Les Cahiers du GERAD ISSN: 0711–2440

Semidefinite Resolution and Exactness of Semidefinite Relaxations for Satisfiability

M.F. Anjos
M.V.C. Vieira

G–2011–35

June 2011

Semidefinite Resolution and Exactness of Semidefinite Relaxations for Satisfiability

Miguel F. Anjos

GERAD and Department of Mathematics and Industrial Engineering
École Polytechnique de Montréal
Montréal (Québec) Canada, H3C 3A7
anjos@stanfordalumni.org

Manuel V.C. Vieira

Departamento de Matemática, Faculdade de Ciências e Tecnologia & CMA
Universidade Nova de Lisboa
Lisboa, Portugal
mvcv@fct.unl.pt

June 2011

Les Cahiers du GERAD

G–2011–35

Abstract

This paper explores new connections between the satisfiability problem and semidefinite programming. We show how the process of resolution in satisfiability is equivalent to a linear transformation between the feasible sets of the relevant semidefinite programming problems. We call this transformation semidefinite programming resolution, and we demonstrate the potential of this novel concept by using it to obtain a direct proof of the exactness of the semidefinite formulation of satisfiability without applying Lasserre's general theory for semidefinite relaxations of 0/1 problems. In particular, our proof explicitly shows how the exactness of the semidefinite formulation for any satisfiability formula can be intrepreted as the implicit application of a finite sequence of resolution steps to verify whether the empty clause can be derived from the given formula.

Résumé

Nous explorons de nouvelles connexions entre le problème de satisfiabilité et la programmation semi-définie. Nous montrons comment le processus de résolution en satisfiabilité est équivalent à une transformation linéaire entre les ensembles admissibles des relaxations semi-définies pertinentes. Nous appelons cela la résolution semi-définie, et nous démontrons le potentiel de ce nouveau concept en l'utilisant pour obtenir une démonstration directe de l'exactitude d'une relaxation semi-définie sans avoir recours à la théorie générale de Lasserre pour l'exactitude de relaxations semi-définies pour problèmes booléens. En particulier, notre démonstration montre explicitement comment l'exactitude de la formulation semi-définie peut être interprétée comme l'application implicite d'une séquence finie d'étapes de résolution afin de vérifier si la clause vide peut être dérivée à partir de la formule donnée.

Acknowledgments: Research of the first author was partially supported by the Natural Sciences and Engineering Research Council of Canada. Research of the second author was partially supported by a bursary from the FCT-Portugal.

1 Introduction

The satisfiability (SAT) problem is a central problem in mathematical logic with numerous practical applications [12]. An instance of SAT is specified by providing a set of boolean variables and a propositional formula whose literals are (some or all of) the given boolean variables and their negations. The SAT problem consists of determining whether or not it is possible to satisfy the formula by at least one assignment of the values true/false to the boolean variables. It is well known that SAT is in general NP-complete, although several important special cases can be solved in polynomial time.

Semidefinite optimization, frequently referred to as semidefinite programming (SDP), is the problem of optimizing a linear function of a matrix variable subject to linear constraints on its elements and the additional constraint that the matrix be positive semidefinite. This includes linear programming (LP) problems as a special case, namely when all the matrices involved are diagonal. Numerous algorithms, including polynomial-time interior-point methods, have been proposed to solve SDP problems, and several excellent solvers are available. We refer the reader to the handbooks [6, 34] for a thorough coverage of SDP theory and algorithms, the software available, and the application areas to which SDP researchers have made significant contributions.

We are interested in the application of SDP to the SAT problem, and in particular in gaining a better understanding of exact SDP relaxations for SAT. An exact SDP relaxation is one that characterizes precisely when the corresponding SAT instance is unsatisfiable, in the sense that the SDP relaxation is infeasible if and only if the SAT instance is unsatisfiable. In [16, 17], de Klerk, van Maaren, and Warners introduced the Gap relaxation and showed that it is exact for some well-known classes of SAT problems. Subsequent papers by Anjos [1, 3, 4, 5] and van Maaren et al. [31, 32] proposed several SDP relaxations for different versions of SAT.

Although most of the exactness results in the above papers refer to specifically structured SAT formulas, it is always possible to write an exact SDP relaxation for a SAT instance, even for instances of SAT for which no particular structure is assumed. This is done by formulating the SAT instance as a 0/1 optimization problem, and then constructing the corresponding Lasserre SDP relaxation as presented in [23]. Lasserre's theory proves that this SDP relaxation is always exact. However, this line of argument yields limited theoretical or practical benefits. Theoretically, this gives no understanding of the deeper connections between SAT and SDP. Practically, since the size of the exact SDP relaxation is exponential in the number of boolean variables in the instance, it is impractical for all but very small instances of SAT.

Our objective in this paper is to improve the situation from the theoretical perspective by exhibiting direct connections between SAT and SDP. The main contribution of this paper is the explicit demonstration that the process of resolution in SAT is equivalent to a linear transformation between the feasible sets of SDP relaxations. (The precise statement is given in Theorem 5.1 below.) We also show how this connection between resolution and SDP, called *SDP resolution*, allows us to write a direct proof of the exactness of Lasserre's SDP relaxation without recourse to Lasserre's general theory. Instead, our proof of the exactness of the (exponentially-sized) SDP relaxation makes clear how the relaxation implicitly deduces whether the empty clause can be derived by a finite sequence of resolution steps starting from the SAT formula. Therefore, our proof makes it immediately clear *why* the SDP relaxation will be infeasible precisely when such a sequence of steps exists.

This paper is structured as follows. In Section 2 we introduce some notation and present an overview of the related work in the literature. In Section 3 we show how to construct the exact SDP relaxation, and introduce permutation matrices that will help manipulate the SDP matrix variables. In Section 4, we define the concept of SDP resolution and prove its validity. Section 5 presents the proof that SDP resolution can be expressed as a linear transformation between the feasible sets of two SDP problems. In Section 6, we apply SDP resolution to obtain a direct proof of the exactness of the SDP relaxation. Finally, Section 7 summarizes some open questions and current research directions.

2 Background and Literature Review

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

One basic technique in SAT is the so-called resolution rule. The resolution rule is an execution of the following binary operation on clauses:

$$\frac{l \vee C_1 \qquad \bar{l} \vee C_2}{C_1 \vee C_2}$$

If clause D_1 contains a literal l and D_2 contains its dual $\bar{l}$ then $\mathrm{Res}(D_1, D_2)$ is defined by eliminating l, $\bar{l}$ from D_1, D_2 respectively, and conjoining the remaining literals into a single clause. If there is more than one such pair of dual literals then the resulting clause is a tautology and does not give us any constraint on the putative valuation. For that reason we will assume that all clauses under consideration are non-tautological, i.e., they do not contain a complementary pair of literals. Let us define $\mathrm{Res}(F)$ as the set of all clause that can be obtained from F in a finite number of steps by resolution. In other words, $F \subseteq \mathrm{Res}(F)$ and if resolution can be applied to clauses $D_1, D_2 \in \mathrm{Res}(F)$, then $\mathrm{Res}(D_1, D_2) \in \mathrm{Res}(F)$.

We will use the following small example to explain the ideas presented in this paper.

Example 2.1. Let $(x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_3 \vee x_4)$ be a CNF formula. Applying resolution rule we obtain a new clause: $(x_2 \vee x_3 \vee x_4)$. Since no other resolution steps are possible, we have that

$$\mathrm{Res}(F) = \{x_1 \vee x_2,\ \bar{x}_1 \vee x_3 \vee x_4,\ x_2 \vee x_3 \vee x_4\}.$$

We define the empty clause as a clause without literals and denote it $\emptyset$. This clause is not satisfiable. The empty clause can be obtained by resolution if $\mathrm{Res}(F)$ contains the unit clauses x and $\bar{x}$ for some variable x. The following well-known theorem states that the presence of $\emptyset$ in $\mathrm{Res}(F)$ precisely characterizes the unsatisfiability of F.

Theorem 2.1. *[27, Completeness Theorem for Resolution] Let F be a CNF formula. Then F is satisfiable if and only if $\emptyset \notin \mathrm{Res}(F)$.*

The fact that a SAT instance can be expressed as the feasibility of an optimization problem is known at least since the pioneering work of Williams [33] and Blair et al. [13]. The first optimization-based approaches to SAT focused mostly on formulating SAT and maximum satisfiability (MAX-SAT) as 0/1 integer linear programming problems whose linear programming relaxations can then be solved efficiently. For certain classes of instances, including Horn formulas and their generalizations, the exactness of the linear programming relaxation has been established, see e.g. [14]). The book of Chandru and Hooker [15] provides an excellent coverage of results at the interface of logical inference and optimization.

