, b, c equals ± 1 .

Lemma 4. *If Y is feasible for the R_2 relaxation and $\text{rank } Y \leq 2$, then each column indexed by a subset $\gamma \in \mathcal{O}$ of cardinality $|\gamma| \geq 3$ is a ± 1 multiple of a column corresponding to a singleton in \mathcal{O} .*

Proof: For every subset γ with $|\gamma| \geq 3$, let i_1, i_2 , and i_3 denote any three variables in γ , and let $\bar{\gamma} = \gamma \setminus \{i_1, i_2, i_3\}$ be the set of all the other variables in γ . (Note that $\bar{\gamma}$ may be empty.) Let Y_γ denote the principal submatrix of the matrix variable Y corresponding to the rows and columns of Y indexed by $\{\emptyset, \{i_1\} \cup \bar{\gamma}, \{i_2\} \cup \bar{\gamma}, \{i_3\} \cup \bar{\gamma}, \gamma\}$. The principal submatrix Y_γ has the form

$$\begin{pmatrix} 1 & Y_{\emptyset, \{i_1\} \cup \bar{\gamma}} & Y_{\emptyset, \{i_2\} \cup \bar{\gamma}} & Y_{\emptyset, \{i_3\} \cup \bar{\gamma}} & Y_{\emptyset, \gamma} \\ Y_{\emptyset, \{i_1\} \cup \bar{\gamma}} & 1 & Y_{\{i_1\}, \{i_2\}} & Y_{\{i_1\}, \{i_3\}} & Y_{\{i_2\}, \{i_3\}} \\ Y_{\emptyset, \{i_2\} \cup \bar{\gamma}} & Y_{\{i_1\}, \{i_2\}} & 1 & Y_{\{i_2\}, \{i_3\}} & Y_{\{i_1\}, \{i_3\}} \\ Y_{\emptyset, \{i_3\} \cup \bar{\gamma}} & Y_{\{i_1\}, \{i_3\}} & Y_{\{i_2\}, \{i_3\}} & 1 & Y_{\{i_1\}, \{i_2\}} \\ Y_{\emptyset, \gamma} & Y_{\{i_2\}, \{i_3\}} & Y_{\{i_1\}, \{i_3\}} & Y_{\{i_1\}, \{i_2\}} & 1 \end{pmatrix}$$

Since $\text{rank } Y \leq 2 \Rightarrow \text{rank } Y_\gamma \leq 2$, it follows by Lemma 3 that at least one of $Y_{\{i_1\}, \{i_2\}}$, $Y_{\{i_1\}, \{i_3\}}$, and $Y_{\{i_2\}, \{i_3\}}$ equals ± 1 .

Suppose (without loss of generality) that $Y_{\{i_1\}, \{i_2\}} = \delta$, where $\delta^2 = 1$. Then for every row index I' in \mathcal{O} the principal submatrix indexed by $\{I', \gamma \setminus \{i_1, i_2\}, \gamma\}$ has the form

$$\begin{pmatrix} 1 & Y_{I', \gamma \setminus \{i_1, i_2\}} & Y_{I', \gamma} \\ Y_{I', \gamma \setminus \{i_1, i_2\}} & 1 & \delta \\ Y_{I', \gamma} & \delta & 1 \end{pmatrix}$$

and by Lemma 1, it follows that $Y_{I', \gamma \setminus \{i_1, i_2\}} = \delta Y_{I', \gamma}$. Hence, the entire column γ equals δ times the column $\gamma \setminus \{i_1, i_2\}$.

If $\gamma \setminus \{i_1, i_2\}$ is a singleton, we are done. Otherwise, repeat the argument using $\gamma \setminus \{i_1, i_2\}$ in place of γ . Eventually a singleton will be reached, and then the column corresponding to the singleton will be a ± 1 multiple of the column γ . \blacksquare

Let us now consider the implications of Lemma 4 for the constraints enforcing satisfiability. For each term of the form $Y(T)$ such that $|T| \geq 3$, Lemma 4 implies that

- it is equal to ± 1 , or
- if $|T|$ is odd, then it is equal to $\pm Y(\{i\})$ for some $i \in T$, or
- if $|T|$ is even, then it is equal to $\pm Y(\{i, j\})$ for some $\{i, j\} \subseteq T$.

Therefore, in the constraints enforcing satisfiability, all the terms corresponding to subsets of cardinality greater than 2 can be replaced either by ± 1 or by a term corresponding to a subset of cardinality at most 2. (The case of 3-SAT is discussed in detail in [3].)

Now, consider a reduced SDP obtained as follows: For every constraint, make all the substitutions for terms corresponding to subsets of cardinality greater than 2 using the observation above. Simplifying the result, each constraint becomes either a tautology or a

constraint of the form corresponding to a clause of length 2. Take every constraint arising from the latter case, and let the matrix \tilde{Y} be indexed by the set of singletons $\{i_k\}$ such that $Y(\{i_k\})$ appears in at least one of these constraints. The SDP defined by the matrix \tilde{Y} and the constraints arising from the second case above, plus the constraint $\tilde{Y} \succeq 0$, is precisely of the form of the Gap relaxation for an instance of 2-SAT. This means that Lemma 4 allows us to reduce the problem to an instance of 2-SAT. Finally, recall that we have a matrix Y feasible for the original R_2 relaxation, and observe that its principal submatrix indexed by the set of singletons used to define \tilde{Y} is feasible for this reduced SDP. By Theorem 12, this implies that the instance of 2-SAT is satisfiable. Consider a model for the 2-SAT instance obtained using the algorithm following Theorem 12. For each of the variables in the original instance of SAT that are not present in this truth assignment, either they should be set equal to (or to the negative of) one of the variables in this assignment (according to the analysis above), or they are “free” and can be assigned the value +1, say. We thus obtain a model for the original SAT instance, and have proved the claim.