Over the last decade, much research at this interface has been concerned with applying SDP techniques to SAT and MAX-SAT problems. A variety of software packages for solving SDP problems are now available, see e.g. [6]. In particular, SDP is the basis for the best known approximation guarantees for MAX-k-SAT problems. (The notation k-SAT refers to the instances of SAT for which all the clauses have length at most k.) Given a set of propositional clauses all of length at most k in conjunctive normal form, the MAX-k-SAT problem consists of determining the largest number of clauses that can be satisfied simultaneously by any given truth assignment. The seminal paper of Goemans and Williamson [19] proposed and analyzed an approximation algorithm using SDP for the MAX-2-SAT problem. The approximation algorithm of Karloff and Zwick for MAX-3-SAT [22] was shown to be optimal (unless $P = NP$), and further extensions have been proposed by Zwick [35] and Halperin and Zwick [21].

As we already mentioned above, this paper is concerned with exact SDP relaxations. The Gap relaxation of de Klerk, van Maaren, and Warners [17, 16] is exact for some well-known classes of SAT problems, such as mutilated chessboard instances, pigeonhole instances, and 2-SAT instances. (Note that unlike MAX-2-SAT, 2-SAT is solvable in polynomial-time [10].) Further results in [4, 5] provide characterizations of unsatisfiability for Tseitin instances of SAT that are known to be hard for many proof systems. In particular, Tseitin [29] showed that proving unsatisfiability of such instances using regular resolution requires a superpolynomial number of resolution steps. Tseitin's result was extended to general resolution in [30] using so-called expander graphs, a more general class of graphs. Tseitin instances are now typically defined in terms of expander graphs (see e.g. [20]).

Other SDP relaxations for SAT have been proposed using the lifting paradigm for constructing SDP relaxations of discrete optimization problems. 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. 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 the Maximum-Cut (max-cut) problem proposed by Anjos and Wolkowicz [9], and its generalization independently proposed by Lasserre [23].

Lasserre's liftings are a recent contribution to a rich research literature on lift-and-project frameworks for 0/1 optimization [11, 26, 28]. The study of SDP relaxations for specific 0/1 problems dates back at least to Lovász's introduction of the so-called theta function as a bound for the stability number of a graph [25]. The Lasserre relaxations arise by viewing a 0/1 problem as a feasibility problem over a semi-algebraic set, and applying Putinar's positivstellensatz to obtain SDP relaxations, see e.g. [7]. Specifically, if one writes the SAT instance as a 0/1 optimization problem, Lasserre's theory [23] immediately yields an exact SDP relaxation (see Section 3). However, the resulting SDP problem has dimension exponential in n. This limitation motivates the study of tight SDP relaxations that have a much smaller matrix variable, as well as fewer linear constraints. Such SDP relaxations were proposed by Anjos [1, 3] and van Maaren et al. [31, 32]. All these constructions are defined explicitly rather than in terms of an iterative process increasingly tightening the relaxations. Their practical performance has been analyzed in [1, 2, 31, 32]. It is also interesting to study conditions that ensure exactness of SDP relaxations even when they are not of exponential size. Such conditions can be stated in terms of the rank of an optimal matrix for the SDP relaxation. For liftings of max-cut, the rank-1 case is obvious since the optimal solution of the SDP is then a vertex of the cut polytope (see e.g. [18]). For second liftings, a rank-2 guarantee of optimality was proved by Anjos and Wolkowicz [8], and this result was extended to the whole of the Lasserre hierarchy by Laurent [24]. For the SAT problem, rank-based conditions for the exactness of SDP relaxations were given in [1, 3].

3 Construction of the Exact SDP Relaxation

In this section, we introduce the exact SDP relaxation for SAT. We use the standard formulation for an SDP problem as follows:

$$\begin{array}{ll} \min & C \bullet X \\ \text{s.t.} & A_i \bullet X = b_i,\ i = 1,\dots,m \\ & X \succeq 0, \end{array}$$

where $C \bullet X := \operatorname{tr}(C\,X)$. The SAT problem is modeled as a feasibility problem using SDP. This means that we only consider the constraints $A_i \bullet X = b_i,\ i = 1,\dots,m$ and $X \succeq 0$ to model SAT, and we assume that we are maximizing an arbitrary linear function C, e.g. the zero function.

Let TRUE be denoted by 1 and FALSE be denoted by -1. Let $C = \bigvee_{i\in I} s_i x_i$ be a clause where the x_i are propositional variables and the s_i indicate whether x_i is negated or not, i.e., for $i \in I$, we let

$$s_i = \begin{cases} 1, & \text{if } x_i \text{ is not negated} \\ -1, & \text{if } x_i \text{ is negated} \end{cases}$$

The clause C is satisfied by $x_i = \pm 1$, $i \in I$ if and only if

$$\prod_{i\in I}(1 - s_i x_i) = 0.$$

Expanding this product we have

$$1 + \sum_{J\subseteq I}(-1)^{|J|}\prod_{i\in J} s_i x_i = 0.$$

Setting $y_J := \prod_{i\in J} x_i$, $y_\emptyset = 1$, $s_J := \prod_{i\in J} s_i$ and $s_\emptyset = 1$ we obtain

$$\sum_{J\subseteq I}(-1)^{|J|} s_J y_J = 0.$$

Given $V := \{1,\ldots,n\}$ let $\mathcal{P}(V)$ denote the collection of all subsets of V. The components of a vector $y \in \mathbb{R}^{\mathcal{P}(V)}$ are denoted as y_I for $I \in \mathcal{P}(V)$; we also set $y_i = y_{\{i\}}$. Given $y \in \mathbb{R}^{\mathcal{P}(V)}$, we consider the set of all matrices $Y \succeq 0$ satisfying

$$Y = (y_{I\Delta J})_{I,J\subseteq V},$$

where Δ denotes the symmetric difference of the sets I and J. The rows and columns of Y are indexed by the sets $I, J \subseteq V$. Note that for Y as defined we have:

$$Y_{(I\Delta P),(P\Delta J)} = Y_{I,J}, \quad \emptyset \subsetneq P \subseteq V.$$

So, we define the following set which encloses all these constraints:

$$\Omega_n = \{Y \succeq 0 | Y_{(I\Delta P),(P\Delta J)} = Y_{I,J},\ \forall \emptyset \subsetneq P \subseteq V \text{ and } y_\emptyset = 1\}.$$

Lasserre proved that an exact relaxation can be constructed using the set Ω_n. This exact formulation is presented in the following definition.

Definition 3.1. Let F be a CNF formula consisting of m clauses and I_i be the index set of variables of each clause $i = 1,\ldots,m$. We define SDP(F) as

$$\begin{aligned} &\sum_{J\subseteq I_i}(-1)^{|J|} s^i_J y_J = 0, \quad i = 1,\ldots,m \\ &Y \in \Omega_n. \end{aligned} \tag{1}$$

Lasserre's Theorem [23, Theorem 3.2] implies that SDP(F) is feasible if and only if F is satisfiable.

We illustrate the construction for our small example.

Example 3.1. The SDP formulation for our example is

$$\begin{aligned} &y_\emptyset - y_1 - y_2 + y_{12} = 0 \\ &y_\emptyset + y_1 - y_3 - y_4 - y_{13} - y_{14} + y_{34} + y_{134} = 0 \\ &Y \in \Omega_4, \end{aligned}$$

where the constraint $Y \in \Omega_4$ implies that the matrix Y is positive semidefinite and has the form:

$$\begin{pmatrix}
y_\emptyset & y_1 & y_2 & y_3 & y_4 & y_{12} & \cdots & y_{34} & y_{123} & \cdots & y_{234} & y_{1234} \\
y_1 & y_\emptyset & y_{12} & y_{13} & y_{14} & y_2 & \cdots & y_{134} & y_{23} & \cdots & y_{1234} & y_{234} \\
y_2 & y_{12} & y_\emptyset & y_{23} & y_{24} & y_1 & \cdots & y_{234} & y_{13} & \cdots & y_{34} & y_{134} \\
y_3 & y_{13} & y_{23} & y_\emptyset & y_{34} & y_{123} & \cdots & y_4 & y_{12} & \cdots & y_{24} & y_{124} \\
y_4 & y_{14} & y_{24} & y_{34} & y_\emptyset & y_{124} & \cdots & y_3 & y_{1234} & \cdots & y_{23} & y_{123} \\
y_{12} & y_2 & y_1 & y_{123} & y_{124} & y_\emptyset & \cdots & y_{1234} & y_3 & \cdots & y_{134} & y_{34} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
y_{34} & y_{134} & y_{234} & y_4 & y_3 & y_{1234} & \cdots & y_\emptyset & y_{124} & \cdots & y_2 & y_{12} \\
y_{123} & y_{23} & y_{13} & y_{12} & y_{1234} & y_3 & \cdots & y_{124} & y_\emptyset & \cdots & y_{14} & y_4 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
y_{234} & y_{1234} & y_{34} & y_{24} & y_{23} & y_{134} & \cdots & y_2 & y_{14} & \cdots & y_\emptyset & y_1 \\
y_{1234} & y_{234} & y_{134} & y_{124} & y_{123} & y_{34} & \cdots & y_{12} & y_4 & \cdots & y_1 & y_\emptyset
\end{pmatrix}$$

Note that the first row (column) of Y is just the vector y. Any other row (column) can be obtained by permutation of the first one. Moreover, any row (column) can be obtained by a permutation of any other row (column). We introduce particular matrices that will help in making use of such permutations.

Let P^S be a matrix indexed by all the subsets of V and $S \subseteq V$, where the element of row I and column J is defined in the following way:

$$P^S_{IJ} = \begin{cases} 1 & \text{if } I \Delta J = S \\ 0 & \text{otherwise.} \end{cases}$$

The matrix P^S is a permutation matrix such that $Y_{\cdot,S} = P^S y$, as we will prove below. Moreover we define the matrix P^{RS} with $S, R \subseteq V$ as

$$P^{RS}_{IJ} = \begin{cases} 1 & \text{if } I \Delta J = R \Delta S \\ 0 & \text{otherwise.} \end{cases}$$

Note that $P^S = P^{\emptyset S}$ and P^{RS} is symmetric because $I \Delta J = J \Delta I$.

Lemma 3.1. *The matrix P^{RS} is a permutation matrix such that*

$$P^{RS} Y_{\cdot,R} = Y_{\cdot,R\Delta S}.$$

Proof. If we fix row I then the entry (I, J) is 1 for J such that $I \Delta J = R \Delta S$. This implies that $I \Delta I \Delta J = I \Delta R \Delta S$, obtaining $J = I \Delta R \Delta S$. Therefore, for row I the entry equal to 1 has column indexed by $I \Delta R \Delta S$.

Let $\left(P^{RS} Y_{\cdot,R}\right)_I$ be the I component of the vector $P^{RS} Y_{\cdot,R}$. Thus,

$$\left(P^{RS} Y_{\cdot,R}\right)_I = y_{I\Delta R\Delta S} = Y_{I,R\Delta S},$$

but this is exactly the row I of the vector $Y_{\cdot,R\Delta S}$ □

Using the definition of P^{RS}, we can define the permutation matrix that transforms the vector of column R into column S as $P^{R(R\Delta S)}$

Lemma 3.2. *We have $P^S P^R = P^{S\Delta R}$ for $S, R \subseteq V$. It follows that $P^R Y_{\cdot,S} = Y_{\cdot,R\Delta S}$.*

Proof. The element (I, J) of the product $P^R P^S$ is given by

$$[P^S P^R]_{IJ} = \left[\sum_{L \subseteq V} P^S_{IL} P^R_{LJ}\right]$$

which is equal to 1 if and only if $I \Delta L = S$ and $L \Delta J = R$ for some $L \subseteq V$. But this is equivalent to $P^S_{IL} = 1$ and $P^R_{LJ} = 1$, thus implying that $P^{R\Delta S}_{IJ} = 1$ because $(I\Delta L)\Delta(L\Delta J) = R\Delta S$, i.e., $I \Delta J = R \Delta S$. Since $P^R P^S$ and $P^{R\Delta S}$ are both permutation matrices, first part of the result follows. Since $P^S y = Y_{\cdot,S}$, we successively have $P^R Y_{\cdot,S} = P^R P^S y = P^{R\Delta S} y = Y_{\cdot,R\Delta S}$. □

We will make use of these permutation matrices in Section 5 below.

4 SDP Resolution

In this section, we introduce the concept of SDP resolution. This technique expresses the resolution rule in the language of SDP.

Let us assume that we can apply the resolution rule to clauses C_1 and C_2, i.e., without loss of generality, we assume that there exists $x_1 \in C_1$ such that $\bar{x}_1 \in C_2$. In terms of SDP(F) this means that we can choose a common variable indexed by a singleton, say y_1, on two different constraints of SDP(F), say $i = 1, 2$, such that $s^1_1 = 1$ and $s^2_1 = -1$. Then we can define a new SDP problem in the following way.

(i) Define new constraints:

$$s^1_{J_1} s^2_{J_2} y_{J_1 \Delta J_2} = y_\emptyset + s^1_1 s^1_{J_1} y_{\{1\}\Delta J_1} + s^2_1 s^2_{J_2} y_{\{1\}\Delta J_2}, \tag{2}$$

for all $J_1 \subseteq I_1 \setminus \{1\}$ and $J_2 \subseteq I_2 \setminus \{1\}$ and $J_1, J_2 \neq \emptyset$;

(ii) Adding all the constraints in (2) indexed by nonempty sets J_1 and J_2 such that $|J_1 \Delta J_2|$ is even and subtracting the constraints such that $|J_1 \Delta J_2|$ is odd, we obtain the single constraint:

$$\begin{aligned} &\sum_{(J_1,J_2)\in T_e} s^1_{J_1} s^2_{J_2} y_{J_1\Delta J_2} - \sum_{(J_1,J_2)\in T_o} s^1_{J_1} s^2_{J_2} y_{J_1\Delta J_2} = \\ &\sum_{(J_1,J_2)\in T_e} \left(y_\emptyset + s^1_{J_1} y_{\{1\}\Delta J_1} - s^2_{J_2} y_{\{1\}\Delta J_2}\right) - \sum_{(J_1,J_2)\in T_o} \left(y_\emptyset + s^1_{J_1} y_{\{1\}\Delta J_1} - s^2_{J_2} y_{\{1\}\Delta J_2}\right), \end{aligned} \tag{3}$$

where

$$T_e = \{(J_1, J_2) | J_1 \subseteq I_1 \setminus \{1\},\ J_2 \subseteq I_2 \setminus \{1\} \text{ and } |J_1 \Delta J_2| \text{ is even }\}$$

and

$$T_o = \{(J_1, J_2) | J_1 \subseteq I_1 \setminus \{1\},\ J_2 \subseteq I_2 \setminus \{1\} \text{ and } |J_1 \Delta J_2| \text{ is odd }\}.$$

(iii) Combining constraint (3) with constraints $i = 1, 2$ of SDP(F) results in a new constraint:

$$\sum_{J\subseteq I_1\cup I_2\setminus\{1\}} (-1)^{|J|} \bar{s}_J y_J = 0,$$

where $\bar{s}_J$ has the same meaning as s^i_J.

Thus we obtain the following SDP problem:

$$\begin{aligned} &\sum_{J\subseteq I_i} (-1)^{|J|} s^i_J y_J = 0, \quad i = 1, \ldots, m \\ &\sum_{J\subseteq I_1\cup I_2\setminus\{1\}} (-1)^{|J|} \bar{s}_J y_J = 0 \\ &Y \in \Omega_n. \end{aligned} \tag{4}$$

We call this process *SDP resolution*, and the resulting SDP problem (4) is the *SDP resolvent*. To illustrate this, we use again our example.

Example 4.1. Since $s^1_1 = 1$ and $s^2_1 = -1$, we can apply SDP resolution. We define new constraints according to step (i) of SDP resolution:

$$\begin{aligned} s^1_{\{2\}} s^2_{\{3\}} y_{\{2\}\Delta\{3\}} &= y_\emptyset + s^1_{\{2\}} y_{\{1\}\Delta\{2\}} - s^2_{\{3\}} y_{\{1\}\Delta\{3\}} \\ s^1_{\{2\}} s^2_{\{4\}} y_{\{2\}\Delta\{4\}} &= y_\emptyset + s^1_{\{2\}} y_{\{1\}\Delta\{2\}} - s^2_{\{4\}} y_{\{1\}\Delta\{4\}} \\ s^1_{\{2\}} s^2_{\{34\}} y_{\{2\}\Delta\{34\}} &= y_\emptyset + s^1_{\{2\}} y_{\{1\}\Delta\{2\}} - s^2_{\{34\}} y_{\{1\}\Delta\{34\}}. \end{aligned}$$

Simplifying:

$$\begin{aligned} y_{23} &= y_\emptyset + y_{12} - y_{13} \\ y_{24} &= y_\emptyset + y_{12} - y_{14} \\ y_{234} &= y_\emptyset + y_{12} - y_{134}. \end{aligned}$$

Adding the first and second constraint and subtracting the third one gives the result of step (ii):

$$y_{23} + y_{24} - y_{234} = y_\emptyset + y_{12} - y_{13} - y_{14} + y_{34} + y_{134}..$$

Combining this constraint with the two equality constraints in Example 3.1, we get

$$(y_\emptyset - y_1 - y_2 + y_{12}) + (y_\emptyset + y_1 - y_3 - y_4 - y_{13} - y_{14} + y_{34} + y_{134})$$
$$+ (y_{23} + y_{24} - y_{234}) = y_\emptyset + y_{12} - y_{13} - y_{14} + y_{134},$$

which is equivalent to

$$y_\emptyset - y_2 - y_3 - y_4 + y_{23} + y_{24} + y_{34} - y_{234} = 0.$$

We have eliminated every variable whose index set contains the element 1. This new constraint represents the new clause obtained by resolution in Example 2.1, and thus this outcome is equivalent to the resolution process that eliminates variable x_1. The SDP resolvent is

$$\begin{aligned}
&y_\emptyset - y_1 - y_2 + y_{12} = 0\\
&y_\emptyset + y_1 - y_3 - y_4 - y_{13} - y_{14} + y_{34} + y_{134} = 0\\
&y_\emptyset - y_2 - y_3 - y_4 + y_{23} + y_{24} + y_{34} - y_{234} = 0\\
&Y \in \Omega_4,
\end{aligned}$$

and the matrix Y is as defined in Example 2.1.