5.3 Computational Proofs of Infeasibility for a Class of Hard Instances

As mentioned at the beginning of this section, an important objective in the study of partial higher liftings for SAT is the construction of SDP relaxations which are more amenable to practical computation than the entire higher liftings. The potential of this computational approach has been shown in the results obtained by the author in [4, 5]. The results in those papers show that the R_3 relaxation yielded proofs of unsatisfiability for some hard instances with up to 260 variables and over 400 clauses. In particular, R_3 is able to prove the unsatisfiability of the smallest unsatisfiable instance that remained unsolved during the SAT Competitions in 2003 and 2004.

Researchers in SDP have developed a variety of excellent solvers, most of which are freely available. An extensive listing of solvers is available at [32]. For application to SAT, it is important to use a solver which, when given an infeasible SDP, provides us with a certificate of infeasibility, because that certificate is for us a proof of unsatisfiability for the SAT instance.

The results we present are for randomly generated SAT instances obtained using the generator `hgen8`. A set of 12 instances generated using `hgen8` was submitted for the SAT Competition 2003 (see [1]). The source code to generate these instances, which includes an explanation of their structure, is available at [35]. The results in Table 1 were obtained using the solver SDPT3 (version 3.0) [57, 56] with its default settings, and running on a 2.4GHz Pentium IV with 1.5Gb of RAM. (The SDP relaxations for the remaining instances in the set were too large for the computing resources available.) In particular, the R_3 relaxation was able to prove the unsatisfiability of the instance n260-01 in Table 1, which was the smallest unsatisfiable instance that remained unsolved during the 2003 Competition. (An instance remained unsolved if none of the top five solvers was able to solve it in two hours, running on an Athlon 1800+ with 1Gb of RAM.) Indeed, the SDP relaxation appears to be quite effective on the type of instances generated by `hgen8`, as we randomly generated several more instances of varying size and all the corresponding SDP relaxations were infeasible. The additional results are presented in Table 2.

Table 1. Results for the Nine hgen8 Instances from SAT Competition 2003

Problem	# of variables	# of clauses	Size of Y	# of SDP constraints	SDP is infeasible	Total CPU seconds	Solved at SAT 2003
n120-01	120	197	542	3862	Yes	704	Yes
n120-02	120	193	537	3838	Yes	611	Yes
n120-03	120	193	539	3846	Yes	583	Yes
n180-01	180	279	793	5668	Yes	2194	Yes
n180-02	180	279	793	5668	Yes	2142	Yes
n180-03	180	280	791	5661	Yes	2188	Yes
n260-01	260	391	1132	8096	Yes	6938	No
n260-02	260	404	1143	8153	Yes	7455	No
n260-03	260	399	1134	8112	Yes	7678	No

n120-01 denotes problem `hgen8-n120-01-S563767109.shuffled-as.sat03-875`

n120-02 denotes problem `hgen8-n120-02-S1654058060.shuffled-as.sat03-876`

n120-03 denotes problem `hgen8-n120-03-S1962183220.shuffled-as.sat03-877`

n180-01 denotes problem `hgen8-n180-01-S1524349002.shuffled-as.sat03-880`

n180-02 denotes problem `hgen8-n180-02-S1125510326.shuffled-as.sat03-881`

n180-03 denotes problem `hgen8-n180-03-S1436192352.shuffled-as.sat03-882`

n260-01 denotes problem `hgen8-n260-01-S1597732451.shuffled-as.sat03-885`

n260-02 denotes problem `hgen8-n260-02-S1396509323.shuffled-as.sat03-886`

n260-03 denotes problem `hgen8-n260-03-S722413478.shuffled-as.sat03-887`

Table 2. Results for the Randomly Generated hgen8 Instances

Problem	# of variables	# of clauses	Size of Y	# of SDP constraints	SDP is infeasible	Total CPU seconds
200-01	200	309	881	6290	Yes	2821
200-02	200	314	882	6299	Yes	3243
200-03	200	306	877	6271	Yes	2530
220-01	220	339	966	6900	Yes	3843
220-02	220	344	969	6917	Yes	4279
220-03	220	341	967	6906	Yes	4757
240-01	240	362	1044	7475	Yes	5455
240-02	240	365	1047	7490	Yes	5563
240-03	240	366	1046	7487	Yes	5376

It is important to point out that the effectiveness of the SDP approach for these instances is due to the fact that no branching is needed to obtain a certificate of infeasibility for the SDP relaxation. Computational results on instances of 3-SAT from the Uniform Random-3-SAT benchmarks available at [36] are reported in Section 6, and it is observed there that if branching is required, then the computational effort required by the SDP-based approach is still too large in comparison to other SAT algorithms in the literature. Nonetheless, the SDP approach already complements existing SAT solvers in the sense that these difficult `hgen8` instances can be solved in reasonable time. (Incidentally, the `n260-01` instance remained unsolved in the SAT Competition 2004.) These results motivate our current research considering this and other classes of hard instances for which the partial higher liftings approach may be effective with little or no branching. The most recent results in this direction are reported in Section 7.

6. Comparisons of the SDP Relaxations for 3-SAT

6.1 Theoretical Comparison

For the case of 3-SAT, we can make some specific comparisons between the feasible sets of the Karloff-Zwick relaxation and the three SDP relaxations in Theorem 14.

It was observed in [19] that for 3-SAT, the Gap relaxation has an interesting connection to the relaxation (12) of Karloff and Zwick. For each clause of length 3, the linear constraint in the Gap relaxation is precisely equivalent to the sum of the three linear constraints used in the Karloff-Zwick relaxation. Using the notation defined in (13) to account for negated variables, the three Karloff-Zwick constraints may be rewritten as:

$$\begin{aligned} s_{j,i_1} Y_{0,i_1} + s_{j,i_2} Y_{0,i_2} - s_{j,i_1} s_{j,i_3} Y_{i_1,i_3} - s_{j,i_2} s_{j,i_3} Y_{i_2,i_3} &\geq 0, \\ s_{j,i_1} Y_{0,i_1} + s_{j,i_3} Y_{0,i_3} - s_{j,i_1} s_{j,i_2} Y_{i_1,i_2} - s_{j,i_2} s_{j,i_3} Y_{i_2,i_3} &\geq 0, \\ s_{j,i_2} Y_{0,i_2} + s_{j,i_3} Y_{0,i_3} - s_{j,i_1} s_{j,i_2} Y_{i_1,i_2} - s_{j,i_1} s_{j,i_3} Y_{i_1,i_3} &\geq 0. \end{aligned}$$

for each 3-clause with variables $\{p_{i_1}, p_{i_2}, p_{i_3}\}$.

Now consider any matrix Y feasible for the R_2 and R_3 relaxations. The principal submatrix of Y corresponding to the rows and columns indexed by $\{\emptyset\} \cup \{x_k : k = 1, \dots, n\}$ is a positive semidefinite matrix with the same structure as the matrix variable in the Gap relaxation. As for the constraints, it is clear that for clauses of length 1 or 2, the linear constraints expressing satisfiability are equivalent to those in either the Karloff-Zwick or Gap relaxations. For clauses of length 3, the linear constraint in R_2 and R_3 can be rewritten as

$$\begin{aligned} s_{j,i_2} Y_{0,i_2} + s_{j,i_3} Y_{0,i_3} - s_{j,i_1} s_{j,i_2} Y_{0,i_1 i_2} - s_{j,i_1} s_{j,i_3} Y_{0,i_1 i_3} = \\ 1 - s_{j,i_1} Y_{0,i_1} + s_{j,i_2} s_{j,i_3} Y_{0,i_2 i_3} - s_{j,i_1} s_{j,i_2} s_{j,i_3} Y_{0,i_1 i_2 i_3} \end{aligned}$$

and the positive semidefiniteness of Y implies that the right-hand side is always non-negative. Relaxing the equation to an inequality, we have

$$s_{j,i_2} Y_{0,i_2} + s_{j,i_3} Y_{0,i_3} - s_{j,i_1} s_{j,i_2} Y_{0,i_1 i_2} - s_{j,i_1} s_{j,i_3} Y_{0,i_1 i_3} \geq 0,$$

which is the first constraint above for the Karloff-Zwick relaxation. The other two constraints can be shown to hold by similar arguments. Hence, the SDP relaxation R_2 , and therefore also R_3 , for 3-SAT are at least as tight as the Karloff-Zwick relaxation.

In summary, we have the following theorem:

Theorem 15. *For instances of 3-SAT, the feasible sets of the Gap, Karloff-Zwick (KZ), R_2 , and R_3 relaxations satisfy*

$$\mathcal{F}_{R_3} \subseteq \mathcal{F}_{R_2} \subseteq \mathcal{F}_{KZ} \subseteq \mathcal{F}_{Gap}.$$

6.2 Computational Comparison

The results of a computational study of the performance of R_1 , R_2 , and R_3 were reported in [3]. A branching algorithm was implemented in Matlab running on a 2.4 GHz Pentium IV with 1.5Gb. The test suite consisted of both satisfiable and unsatisfiable instances of 3-SAT from the Uniform Random-3-SAT benchmarks available at [36]. Two sets of problems were considered, with $n = 50$ and $n = 75$ variables, and $m = 218$ and $m = 325$ clauses respectively. A total of 40 instances of 3-SAT from each set were used, half of them satisfiable and the other half unsatisfiable. For all the relaxations, the algorithm was stopped after 2 hours on the instances with 50 variables, and after 3 hours on the instances with 75 variables. The results are for small problems, but they clearly illustrate the tradeoffs involved in the choice of SDP relaxation as well as the advantage of the R_2 relaxation over the other two relaxations when applied together with a branching algorithm. The main conclusions were that:

- The Gap relaxation is solved most quickly of all three, but the branching algorithm reaches a large depth in the search tree before it stops. As a result, its total time is higher than that of the algorithm using R_2 .
- The opposite happens with the algorithm using R_3 : each SDP relaxation takes much longer to solve, but the depth reached in the search tree is less than for the other two relaxations.