The next result proves the validity of the process in general.

Proposition 4.1. *The constraints*

$$\sum_{J\subseteq I_i} (-1)^{|J|} s^i_J y_J = 0, \quad i = 1, 2$$

and (3) *imply the constraint*

$$\sum_{J\subseteq I_1\cup I_2\setminus\{1\}} (-1)^{|J|} \bar{s}_J y_J = 0.$$

Proof. For simplicity of notation in this proof, we use the notation $y_{IJ} = y_{I\Delta J}$, $I_1' = I_1\setminus\{1\}$ and $I_2' = I_2\setminus\{1\}$. Let us assume that $P = I_1' \cap I_2'$ and $|P| = p$. Adding constraints $i = 1, 2$ we get

$$2y_\emptyset + \sum_{J\in Q_1} (-1)^{|J|} s^1_J y_J + \sum_{J\in Q_2} (-1)^{|J|} s^2_J y_J = 0, \tag{5}$$

with $Q_i = \{J \subseteq I_i | J \neq \{1\} \text{ and } J \neq \emptyset\}$ for $i = 1, 2$. As we keep the element $y_\emptyset$ outside the compacted sum, the summations over the sets do not include the empty set. We now expand the two summations in (5) according to the following partition of Q_1:

$$Q_1 = \{J \subseteq I_1 | 1 \in J\} \cup \{J \subseteq I_1 | 1 \notin J\}.$$

Now,

$$\{J \subseteq I_1 | 1 \in J\} = \{J \subseteq I_1 | 1 \in J,\ J \nsubseteq P\} \cup \{J \subseteq I_1 | 1 \in J,\ J \subseteq P\}$$

and similarly,

$$\{J \subseteq I_1 | 1 \notin J\} = \{J \subseteq I_1 | 1 \notin J,\ J \nsubseteq P\} \cup \{J \subseteq I_1 | 1 \notin J,\ J \subseteq P\}.$$

Thus, since $s^1_1 = 1$,

$$\begin{aligned}
&\sum_{J\in Q_1} (-1)^{|J|} s^1_J y_J\\
&= \sum_{J\subseteq I_1'} (-1)^{|J|+1} s^1_1 s^1_J y_{\{1\}J} + \sum_{J\subseteq I_1'} (-1)^{|J|} s^1_J y_J\\
&= \sum_{\substack{J\subseteq I_1'\\ J\nsubseteq P}} (-1)^{|J|+1} s^1_J y_{\{1\}J} + \sum_{J\subseteq P} (-1)^{|J|+1} s^1_J y_{\{1\}J}\\
&\quad + \sum_{\substack{J\subseteq I_1'\\ J\nsubseteq P}} (-1)^{|J|} s^1_J y_J + \sum_{J\subseteq P} (-1)^{|J|} s^1_J y_J.
\end{aligned}$$

Similarly, since $s_1^2 = -1$, we obtain

$$\begin{aligned}
&\sum_{J\in Q_2}(-1)^{|J|}s_J^2 y_J = \\
&= \sum_{J\subseteq I_2'}(-1)^{|J|+1}s_1^2 s_J^2 y_{\{1\}J} + \sum_{J\subseteq I_2'}(-1)^{|J|}s_J^2 y_J \\
&= -\sum_{\substack{J\subseteq I_2'\\ J\not\subseteq P}}(-1)^{|J|+1}s_J^2 y_{\{1\}J} - \sum_{J\subseteq P}(-1)^{|J|+1}s_J^2 y_{\{1\}J} \\
&+ \sum_{\substack{J\subseteq I_1'\\ J\not\subseteq P}}(-1)^{|J|}s_J^2 y_J + \sum_{J\subseteq P}(-1)^{|J|}s_J^2 y_J.
\end{aligned}$$

Summing both terms and splitting the summation over the odd sets and even sets we obtain

$$\begin{aligned}
&\sum_{J\in Q_1}(-1)^{|J|}s_J^1 y_J + \sum_{J\in Q_2}(-1)^{|J|}s_J^2 y_J = \\
&= \sum_{\substack{J\subseteq I_1'\\ J\not\subseteq P}}(-1)^{|J|+1}s_J^1 y_{\{1\}J} + \sum_{\substack{J\subseteq I_1'\\ J\not\subseteq P}}(-1)^{|J|}s_J^1 y_J + 2\sum_{J\subseteq P}(-1)^{|J|}s_J^1 y_J \\
&\quad - \sum_{\substack{J\subseteq I_2'\\ J\not\subseteq P}}(-1)^{|J|+1}s_J^2 y_{\{1\}J} + \sum_{\substack{J\subseteq I_2'\\ J\not\subseteq P}}(-1)^{|J|}s_J^2 y_J \qquad (s_J^1 = s_J^2 \text{ for all } J\subseteq P) \\
&= \underbrace{-\sum_{\substack{J\subseteq I_1'\\ J\not\subseteq P\\ |J|_e}} s_J^1 y_{\{1\}J} + \sum_{\substack{J\subseteq I_1'\\ J\not\subseteq P\\ |J|_o}} s_J^1 y_{\{1\}J} + \sum_{\substack{J\subseteq I_2'\\ J\not\subseteq P\\ |J|_e}} s_J^2 y_{\{1\}J} - \sum_{\substack{J\subseteq I_2'\\ J\not\subseteq P\\ |J|_o}} s_J^2 y_{\{1\}J}}_{(A_2)} \\
&\quad \underbrace{+ \sum_{\substack{J\subseteq I_1'\\ J\not\subseteq P\\ |J|_e}} s_J^1 y_J - \sum_{\substack{J\subseteq I_1'\\ J\not\subseteq P\\ |J|_o}} s_J^1 y_J + 2\Big(\sum_{\substack{J\subseteq P\\ |J|_e}} s_J^1 y_J - \sum_{\substack{J\subseteq P\\ |J|_o}} s_J^1 y_J\Big) + \sum_{\substack{J\subseteq I_2'\\ J\not\subseteq P\\ |J|_e}} s_J^2 y_J - \sum_{\substack{J\subseteq I_2'\\ J\not\subseteq P\\ |J|_o}} s_J^2 y_J}_{(A_3)},
\end{aligned}$$

where $|J|_e$, (respectively $|J|_o$) means that J has cardinality even (respectively odd). Therefore, we write equation (5) as

$$2y_\emptyset + (A_2) + (A_3) = 0.$$

Now, we deal with the constraint (3). Its terms are treated separately. We start with the terms involving $y_\emptyset$,

$$\sum_{(J_1,J_2)\in T_e} y_\emptyset - \sum_{(J_1,J_2)\in T_o} y_\emptyset = \underbrace{(2-2^p)y_\emptyset}_{(B_1)}.$$

To see this we have to count the number of elements of T_e and T_o. This is done in the Appendix.

Next we deal with the terms of the form $y_{\{1\}J}$ appearing in constraint (3). We start by partitioning the set T_e over the first coordinate:

$$\begin{aligned}
T_e =& \{(J_1,J_2)|J_1\subseteq I_1', J_2\subseteq I_2', |J_1\Delta J_2|_e\} \\
=& \{(J_1,J_2)|\ |J_1|_e, |J_2|_e\} \cup \{(J_1,J_2)|\ |J_1|_o, |J_2|_o\} \\
=& (\{(J_1,J_2)|J_1\not\subseteq P, |J_1|_e, |J_2|_e\} \cup \{(J_1,J_2)|J_1\subseteq P, |J_1|_e, |J_2|_e\}) \\
& \cup (\{(J_1,J_2)|J_1\not\subseteq P, |J_1|_o, |J_2|_o\} \cup \{(J_1,J_2)|J_1\subseteq P, |J_1|_o, |J_2|_o\})
\end{aligned}$$

Next we partition over the second coordinate:

$$\begin{aligned}T_e = &\{(J_1, J_2)|J_1 \not\subseteq P, J_2 \not\subseteq P, |J_1|_e, |J_2|_e\} \cup \{(J_1, J_2)|J_1 \not\subseteq P, J_2 \subseteq P, J_2|J_1|_e, |J_2|_e\} \\ &\cup (\{(J_1, J_2)|J_1 \subseteq P, J_2 \not\subseteq P, |J_1|_e, |J_2|_e\} \cup \{(J_1, J_2)|J_1 \subseteq P, J_2 \subseteq P, |J_1|_e, |J_2|_e\}) \\ &\cup (\{(J_1, J_2)|J_1 \not\subseteq P, J_2 \not\subseteq P, |J_1|_o, |J_2|_o\} \cup \{(J_1, J_2)|J_1 \not\subseteq P, J_2 \subseteq P, |J_1|_o, |J_2|_o\}) \\ &\cup (\{(J_1, J_2)|J_1 \subseteq P, J_2 \not\subseteq P, |J_1|_o, |J_2|_o\} \cup \{(J_1, J_2)|J_1 \subseteq P, J_2 \subseteq P, |J_1|_o, |J_2|_o\}) \,.\end{aligned}$$