Therefore, in comparison with the other two relaxations, the R_2 relaxation is the most effective, and it can be routinely used to prove both satisfiability and unsatisfiability for instances with a few hundred clauses. However, we observe that the computational time for the branching algorithm using R_2 still increases quite rapidly. This was illustrated in [3] by allowing the branching algorithm using R_2 to run to completion on 40 instances of 3-SAT with $n = 100$ and $m = 430$ from the same set of benchmarks, evenly divided between satisfiable and unsatisfiable instances. The results show that proofs of satisfiability require over one hour on average, and proofs of unsatisfiability over six hours.

Although the Karloff-Zwick relaxation was not considered, it is clear that the computational time for any SDP-based algorithm is dominated by the effort required to solve the SDPs, and that regardless of the choice of SDP relaxation, the computational effort is still too high to be competitive with other SAT solvers whenever branching is actually used. Therefore, even though branching is not always necessary (as observed in Section 5.3), the competitiveness of the SDP approach depends on the development of novel algorithms for solving SDPs by taking advantage of their structure. Important advances have been made recently on algorithms for SDP that exploit certain types of structure (see e.g [29, 51]), and current research is considering how to apply them for solving the SAT relaxations.

7. Recent Research Results

The application of SDP to SAT problems continues to be an active area of research. We close this survey with a summary of two recent working papers on the application of SDP to MAX-SAT and SAT.

7.1 Solving MAX-SAT Using SDP Formulations of Sums of Squares

In their working paper [61], van Maaren and van Norden consider the application of Hilbert's Positivstellensatz to MAX-SAT. The idea is to formulate MAX-SAT as a global polynomial optimization problem, akin to the approaches we have seen, but 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 starting point for this approach is the observation that for each clause $C_j = \bigvee_{k \in I_j} x_k \vee \bigvee_{k \in \bar{I}_j} \bar{x}_k$ and for each assignment of values to $x_k \in \{-1, 1\}$, $i \in I_j \cup \bar{I}_j$, the polynomial defined as

$$F_{C_j}(x) := \prod_{k \in I_j \cup \bar{I}_j} \frac{1}{2}(1 - s_{j,k}x_k),$$

where $x = (x_1, \dots, x_n)^T$ and the parameters $s_{j,k}$ are as defined in (13), satisfies

$$F_{C_j}(x) = \begin{cases} 0, & \text{if } C_j \text{ is satisfied by the truth assignment represented by } x \\ 1, & \text{otherwise.} \end{cases}$$

With a given CNF formula Φ , we thus associate two aggregate polynomials:

$$F_\Phi(x) := \sum_{j=1}^m (F_{C_j}(x))^2 \quad \text{and} \quad F_\Phi^B(x) := \sum_{j=1}^m F_{C_j}(x).$$

Clearly, $F_\Phi(x)$ is a non-negative polynomial on \mathfrak{R}^n , $F_\Phi^B(x)$ is non-negative on $\{-1, 1\}^n$, and for $x \in \{-1, 1\}^n$, both polynomials give the number of unsatisfied clauses. Hence, MAX-SAT is equivalent to the minimization of either of these polynomials over $\{-1, 1\}^n$.

A first SDP relaxation is obtained as follows. Suppose we are given a column vector β of monomials in the variables x_1, \dots, x_n and a polynomial $p(x)$. Then $p(x)$ can be written as an SOS in terms of the elements of β if and only if there exists a matrix $S \succeq 0$ such that $\beta^T S \beta = p$ (see e.g. [52]). Note that by Theorem 5,

$$\beta^T S \beta = p \quad \Rightarrow \quad \|W\beta\|_2^2 = p, \quad \text{where } S = W^T W,$$

and hence we have an explicit decomposition of p as an SOS. Therefore, if

$$\begin{aligned} g^* &= \max g \\ &\text{s.t.} \\ &F_\Phi(x) - g = \beta^T S \beta \\ &S \succeq 0, \end{aligned} \tag{26}$$

then $m - g^*$ is an upper bound for MAX-SAT. Note that since β is fixed, the equation $F_{\Phi}(x) - g = \beta^T S \beta$ is linear in S and g , and hence problem (26) is indeed an SDP. Also, $g^* \geq 0$ since $F_{\Phi}(x)$ is an SOS by definition.

Another SDP relaxation is obtained by considering $F_{\Phi}^{\mathcal{B}}(x)$, but here we have to work modulo $I_{\mathcal{B}}$, the ideal generated by the polynomials $x_k^2 - 1, k = 1, \dots, n$. (The fact that each polynomial that is non-negative on $\{-1, 1\}^n$ can be expressed as an SOS modulo $I_{\mathcal{B}}$ follows from the work of Putinar [53].) The resulting SDP problem is

$$\begin{aligned} g_{\mathcal{B}}^* &= \max g \\ &\text{s.t.} \\ &F_{\Phi}^{\mathcal{B}}(x) - g \equiv \beta^T S \beta \text{ modulo } I_{\mathcal{B}} \\ &S \succeq 0 \end{aligned} \tag{27}$$

and thus $m - g_{\mathcal{B}}^*$ is also an upper bound for MAX-SAT. For the remainder of this section, we focus on the SDP approach using (27).

The key point in this SOS approach is the choice of β , and van Maaren and van Norden consider the following possibilities:

- β_{GW} is the basis containing $1, x_1, \dots, x_n$;
- β_{p} is the basis containing $1, x_1, \dots, x_n$, plus the monomial $x_{k_1} x_{k_2}$ for each pair of variables that appear together in a clause;
- β_{ap} is the basis containing $1, x_1, \dots, x_n$, plus the monomials $x_{k_1} x_{k_2}$ for all pairs of variables;
- β_{t} is the basis containing $1, x_1, \dots, x_n$, plus the monomial $x_{k_1} x_{k_2} x_{k_3}$ for each triple of variables that appear together in a clause;
- β_{pt} is the basis containing $1, x_1, \dots, x_n$, plus the monomial $x_{k_1} x_{k_2}$ for each pair of variables that appear together in a clause, plus $x_{k_1} x_{k_2} x_{k_3}$ for each triple of variables that appear together in a clause.

The SDP relaxations are referred to as SOS_{GW} , SOS_{p} , SOS_{ap} , SOS_{t} , and SOS_{pt} respectively.

Starting with MAX-2-SAT, van Maaren and van Norden prove that SOS_{GW} is precisely the dual of the SDP (10). Furthermore, they show that for each triple $x_{k_1} x_{k_2} x_{k_3}$, adding the monomials $x_{k_1} x_{k_2}$, $x_{k_1} x_{k_3}$, and $x_{k_2} x_{k_3}$ gives an SDP relaxation at least as tight as that obtained by adding the corresponding triangle inequality to (10). A comparison of the relaxations SOS_{pt} and (12) for MAX-3-SAT is also provided. The results are summarized in the following theorem:

Theorem 16. [61]

For every instance of MAX-2-SAT,

- *The SDP relaxation SOS_{GW} gives the same upper bound as the relaxation (10) of Goemans and Williamson.*
- *The SDP relaxation SOS_{ap} is at least as tight as the Feige-Goemans relaxation consisting of (10) plus all the triangle inequalities (9).*

For every instance of MAX-3-SAT, the SDP relaxation SOS_{pt} provides a bound at least as tight as the Karloff-Zwick relaxation (12).

From the computational point of view, van Maaren and van Norden provide computational results comparing several of these relaxations on instances of varying sizes and varying ratios of number of clauses to number of variables. They also propose rounding schemes for MAX-2-SAT and MAX-3-SAT based on SOS_p and SOS_t respectively, and present preliminary results comparing their performance with the rounding schemes presented in Section 3 above.

The SOS approach can also be applied to obtain proofs of unsatisfiability. It is straightforward to prove that:

Proposition 2. *Given a formula Φ in CNF, if there exists a monomial basis β and an $\epsilon > 0$ such that $F_{\Phi}^{\mathcal{B}}(x) - \epsilon$ is a SOS modulo $I_{\mathcal{B}}$, then Φ is unsatisfiable.*

van Maaren and van Norden compare the performance of the R_3 relaxation with the SOS approach using either SOS_t or SOS_{pt} . Their preliminary results suggest that SOS_{pt} offers the best performance.

7.2 An Explicit Semidefinite Characterization of SAT for Tseitin Instances

The results in this section apply to a specific class of SAT instances that has been studied for over 30 years, and is known to be hard for many proof systems. They are the Tseitin propositional formulas, first introduced in [58]. These instances are constructed using graphs whose vertices are points on the plane with integer coordinates, and whose edges are segments of unit length along the coordinate axes.

Consider a $p \times q$ toroidal grid graph, and label the rows $\{0, 1, \dots, p-1\}$ and the columns $\{0, 1, \dots, q-1\}$. Identify each vertex by a pair (i, j) with $i \in \{0, 1, \dots, p-1\}$ and $j \in \{0, 1, \dots, q-1\}$. Each vertex (i, j) has degree four, and its four incident edges are denoted by:

$$\{(i-1, j), (i, j)\}, \{(i+1, j), (i, j)\}, \{(i, j-1), (i, j)\}, \{(i, j), (i, j+1)\},$$

where (here and in the sequel) the subtractions and sums are taken mod p for the first index, and mod q for the second index. For each vertex (i, j) , fix the parameter $t(i, j) \in \{0, 1\}$, and associate with each edge a boolean variable:

$$\begin{aligned} v^r(i, j) & \text{ is the variable corresponding to the edge } \{(i+1, j), (i, j)\} \\ v^d(i, j) & \text{ is the variable corresponding to the edge } \{(i, j), (i, j+1)\} \end{aligned}$$

Thus, there are $2pq$ boolean variables in the SAT instance. For notation purposes, since vertex (i, j) has two other edges incident to it, we further define $v^u(i, j) := v^d(i-1, j)$ and $v^l(i, j) := v^r(i, j-1)$. Furthermore, each vertex (i, j) contributes eight clauses to the SAT instance, and the structure of the clauses is determined by the value of $t(i, j)$:

1. if $t(i, j) = 0$ then all clauses of length four on $v^l(i, j), v^r(i, j), v^u(i, j), v^d(i, j)$ with an odd number of negated variables are added; and
2. if $t(i, j) = 1$ then all clauses of length four on $v^l(i, j), v^r(i, j), v^u(i, j), v^d(i, j)$ with an even number of negated variables are added.

We denote the eight clauses thus obtained by $C_\tau(i, j), \tau = 1, \dots, 8$. Hence there are $8pq$ clauses in the SAT instance. It is well known that the SAT instance is unsatisfiable if and only if the sum of the $t(i, j)$ is odd (see e.g. [15, Lemma 10]).