Expanding the summation according to this partition gives

$$\begin{aligned}\sum_{(J_1,J_2)\in T_e} s^1_{J_1} y_{\{1\}J_1} = &\sum_{\substack{J_2\subseteq I'_2 \\ J_2\not\subseteq P \\ |J_2|_e}} \left(\sum_{\substack{J_1\subseteq I'_1 \\ J_1\not\subseteq P \\ |J_1|_e}} s^1_{J_1} y_{\{1\}J_1} + \sum_{\substack{J_1\subseteq P \\ |J_1|_e}} s^1_{J_1} y_{\{1\}J_1} \right) \\ &+ \sum_{\substack{J_2\subseteq P \\ |J_2|_e}} \left(\sum_{\substack{J_1\subseteq I'_1 \\ J_1\not\subseteq P \\ |J_1|_e}} s^1_{J_1} y_{\{1\}J_1} + \sum_{\substack{J_1\subseteq P \\ |J_1|_e}} s^1_{J_1} y_{\{1\}J_1} \right) \\ &+ \sum_{\substack{J_2\subseteq I'_2 \\ J_2\not\subseteq P \\ |J_2|_o}} \left(\sum_{\substack{J_1\subseteq I'_1 \\ J_1\not\subseteq P \\ |J_1|_o}} s^1_{J_1} y_{\{1\}J_1} + \sum_{\substack{J_1\subseteq P \\ |J_1|_o}} s^1_{J_1} y_{\{1\}J_1} \right) \\ &+ \sum_{\substack{J_2\subseteq P \\ |J_2|_o}} \left(\sum_{\substack{J_1\subseteq I'_1 \\ J_1\not\subseteq P \\ |J_1|_o}} s^1_{J_1} y_{\{1\}J_1} + \sum_{\substack{J_1\subseteq P \\ |J_1|_o}} s^1_{J_1} y_{\{1\}J_1} \right) .\end{aligned}$$

Now observe that the terms inside the parentheses do not depend on J_2. Therefore we only need to count the number of elements that appear in each of the summations over J_2, which is done in the Appendix. The result is

$$\begin{aligned}\sum_{(J_1,J_2)\in T_e} s^1_{J_1} y_{\{1\}J_1} = &\frac{2^{|I'_2|} - 2^p}{2} \left(\sum_{\substack{J_1\subseteq I'_1 \\ J_1\not\subseteq P \\ |J_1|_e}} s^1_{J_1} y_{\{1\}J_1} + \sum_{\substack{J_1\subseteq P \\ |J_1|_e}} s^1_{J_1} y_{\{1\}J_1} \right) \\ &+ (2^{p-1} - 1) \left(\sum_{\substack{J_1\subseteq I'_1 \\ J_1\not\subseteq P \\ |J_1|_e}} s^1_{J_1} y_{\{1\}J_1} + \sum_{\substack{J_1\subseteq P \\ |J_1|_e}} s^1_{J_1} y_{\{1\}J_1} \right)\end{aligned}$$

$$+\frac{2^{|I_2'|}-2^p}{2}\left(\sum_{\substack{J_1\subseteq I_1'\\ J_1\nsubseteq P\\ |J_1|_o}} s^1_{J_1}y_{\{1\}J_1}+\sum_{\substack{J_1\subseteq P\\ |J_1|_o}} s^1_{J_1}y_{\{1\}J_1}\right)$$

$$+2^{p-1}\left(\sum_{\substack{J_1\subseteq I_1'\\ J_1\nsubseteq P\\ |J_1|_o}} s^1_{J_1}y_{\{1\}J_1}+\sum_{\substack{J_1\subseteq P\\ |J_1|_o}} s^1_{J_1}y_{\{1\}J_1}\right).$$

Now we also partition T_o:

$$\begin{aligned}T_e=&\{(J_1,J_2)|J_1\subseteq I_1',J_2\subseteq I_2',|J_1\Delta J_2|_o\}\\ =&\{(J_1,J_2)|\ |J_1|_e,|J_2|_o\}\cup\{(J_1,J_2)|\ |J_1|_o,|J_2|_e\},\end{aligned}$$

and proceeding similarly we obtain

$$\begin{aligned}\sum_{(J_1,J_2)\in T_o} s^1_{J_1}y_{\{1\}J_1}=&\frac{2^{|I_2'|}-2^p}{2}\left(\sum_{\substack{J\subseteq I_1'\\ J_1\nsubseteq P\\ |J_1|_o}} s^1_{J_1}y_{\{1\}J_1}+\sum_{\substack{J_1\subseteq P\\ |J_1|_o}} s^1_{J_1}y_{\{1\}J_1}\right)\\ &+(2^{p-1}-1)\left(\sum_{\substack{J\subseteq I_1'\\ J_1\nsubseteq P\\ |J_1|_o}} s^1_{J_1}y_{\{1\}J_1}+\sum_{\substack{J_1\subseteq P\\ |J_1|_o}} s^1_{J_1}y_{\{1\}J_1}\right)\\ &+\frac{2^{|I_2'|}-2^p}{2}\left(\sum_{\substack{J\subseteq I_1'\\ J_1\nsubseteq P\\ |J_1|_e}} s^1_{J_1}y_{\{1\}J_1}+\sum_{\substack{J_1\subseteq P\\ |J_1|_e}} s^1_{J_1}y_{\{1\}J_1}\right)\\ &+2^{p-1}\left(\sum_{\substack{J\subseteq I_1'\\ J_1\nsubseteq P\\ |J_1|_e}} s^1_{J_1}y_{\{1\}J_1}+\sum_{\substack{J_1\subseteq P\\ |J_1|_e}} s^1_{J_1}y_{\{1\}J_1}\right).\end{aligned}$$

Therefore,

$$\begin{aligned}&\sum_{(J_1,J_2)\in T_e} s^1_{J_1}y_{\{1\}J_1}-\sum_{(J_1,J_2)\in T_o} s^1_{J_1}y_{\{1\}J_1}=\\ &\sum_{\substack{J_1\subseteq I_1'\\ J_1\nsubseteq P\\ |J_1|_o}} s^1_{J_1}y_{\{1\}J_1}-\sum_{\substack{J_1\subseteq I_1'\\ J_1\nsubseteq P\\ |J_1|_e}} s^1_{J_1}y_{\{1\}J_1}+\sum_{\substack{J_1\subseteq P\\ |J_1|_o}} s^1_{J_1}y_{\{1\}J_1}-\sum_{\substack{J_1\subseteq P\\ |J_1|_e}} s^1_{J_1}y_{\{1\}J_1}.\end{aligned}$$

Similarly to the above derivation, we obtain

$$\begin{aligned}
&\sum_{(J_1,J_2)\in T_e} s^1_{J_1} y_{\{1\}J_1} - \sum_{(J_1,J_2)\in T_o} s^1_{J_1} y_{\{1\}J_1} \\
&- \left(\sum_{(J_1,J_2)\in T_e} s^2_{J_2} y_{\{1\}J_2} - \sum_{(J_1,J_2)\in T_o} s^2_{J_2} y_{\{1\}J_2} \right) = \\
&= \sum_{\substack{J_1\subseteq I'_1\\ J_1\nsubseteq P\\ |J_1|_o}} s^1_{J_1} y_{\{1\}J_1} - \sum_{\substack{J_1\subseteq I'_1\\ J_1\nsubseteq P\\ |J_1|_e}} s^1_{J_1} y_{\{1\}J_1} + \sum_{\substack{J_1\subseteq P\\ |J_1|_o}} s^1_{J_1} y_{\{1\}J_1} - \sum_{\substack{J_1\subseteq P\\ |J_1|_e}} s^1_{J_1} y_{\{1\}J_1} \\
&- \sum_{\substack{J_2\subseteq I'_2\\ J_2\nsubseteq P\\ |J_2|_o}} s^2_{J_2} y_{\{1\}J_2} + \sum_{\substack{J_2\subseteq I'_2\\ J_2\nsubseteq P\\ |J_2|_e}} s^2_{J_2} y_{\{1\}J_2} - \sum_{\substack{J_2\subseteq P\\ |J_2|_o}} s^2_{J_2} y_{\{1\}J_2} + \sum_{\substack{J_2\subseteq P\\ |J_2|_e}} s^2_{J_2} y_{\{1\}J_2} \\
&= \underbrace{\sum_{\substack{J_1\subseteq I'_1\\ J_1\nsubseteq P\\ |J_1|_o}} s^1_{J_1} y_{\{1\}J_1} - \sum_{\substack{J_1\subseteq I'_1\\ J_1\nsubseteq P\\ |J_1|_e}} s^1_{J_1} y_{\{1\}J_1} - \sum_{\substack{J_2\subseteq I'_2\\ J_2\nsubseteq P\\ |J_2|_o}} s^1_{J_2} y_{\{1\}J_2} + \sum_{\substack{J_2\subseteq I'_2\\ J_2\nsubseteq P\\ |J_2|_e}} s^1_{J_2} y_{\{1\}J_2}}_{(B_2)} .
\end{aligned}$$

Using the appropriate partitions of T_e, we expand the summation with elements of the form $s^1_{J_1} s^2_{J_2} y_{J_1J_2}$:

$$\begin{aligned}
&\sum_{(J_1,J_2)\in T_e} s^1_{J_1} s^2_{J_2} y_{J_1J_2} = \sum_{\substack{J_1\subseteq I'_1, J_2\subseteq I'_2\\ |J_1\Delta J_2|_e}} s^1_{J_1} s^2_{J_2} y_{J_1J_2} \\
&= \sum_{\substack{J_1\subseteq I'_1, J_2\subseteq I'_2\\ J_1,J_2\nsubseteq P\\ |J_1\Delta J_2|_e}} s^1_{J_1} s^2_{J_2} y_{J_1J_2} + \sum_{\substack{J_1\subseteq P, J_2\subseteq I'_2\\ J_2\nsubseteq P\\ |J_1\Delta J_2|_e}} s^1_{J_1} s^2_{J_2} y_{J_1J_2} \\
&+ \sum_{\substack{J_1\subseteq I'_1, J_2\subseteq P\\ J_1\nsubseteq P\\ |J_1\Delta J_2|_e}} s^1_{J_1} s^2_{J_2} y_{J_1J_2} + \sum_{\substack{J_1,J_2\subseteq P\\ |J_1\Delta J_2|_e\\ J_1\neq J_2}} s^1_{J_1} s^2_{J_2} y_{J_1J_2}.
\end{aligned}$$

In every summation term there are repeated elements $y_{J_1J_2}$. It is enough to verify that for $J_1 \subseteq I'_1$, $J_2 \subseteq I'_2$, $J_1, J_2 \nsubseteq P$ we have that

$$J_1\Delta J_2 = (J_1\Delta Q)\Delta(J_2\Delta Q)$$

for all $Q \subseteq P$ and both $J_1\Delta Q$ and $J_2\Delta Q$ are nonempty. Recall that we did not index these constraints to the empty set. This means that the element $y_{J_1J_2}$ appears 2^p times (the number of elements of the power set of P.) So, if we make $J = J_1\Delta J_2$ we obtain that $J \subseteq I'_1 \cup I'_2$, $J \nsubseteq P$ and $|J|$ is even. We also must guarantee that J contains elements of both I'_1 and I'_2, which are not in P. Therefore, removing all the repetitions, we obtain

$$\sum_{\substack{J_1\subseteq I'_1, J_2\subseteq I'_2\\ J_1,J_2\nsubseteq P\\ |J_1\Delta J_2|_e}} s^1_{J_1} s^2_{J_2} y_{J_1J_2} = 2^p \sum_{\substack{J\subseteq I'_1\cup I'_2\\ J\cap(I'_i\setminus P)\neq\emptyset,\ i=1,2\\ J\nsubseteq P,\ |J|_e}} \bar{s}_J y_J,$$

where $\bar{s}_J = s^1_{J_1} s^2_{J_2}$. In the case $J_1 \subseteq P$ and $J_2 \subseteq I'_2$ we have $2^p - 1$ repetitions because we cannot consider the choice of $Q = J_1$, in which case we would have $J_1\Delta J_1 = \emptyset$. When $J_1, J_2 \subseteq P$ we cannot consider $Q = J_1$ or $Q = J_2$, thus we have $2^p - 2$ repetitions of $y_{J_1J_2}$. Thus,

$$\sum_{(J_1,J_2)\in T_e} s^1_{J_1}s^2_{J_2}y_{J_1J_2} =$$
$$=\underbrace{2^p \sum_{\substack{J\subseteq I_1'\cup I_2'\\ J\cap(I_i'\setminus P)\neq\emptyset,\ i=1,2\\ J\nsubseteq P,\ |J|_e}} \bar{s}_J y_J + (2^p-1)\Big(\sum_{\substack{J\subseteq I_2'\\ J\nsubseteq P\\ |J|_e}}\bar{s}_J y_J + \sum_{\substack{J\subseteq I_1'\\ J\nsubseteq P\\ |J|_e}}\bar{s}_J y_J\Big) + (2^p-2)\sum_{\substack{J\subseteq P\\ |J|_e}}\bar{s}_J y_J}_{(A_4)}.$$

Analogously,

$$\sum_{(J_1,J_2)\in T_o} s^1_{J_1}s^2_{J_2}y_{J_1J_2}$$
$$= \sum_{\substack{J_1\subseteq I_1', J_2\subseteq I_2'\\ J_1,J_2\nsubseteq P\\ |J_1\Delta J_2|_o}} s^1_{J_1}s^2_{J_2}y_{J_1J_2} + \sum_{\substack{J_1\subseteq P, J_2\subseteq I_2'\\ J_2\nsubseteq P\\ |J_1\Delta J_2|_o}} s^1_{J_1}s^2_{J_2}y_{J_1J_2}$$
$$+ \sum_{\substack{J_1\subseteq I_1', J_2\subseteq P\\ J_1\nsubseteq P\\ |J_1\Delta J_2|_o}} s^1_{J_1}s^2_{J_2}y_{J_1J_2} + \sum_{\substack{J_1,J_2\subseteq P\\ |J_1\Delta J_2|_o\\ J_1\neq J_2}} s^1_{J_1}s^2_{J_2}y_{J_1J_2}$$
$$=\underbrace{2^p \sum_{\substack{J\subseteq I_1'\cup I_2'\\ J\cap(I_i'\setminus P)\neq\emptyset,\ i=1,2\\ J\nsubseteq P,\ |J|_o}} \bar{s}_J y_J + (2^p-1)\Big(\sum_{\substack{J\subseteq I_2'\\ |J|_o}}\bar{s}_J y_J + \sum_{\substack{J\subseteq I_1'\\ |J|_o}}\bar{s}_J y_J\Big) + (2^p-2)\sum_{\substack{J\subseteq P\\ |J|_o}}\bar{s}_J y_J}_{(A_5)}$$

We have now to simplify the resulting equation. The term (B_1) is moved to the left-hand side of the equation and sums with the term $2y_\emptyset$ in equation (5), summing up to $2^p y_\emptyset$. All the terms in (A_2) will cancel with the terms of (B_2). Finally we add (A_3) to (A_4), and subtract (A_5), obtaining the following equation:

$$2^p y_\emptyset + 2^p \sum_{\substack{J\subseteq I_1'\cup I_2'\\ J\cap(I_i'\setminus P)\neq\emptyset,\ i=1,2\\ J\nsubseteq P}} \bar{s}_J y_J + 2^p \sum_{\substack{J\subseteq I_2'\\ J\nsubseteq P}} \bar{s}_J y_J + 2^p \sum_{\substack{J\subseteq I_1'\\ J\nsubseteq P}} \bar{s}_J y_J + 2^p \sum_{J\subseteq P} \bar{s}_J y_J = 0.$$

The result follows. □

5 Direct Representation of SDP Resolution

In this section we show that SDP resolution can be represented as a linear transformation between the feasible sets of two SDP problems.

Consider the linear transformation $\mathcal{A} : \mathbf{S}^{2^n} \to \mathbf{S}^{2^n}$ defined by

$$\mathcal{A}(Y) := Y + \sum_{S\subseteq V} P^S W P^S Y I_S,$$

where the matrix I_S is equal to 1 in position (S,S) and 0 elsewhere; the matrix W has only two nonzero rows, $W_{I_1,\cdot}$ and $W_{I_2,\cdot}$ which contain the coefficients of the constraints $i = 1,2$ in (1), written in a such way that the coefficients of y_{I_i} are -1. For instance, for $i = 1$,

$$\sum_{J\subseteq I_1} (-1)^{|J|} s^1_J y_J = (-1)^{|I_1|} s^1_{I_1} y_{I_1} + \sum_{J\subset I_1} (-1)^{|J|} s^1_J y_J = 0,$$

which is equivalent to

$$-y_{I_1} - \sum_{J\subset I_1} (-1)^{|I_1\setminus J|} s^1_{I_1\setminus J} y_J = 0.$$

In other words, the linear transformation $\mathcal{A}$ simply replaces every element y_{I_1}, y_{I_2} in matrix Y using the constraints $i = 1, 2$ of (1).