In the recent working paper [6], the author proposes an SDP problem which characterizes the satisfiability of these instances, and is of dimension linear in the size of the instance. This is a result in the same vein as Theorems 12 and 13 for the Gap relaxation. These are not the first proofs that Tseitin or pigeonhole instances can be solved in polynomial-time (see e.g. [26]), but again we stress that the SDP approach not only establishes satisfiability or unsatisfiability in polynomial-time, but also computes an explicit proof of it in polynomial-time.

To construct the SDP problem, let $J_\tau(i, j)$ denote the set of variables negated in clause τ corresponding to vertex (i, j) . For each boolean variable $v^\sigma(i, j)$, we introduce a corresponding binary variable $x^\sigma(i, j) \in \{-1, 1\}$, and as usual let TRUE be denoted by 1 and FALSE by -1 . For clause $C_\tau(i, j)$ and variable $v^\sigma(i, j)$, define

$$s_\tau^\sigma(i, j) := \begin{cases} 1, & \text{if } v^\sigma(i, j) \notin J_\tau(i, j) \\ -1, & \text{if } v^\sigma(i, j) \in J_\tau(i, j) \end{cases} \quad (28)$$

Introducing new variables

$$y(i, j, S) := \prod_{\sigma \in S} x^\sigma(i, j)$$

for all $S \neq \emptyset, S \subseteq \{l, r, u, d\}, i \in \{0, 1, \dots, p-1\}, j \in \{0, 1, \dots, q-1\}$, and

$$\pi(i_1 \rightarrow i_2, j_1 \rightarrow j_2) := \left(\prod_{i=i_1}^{i_2} x^l(i, j_1) x^r(i, j_2) \right) \left(\prod_{j=j_1}^{j_2} x^u(i_1, j) x^d(i_2, j) \right),$$

we define a rank-one matrix of dimension $14pq$:

$$\bar{Y} := \begin{pmatrix} 1 \\ y(0, 0, \{l\}) \\ \vdots \\ y(p-1, q-1, \{l, r, u, d\}) \\ \pi(0 \rightarrow 1, 0) \\ \vdots \\ \pi(0 \rightarrow 1, q-1) \\ \pi(0 \rightarrow 2, 0) \\ \vdots \\ \pi(0 \rightarrow 2, q-1) \\ \vdots \\ \pi(0 \rightarrow p-1, 0) \\ \vdots \\ \pi(0 \rightarrow p-1, q-1) \\ \pi(0 \rightarrow p-1, 0 \rightarrow 1) \\ \vdots \\ \pi(0 \rightarrow p-1, 0 \rightarrow q-1) \end{pmatrix} \begin{pmatrix} 1 \\ y(0, 0, \{l\}) \\ \vdots \\ y(p-1, q-1, \{l, r, u, d\}) \\ \pi(0 \rightarrow 1, 0) \\ \vdots \\ \pi(0 \rightarrow 1, q-1) \\ \pi(0 \rightarrow 2, 0) \\ \vdots \\ \pi(0 \rightarrow 2, q-1) \\ \vdots \\ \pi(0 \rightarrow p-1, 0) \\ \vdots \\ \pi(0 \rightarrow p-1, q-1) \\ \pi(0 \rightarrow p-1, 0 \rightarrow 1) \\ \vdots \\ \pi(0 \rightarrow p-1, 0 \rightarrow q-1) \end{pmatrix}^T.$$

Using this matrix, the resulting SDP relaxation is:

$$\begin{aligned}
& \text{find } \bar{Y} \in \mathbb{S}^{14pq} \\
& \text{s.t.} \\
& \sum_{\sigma \in \{l,r,u,d\}} s_{\tau}^{\sigma}(i,j) \bar{Y}_{\emptyset, y(i,j, \{\sigma\})} - \sum_{S \subset \{l,r,u,d\}, |S|=2} \left(\prod_{\sigma \in S} s_{\tau}^{\sigma}(i,j) \right) \bar{Y}_{\emptyset, y(i,j,S)} \\
& + \sum_{S \subset \{l,r,u,d\}, |S|=3} \left(\prod_{\sigma \in S} s_{\tau}^{\sigma}(i,j) \right) \bar{Y}_{\emptyset, y(i,j,S)} - \left(\prod_{\sigma \in \{l,r,u,d\}} s_{\tau}^{\sigma}(i,j) \right) \bar{Y}_{\emptyset, y(i,j,S)} = 1, \\
& \quad \tau = 1, \dots, 8, \text{ and } i \in \{0, 1, \dots, p-1\}, j \in \{0, 1, \dots, q-1\} \\
& \bar{Y}_{y(0,j, \{l,r,u,d\}), y(1,j, \{l,r,u,d\})} = \bar{Y}_{\emptyset, \pi(0 \rightarrow 1, j)}, j = 0, \dots, q-1 \\
& \bar{Y}_{\pi(0 \rightarrow i, j), y(i+1, j, \{l,r,u,d\})} = \bar{Y}_{\emptyset, \pi(0 \rightarrow i+1, j)}, \text{ for } i = 1, \dots, p-2, j = 0, \dots, q-1 \\
& \bar{Y}_{\pi(0 \rightarrow p-1, 0), \pi(0 \rightarrow p-1, 1)} = \bar{Y}_{\emptyset, \pi(0 \rightarrow p-1, 0 \rightarrow 1)} \\
& \bar{Y}_{\pi(0 \rightarrow p-1, 0 \rightarrow j), \pi(0 \rightarrow p-1, j+1)} = \bar{Y}_{\emptyset, \pi(0 \rightarrow p-1, 0 \rightarrow j+1)}, j = 1, \dots, q-3. \\
& \bar{Y}_{\emptyset, \pi(0 \rightarrow p-1, 0 \rightarrow q-2)} = \bar{Y}_{\emptyset, \pi(0 \rightarrow p-1, q-1)} \\
& \text{diag}(\bar{Y}) = e \\
& \bar{Y} \succeq 0.
\end{aligned} \tag{29}$$