Example 5.1. Using Example 3.1, the matrix W is given by

$$
\begin{array}{rc}
 & \begin{array}{cccccccccccccccc} \emptyset & \{1\} & \{2\} & \{3\} & \{4\} & \{12\} & \{13\} & \{14\} & & & & \{34\} & & & \{134\} & \end{array} \\
\begin{array}{r} \\ \\ \text{row } \{12\} \longrightarrow \\ \\ \text{row } \{134\} \longrightarrow \\ \\ \\ \end{array} &
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-1 & 1 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-1 & -1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
$$

where the rows $\{12\}$ and $\{134\}$ contains the coefficients of the equations $y_\emptyset - y_1 - y_2 + y_{12} = 0$ and $y_\emptyset + y_1 - y_3 - y_4 - y_{13} - y_{14} + y_{34} + y_{134} = 0$, respectively.

Lemma 5.1. *In the matrix $P^S W P^S Y I_S$ the element of row $S\Delta I_1$ and column S is*

$$
-y_{I_1} - \sum_{J \subset I_1} (-1)^{|I_1 \setminus J|} s^1_{I_1 \setminus J} y_J, \tag{6}
$$

the element of row $S\Delta I_2$ and column S is

$$
-y_{I_2} - \sum_{J \subset I_2} (-1)^{|I_2 \setminus J|} s^2_{I_2 \setminus J} y_J, \tag{7}
$$

and all other elements are zero.

Proof. The product $Y I_S$ gives a zero matrix except for column S which contains the vector $P^S y$. Since $P^S P^S$ is the identity matrix, $P^S Y I_S$ just contains vector y in column S. Therefore, multiplying W by $P^S Y I_S$ results a matrix where column S and rows I_1 and I_2 have the elements (6) and (7), respectively. By Lemma 3.2, pre-multiplication of $W P_S Y I_S$ by P^S moves these elements to rows $S\Delta I_1$ and $S\Delta I_2$. $\square$

Corollary 5.1. *Each (L, J) element of the matrix*

$$
\sum_{S \subseteq V} P^S W P^S Y I_S
$$

such that $L\Delta J = I_1$ (respectively $L\Delta J = I_2$) contains the element (6) (respectively (7)). All other elements of the matrix are zero. Moreover the matrix is symmetric..

Proof. With $L = S\Delta I_1$ and $J = S$, $L\Delta J = S\Delta I_1 \Delta S = I_1$. Since this holds for every $S \subseteq V$, the result follows by Lemma 5.1. The matrix is symmetric because $L\Delta J = J\Delta L$. $\square$

We can easily verify that element (6) can only be found once in each column (row). The same holds for element (7). Thus, it is clear that we can rewrite the SDP problem (1) as

$$
\begin{aligned}
&\sum_{J \subseteq I_i} (-1)^{|J|} s^i_J y_J = 0, \qquad i = 3, \ldots, m \\
&\mathcal{A}(Y) \in \Omega_n.
\end{aligned} \tag{8}
$$

Next define the matrix W_r in a similar way to W. However W_r contains just one non-zero row: row $I'_1 \cup I'_2$ contains the coefficients of equation (3) such that the coefficient of $y_{I'_1 \cup I'_2}$ is -1. This means that we have to multiply both sides of equation (3) by the factor

$$
-\frac{1}{2^p} (-1)^{|I'_1 \cup I'_2|} \bar{s}_{I'_1 \cup I'_2}.
$$

Using W_r, define

$$\mathcal{A}_r(Y) = Y + \sum_{S\subseteq I} P^S W_r P^S Y I_S.$$

In a similar way, the feasibility problem $\mathcal{A}_r(Y) \in \Omega_n$ is equivalent to system defined by the equations (3) and $Y \in \Omega_n$.

We will show that we can rewrite the SDP problem (4) using the linear applications $\mathcal{A}$ and $\mathcal{A}_r$. Let us define $\mathcal{B} = \mathcal{A}_r \circ \mathcal{A}$. Then

$$\begin{aligned}
\mathcal{B}(Y) =& \mathcal{A}_r(\mathcal{A}(Y))\\
=& \mathcal{A}(Y) + \sum_{S\subseteq V} P^S W_r P^S \mathcal{A}(Y) I_S\\
=& \mathcal{A}(Y) + \sum_{S\subseteq V} P^S W_r P^S (Y + \sum_{T\subseteq V} P^S W P^S Y I_T) I_S\\
=& \mathcal{A}(Y) + \sum_{S\subseteq V} P^S W_r P^S Y I_S + \sum_{T,S\subseteq V} P^S W_r P^S P^T W P^T Y I_T I_S\\
=& \mathcal{A}(Y) + \sum_{S\subseteq V} P^S W_r P^S Y I_S + \sum_{S\subseteq V} P^S W_r W P^S Y I_S\\
=& \mathcal{A}(Y) + \sum_{S\subseteq V} P^S W_r (\mathcal{I} + W) P^S Y I_S,
\end{aligned}$$

where $\mathcal{I}$ is the identity matrix. We note that $\mathcal{B}(Y)$ resembles $\mathcal{A}(Y)$. In $\mathcal{B}(Y)$ we replace Y by $\mathcal{A}(Y)$ and W by $W_r(\mathcal{I} + W)$. Furthermore, the matrix $W_r(\mathcal{I} + W) = W_r + W_r W$ represents the process of generating a new equation by multiplying two equations by two scalars, summing both with a third one and keeping the generated equation.

Lemma 5.2. *The matrix $P^S W_r(\mathcal{I} + W) P^S Y I_S$ has the following form: the element of row $S\Delta(I_1' \cup I_2')$ and column S is*

$$- y_{I_1'\cup I_2'} - \sum_{J\subseteq I_1'\cup I_2'} (-1)^{|I_1'\cup I_2'\setminus J|} s^1_{I_1'\cup I_2'\setminus J} y_J. \tag{9}$$

All other elements are zero.

Proof. As mentioned above the matrix $W_r(\mathcal{I} + W)$ executes exactly the process that we have done in the proof of Proposition 4.1. So $W_r(\mathcal{I} + W)$ contains the coefficients of equation (9) in row $I_1' \cup I_2'$. Applying Lemma 5.1, the result follows. □

Proposition 5.1. *The SDP problem (4) is equivalent to the SDP problem*

$$\begin{aligned}
&\sum_{J\subseteq I_i} (-1)^{|J|} s^i_J y_J = 0, \qquad i = 3, \ldots, m\\
&\mathcal{B}(Y) \in \Omega_n.
\end{aligned} \tag{10}$$

Proof. We only need to prove that $\mathcal{B}(Y) \in \Omega_n$ is equivalent to the system defined by equations 1 and 2 in (4) together with

$$\begin{aligned}
&\sum_{J\subseteq I_1\cup I_2\setminus\{1\}} (-1)^{|J|} \bar{s}_J y_J = 0\\
&Y \in \Omega_n.
\end{aligned}$$

From Lemmata 5.1 and 5.2, we know that in each row L and column J such that $L\Delta J = I_1$, $L\Delta J = I_2$ and $L\Delta J = I_1' \cup I_2'$, the matrix $\mathcal{B}(Y)$ contains the elements

$$- \sum_{J\subset I_1} (-1)^{|I_1\setminus J|} s^1_{I_1\setminus J} y_J,$$

$$-\sum_{J\subset I_2}(-1)^{|I_2\setminus J|}s^2_{I_2\setminus J}y_J,$$

and

$$-\sum_{J\subset I_1'\cup I_2'}(-1)^{|I_1'\cup I_2'\setminus J|}s^1_{I_1'\cup I_2'\setminus J}y_J,$$

respectively. Calling these expressions y_{I_1}, y_{I_2} and $y_{I_1'\cup I_2'}$ the result follows. □

Proposition 5.2. *The SDP formulation $\mathcal{A}(Y)\in\Omega_n$ is feasible if and only if*

$$\begin{aligned}&\mathcal{A}_r(Y)\in\Omega_n\\ &\sum_{J\subseteq I_i}(-1)^{|J|}s^i_J y_J=0,\quad i=1,2\end{aligned}$$

is feasible.

Proof. The sufficient condition follows from the fact that $\mathcal{A}_r(Y)\in\Omega_n$ is the intersection of $\mathcal{A}(Y)\in\Omega_n$ with the hyperplane (3). Now let us assume that $\mathcal{A}(Y)\in\Omega_n$ has a solution. Using this solution, call it y, we redefine the value of $y_{I_1'\cup I_2'}$ such that the hyperplane (3) is satisfied. So it remains to prove that the new vector y satisfies $\mathcal{A}_r(Y)\in\Omega_n$. Since

$$\mathcal{A}_r(Y)=Y+\sum_{S\subseteq V}P^SW_rP^SYI_S,$$

Y is in Ω_n and $P^SW_rP^SYI_S=0$ for each $S\subseteq V$, the results follows. Note that, by definition, $P^SW_rP^SYI_S=0$ for a vector that satisfies condition (3). □

We now state and prove the main result of this section. Theorem 5.1 states that the SDP formulation of a SAT instance is feasible if and only the SDP formulation representing the same SAT instance plus the clause obtained by resolution is feasible.