The construction of the SDP relaxation (29) follows the paradigm of partial higher liftings introduced in Section 5. Since the matrix variable has dimension $14pq$, and since there are $23pq - 1$ linear equality constraints, the SDP problem is linearly-sized with respect to $2pq$, the number of boolean variables in the SAT instance. Furthermore, the structure of the SDP relaxation is directly related to the structure of the SAT instance. Note also that there are many more valid linear constraints that could be added to the SDP problem. Such constraints equate elements of the matrix variable that would be equal if the matrix were restricted to have rank equal to one. The motivation for our particular choice of additional constraints in (29) is that although they are relatively few, they are sufficient for proving the following characterization of unsatisfiability:

Theorem 17. [6] *The Tseitin instance is unsatisfiable if and only if the corresponding SDP problem (29) is infeasible.*

Finally, we point out that even though Theorem 17 as stated is specific to the instances defined here, the SDP formulation and the proof of exactness can in principle be extended to other graph-based instances whose satisfiability is determined using quantities akin to the $t(i, j)$ parameters. We refer the reader to [6] for more details.

Acknowledgments

The author thanks the anonymous referees whose comments helped improve this paper.

References

- [1] <http://www.satcompetition.org>.
- [2] 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>.
- [3] M.F. Anjos. On semidefinite programming relaxations for the satisfiability problem. *Math. Meth. Oper. Res.*, 60(3), 2004.
- [4] M.F. Anjos. Proofs of unsatisfiability via semidefinite programming. In *Operations Research Proceedings 2003*, pages 308–315, Springer, Berlin, 2004.
- [5] M.F. Anjos. An improved semidefinite programming relaxation for the satisfiability problem. *Math. Program.*, 102(3):589–608, 2005.
- [6] M.F. Anjos. An explicit semidefinite characterization of satisfiability for Tseitin instances. Technical Report 2005-02, Department of Management Sciences, University of Waterloo, May 2005. http://www.optimization-online.org/DB_HTML/2005/05/1139.html.
- [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] T. Asano and D.P. Williamson. Improved approximation algorithms for MAX SAT. *J. Algorithms*, 42(1):173–202, 2002.
- [11] 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.
- [12] 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.
- [13] E. Boros, P.L. Hammer, and X. Sun. Recognition of q -Horn formulae in linear time. *Discrete Appl. Math.*, 55(1):1–13, 1994.
- [14] 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.
- [15] 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.

- [16] S.A. Cook. The complexity of theorem-proving procedures. In *Proc. of 3rd Annual ACM Symp. on Theory of Computing*, pages 151–158, 1971.
- [17] P.I. Davies and N.J. Higham. Numerically stable generation of correlation matrices and their factors. *BIT*, 40(4):640–651, 2000.
- [18] E. de Klerk. *Aspects of Semidefinite Programming*, volume 65 of *Applied Optimization*. Kluwer Academic Publishers, Dordrecht, 2002.
- [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.M. Deza and M. Laurent. *Geometry of Cuts and Metrics*, volume 15 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 1997.
- [22] U. Feige and M. Goemans. Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT. In *Proceedings of the 3rd Israel Symposium on the Theory of Computing and Systems*, pages 182–189, 1995.
- [23] M.X. Goemans and D.P. Williamson. New $\frac{3}{4}$ -approximation algorithms for the maximum satisfiability problem. *SIAM J. Discrete Math.*, 7(4):656–666, 1994.
- [24] 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.
- [25] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.
- [26] D. Grigoriev, E.A. Hirsch, and D.V. Pasechnik. Complexity of semialgebraic proofs. *Moscow Math. J.*, 2(4):647–679, 2002.
- [27] D. Grigoriev, E.A. Hirsch, and D.V. Pasechnik. Exponential lower bound for static semi-algebraic proofs. In *Automata, Languages and Programming*, volume 2380 of *Lecture Notes in Comput. Sci.*, pages 257–268. Springer, Berlin, 2002.
- [28] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, second edition, 1993.
- [29] M. Kojima H. Waki, S. Kim and M. Muramatsu. Sums of squares and semidefinite programming relaxations for polynomial optimization problems with structured sparsity. Technical report, Tokyo Inst. Tech., October 2004.
- [30] E. Halperin and U. Zwick. Approximation algorithms for MAX 4-SAT and rounding procedures for semidefinite programs. *J. Algorithms*, 40(2):184–211, 2001.