Theorem 5.1. *The SDP formulation $\mathcal{B}(Y)\in\Omega_n$ is feasible if and only if $\mathcal{A}(Y)\in\Omega_n$ is feasible.*

Proof. Assuming that $\mathcal{A}(Y)\in\Omega_n$ is feasible, it immediately follows by Propositions 5.2 and 5.1 that $\mathcal{B}(Y)\in\Omega_n$ is feasible. The converse is proved by noting that

$$\Omega_n\ni\mathcal{B}(Y)=\mathcal{A}(Y)+\sum_{S\subseteq V}P^SW_r(\mathcal{I}+W)P^SYI_S=\mathcal{A}(Y)$$

because the explicit nonzero elements of matrix $P^SW_r(\mathcal{I}+W)P^SYI_S$ are zero when the expressions (6), (7) and (9) are equal to zero. This follows by Proposition 5.1. □

6 Application: Direct Proof of Exactness of the Semidefinite Formulation for SAT

In this section, we apply SDP resolution to directly derive the result that $\mathrm{SDP}(F)$ is feasible if and only if F is satisfiable.

Lemma 6.1. *We have $\emptyset\in\mathrm{Res}(F)$ if and only if* $\mathrm{SDP}(\mathrm{Res}(F))$ *contains explicitly the constraint* $2=0$.

Proof. Saying that $\emptyset\in\mathrm{Res}(F)$ is equivalent to say that the unit clauses x and $\bar{x}$ belong to $\mathrm{Res}(F)$. But, using the SDP language, this means that $\mathrm{SDP}(\mathrm{Res}(F))$ contains the constraints $1-x=0$ and $1+x=0$.. Applying the SDP resolution process we conclude that $\mathrm{SDP}(\mathrm{Res}(F))$ contains the constraint $(1-x)+(1+x)=0$ $\Leftrightarrow 2=0$. The reciprocal is clear, because $\mathrm{SDP}(\mathrm{Res}(F))$ only contains explicitly the constraint $2=0$ if we have previously deduced it by doing SDP resolution from equations $1-x=0$ and $1+x=0$. □

Theorem 6.1. *The following conditions are equivalent.*

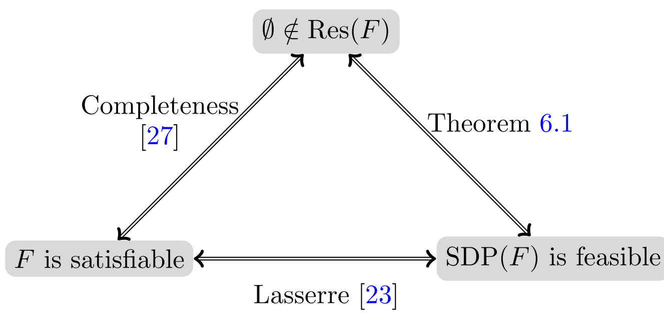
(i) F *is satisfiable;*

(ii) $\emptyset \notin \mathrm{Res}(F)$*;*

(iii) *SDP(*F*) is feasible.*

Proof. The equivalence between (i) and (ii) is just the completeness result for resolution rule, Theorem 2.1. $(ii) \Leftrightarrow (iii)$: We say that $\emptyset \in \mathrm{Res}(F)$ if and only if $\mathrm{SDP}(\mathrm{Res}(F))$ contains the constraint $2 = 0$, meaning that $\mathrm{SDP}(\mathrm{Res}(F))$ is infeasible. Therefore, $\emptyset \notin \mathrm{Res}(F)$ is equivalent to saying that $\mathrm{SDP}(\mathrm{Res}(F))$ is feasible. Thus, by repeated application of Theorem 5.1, $\mathrm{SDP}(\mathrm{Res}(F))$ is feasible if and only if $\mathrm{SDP}(F)$ is feasible. $\square$

The situation can be visualized as follows:



An important observation is that the statement and proof of Theorem 6.1 show that the SDP problem $\mathrm{SDP}(F)$ implicitly computes $\mathrm{Res}(F)$, and thus the SDP problem is feasible if and only if F is satisfiable. This is made explicit through our short and direct proof based on SDP resolution.

7 Conclusion and Future Research

In this paper we introduced the concept of SDP resolution. This concept follows from the fact that the process of resolution in SAT is shown here to be equivalent to a linear transformation between the feasible sets of the relevant SDP problems. We demonstrated the potential of this novel concept by using it to obtain a short and direct proof of the exactness of an exponential-sized SDP relaxation of SAT without applying Lasserre's general theory for SDP relaxations of 0/1 problems. In particular, our proof explicitly shows that the SDP relaxation implicitly detects whether a finite sequence of resolution steps leads to the empty clause. In this way, SDP resolution provided a new and deep connection between SAT and SDP.

If an SDP problem is infeasible, SDP solvers based on the application of an interior-point method will return a (numerical) certificate of infeasibility. One important question that remains unanswered is how to interpret the information contained in this certificate in terms of SAT. We expect that the concept of SDP resolution will help in making progress in this direction.

From a computational point of view, we expect that the concept of SDP resolution will provide information to help design new polynomial-sized SDP relaxations of SAT in a dynamic fashion. Ultimately, the objective of this research is an efficient SDP-based algorithm for proving unsatisfiability of large SAT instances which remains a very challenging problem.

8 Appendix

Let I_1 and I_2 be two finite nonempty sets with p common elements, i.e., $|P| = |I_1 \cap I_2| = p$. The total number of subsets of I_1 is $2^{|I_1|}$, and the number of subsets of I_1 with a even (odd) number of elements is $2^{|I_1|-1}$. If we do not include the empty set in this counting, then the number of subsets of I_1 of cardinality even is $2^{|I_1|-1} - 1$.

Let $I_1 \Delta I_2$ denote the symmetric difference of I_1 and I_2:

$$I_1 \Delta I_2 := (I_1 \cup I_2) \setminus (I_1 \cap I_2).$$

Consider the sets

$$T_e = \{(J_1, J_2) | J_1 \subseteq I_1 \setminus \{1\},\ J_2 \subseteq I_2 \setminus \{1\} \text{ and } |J_1 \Delta J_2| \text{ is even }\}$$

and

$$T_o = \{(J_1, J_2) | J_1 \subseteq I_1,\ J_2 \subseteq I_2 \setminus \{1\} \text{ and } |J_1 \Delta J_2| \text{ is odd }\}.$$

Note that $|J_1 \Delta J_2|$ is even if and only if $|J_1|$ and $|J_2|$ are both even or both odd. Therefore

$$\begin{aligned} |T_e| &= (2^{|I_1|-1} - 1)(2^{|I_2|-1} - 1) + 2^{|I_1|-1} 2^{|I_2|-1} \\ &= 2^{|I_1|+|I_2|-1} - 2^{|I_1|-1} - 2^{|I_2|-1} + 1. \end{aligned}$$

Similarly $|J_1 \Delta J_2|$ is odd if and only if $|J_1|$ is odd and $|J_2|$ is even or vice versa. Thus,

$$\begin{aligned} |T_o| &= 2^{|I_1|-1}(2^{|I_2|-1} - 1) + 2^{|I_2|-1}(2^{|I_1|-1} - 1) \\ &= 2^{|I_1|+|I_2|-1} - 2^{|I_1|} - 2^{|I_2|}. \end{aligned}$$

We also have $|T_e| = |T_o| + 1$.

The number of sets $J_1 \subseteq I_1$ such that $J_1 \not\subseteq P$ is given by $2^{|I_1|} - 2^p$. Therefore the number of odd and even sets in the previous conditions are $\frac{1}{2}(2^{|I_1|} - 2^p)$. Note that the empty set is removed in this counting.

References

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

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

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

[4] M.F. Anjos. An explicit semidefinite characterization of satisfiability for Tseitin instances on toroidal grid graphs. *Ann. Math. Artif. Intell.*, 48(1-2):1–14, 2006.

[5] M.F. Anjos. An extended semidefinite relaxation for satisfiability. *J. Satisf. Boolean Model. Comput.*, 4(1):15–31, 2008.

[6] M.F. Anjos and J.B. Lasserre, editors. *Handbook on Semidefinite, Cone and Polynomial Optimization*. International Series in Operations Research & Management Science.. Springer-Verlag, to appear in 2011.

[7] M.F. Anjos and J.B. Lasserre. *Introduction to Semidefinite, Conic and Polynomial Optimization*, chapter 1. In *International Series in Operations Research & Management Science* [6], to appear in 2011.

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

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

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

[11] 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] A. Biere, M.J.H. Heule, H. van Maaren, and T. Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, February 2009.

[13] 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.

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

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

[17] 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.

[18] M.M. Deza and M. Laurent. *Geometry of Cuts and Metrics*, volume 15 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 1997.

[19] 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.

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

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

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

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

[24] M. Laurent. Semidefinite relaxations for max-cut. In M. Grötschel, editor, *The Sharpest Cut: The Impact of Manfred Padberg and His Work*. SIAM, 2004.

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

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

[27] V.W. Marek. *Introduction to mathematics of satisfiability*. Chapman & Hall/CRC Studies in Informatics Series. CRC Press, Boca Raton, FL, 2009.

[28] 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.

[29] 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.

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

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

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

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

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

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