- [31] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48:798–859, 2001.
- [32] C. Helmberg. http://www-user.tu-chemnitz.de/~helmberg/sdp_software.html.
- [33] C. Helmberg. <http://www-user.tu-chemnitz.de/~helmberg/semidef.html>.
- [34] N.J. Higham. Computing the nearest correlation matrix—a problem from finance. *IMA J. Numer. Anal.*, 22(3):329–343, 2002.
- [35] E.A. Hirsch. <http://logic.pdmi.ras.ru/~hirsch/benchmarks/hgen8.html>.
- [36] H.H. Hoos and T. Stützle. SATLIB: An online resource for research on SAT. In *SAT 2000*, pages 283–292. IOS Press, 2000. SATLIB is available online at www.satlib.org.
- [37] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1990.
- [38] R.G. Jeroslow. Representability in mixed integer programming. I. Characterization results. *Discrete Appl. Math.*, 17(3):223–243, 1987.
- [39] D.S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. System Sci.*, 9:256–278, 1974.
- [40] H. Karloff and U. Zwick. A 7/8-approximation algorithm for MAX 3SAT? In *Proc. of 38th FOCS*, pages 406–415, 1997.
- [41] J.B. Lasserre. Optimality conditions and LMI relaxations for 0-1 programs. Technical report, LAAS-CNRS, Toulouse, France, 2000.
- [42] J.B. Lasserre. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM J. Optim.*, 12(3):756–769 (electronic), 2002.
- [43] M. Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0-1 programming. *Math. Oper. Res.*, 28(3):470–496, 2003.
- [44] M. Laurent. Semidefinite relaxations for max-cut. In M. Grötschel, editor, *The Sharpest Cut: The Impact of Manfred Padberg and His Work*. SIAM, 2004.
- [45] M. Laurent and F. Rendl. Semidefinite programming and integer programming. In G. Nemhauser K. Aardal and R. Weismantel, editors, *Handbook on Discrete Optimization*. Elsevier, to appear.
- [46] M. Lewin, D. Livnat, and U. Zwick. Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems. In *Integer Programming and Combinatorial Optimization*, volume 2337 of *Lecture Notes in Comput. Sci.*, pages 67–82. Springer, Berlin, 2002.
- [47] L. Lovász. On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory*, 25(1):1–7, 1979.
- [48] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, 1(2):166–190, 1991.

- [49] S. Mahajan and H. Ramesh. Derandomizing approximation algorithms based on semidefinite programming. *SIAM J. Comput.*, 28(5):1641–1663 (electronic), 1999.
- [50] S. Matuura and T. Matsui. 0.935-approximation randomized algorithm for max 2sat and its derandomization. Technical report, University of Tokyo, Tokyo, Japan, 2001. Available at <http://citeseer.lcs.mit.edu/573727.html>.
- [51] R.D.C. Monteiro. First- and second-order methods for semidefinite programming. *Math. Program.*, 97(1-2, Ser. B):209–244, 2003.
- [52] P.A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Math. Program.*, 96(2, Ser. B):293–320, 2003.
- [53] M. Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana Univ. Math. J.*, 42(3):969–984, 1993.
- [54] M.V. Ramana. An exact duality theory for semidefinite programming and its complexity implications. *Math. Program.*, 77(2, Ser. B):129–162, 1997.
- [55] 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.
- [56] K.C. Toh, M.J. Todd, and R.H. Tütüncü. <http://www.math.nus.edu.sg/~mattohc/sdpt3.html>.
- [57] K.C. Toh, M.J. Todd, and R.H. Tütüncü. SDPT3—a MATLAB software package for semidefinite programming, version 1.3. *Optim. Methods Softw.*, 11/12(1-4):545–581, 1999.
- [58] 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.
- [59] H. van Maaren. On the use of second order derivatives for the satisfiability problem. In *Satisfiability Problem: Theory and Applications (Piscataway, NJ, 1996)*, volume 35 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 677–687. Amer. Math. Soc., Providence, RI, 1997.
- [60] H. van Maaren. Elliptic approximations of propositional formulae. *Discrete Appl. Math.*, 96/97:223–244, 1999.
- [61] H. van Maaren and L. van Norden. Sums of squares, satisfiability and maximum satisfiability. *JSAT Addendum*, July 5, 2005.
- [62] H. van Maaren and J.P. Warners. Solving satisfiability problems using elliptic approximations. A note on volumes and weights. *Ann. Math. Artif. Intell.*, 37(3):273–283, 2003.
- [63] J.P. Warners and H. van Maaren. A two-phase algorithm for solving a class of hard satisfiability problems. *Oper. Res. Lett.*, 23(3-5):81–88, 1998.

- [64] J.P. Warners and H. van Maaren. Recognition of tractable satisfiability problems through balanced polynomial representations. In *Proceedings of the 5th Twente Workshop on Graphs and Combinatorial Optimization (Enschede, 1997)*, volume 99, pages 229–244, 2000.
- [65] J.P. Warners and H. van Maaren. Solving satisfiability problems using elliptic approximations—effective branching rules. *Discrete Appl. Math.*, 107(1-3):241–259, 2000.
- [66] H.P. Williams. Logical problems and integer programming. *Bull. Inst. Math. Appl.*, 13(1):18–20, 1977.
- [67] H.P. Williams. Logic applied to integer programming and integer programming applied to logic. *Eur. J. Oper. Res.*, 81(3):605–616, 1995.
- [68] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming*. Kluwer Academic Publishers, Boston, MA, 2000.
- [69] M. Yannakakis. On the approximation of maximum satisfiability. *J. Algorithms*, 17(3):475–502, 1994.
- [70] U. Zwick. Computer assisted proof of optimal approximability results. In *Proceedings of 13th SODA*, pages 496–505, 2002.