On semidefinite least squares and minimal unsatisfiability

Citation for published version:
Anjos, MF & Vieira, MVC 2017, 'On semidefinite least squares and minimal unsatisfiability' Discrete Applied Mathematics, vol. 217, pp. 79-96. DOI: 10.1016/j.dam.2016.09.008

Digital Object Identifier (DOI):
10.1016/j.dam.2016.09.008

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Discrete Applied Mathematics

On Semidefinite Least Squares and Minimal Unsatisfiability

Miguel F. Anjos[‡]
Manuel V.C. Vieira[§]

September 2016

Abstract

This paper provides new results on the application of semidefinite optimization to satisfiability by studying the connection between semidefinite optimization and minimal unsatisfiability. We use a semidefinite least squares problem to assign weights to the clauses of a propositional formula in conjunctive normal form. We then show that these weights are a measure of the necessity of each clause in rendering the formula unsatisfiable, the weight of a necessary clause is strictly greater than the weight of any unnecessary clause. In particular, we show the following results: first, if a formula is minimal unsatisfiable, then all of its clauses have the same weight; second, if a clause does not belong to any minimal unsatisfiable subformula, then its weight is zero. An additional contribution of this paper is a demonstration of how the infeasibility of a semidefinite optimization problem can be tested using a semidefinite least squares problem by extending an earlier result for linear optimization. The connection between the semidefinite least squares problem and Farkas' Lemma for semidefinite optimization is also discussed.

1 Introduction

The Boolean satisfiability (SAT) problem is at the crossroads of several important areas, including logic, computer science, graph theory, and operations research. It has numerous practical applications in these fields and others, as documented in the Handbook [9]. The problem consists of determining whether or not it is possible to satisfy a given propositional formula by at least one assignment of the values true/false to the Boolean variables appearing in the formula. It is a famous result that SAT is in general NP-complete [16], and it is in general a challenging problem to detect that a SAT instance is unsatisfiable and to provide insight into its unsatisfiability.

Unsatisfiability can occur for multiple reasons, and explaining its causes is a key requirement in a number of practical applications. There is an important body of literature concerned with, given a propositional formula that is unsatisfiable, obtaining an unsatisfiable subformula, and proving guarantees on the size of computed subformulas, see e.g. [22, 31, 30, 25]. Most of this work has focused on computing one or all minimal unsatisfiable subformulas (MUSs). In particular, Kullmann et al. [28] provided a differentiated analysis of the causes of unsatisfiability through a classification

[‡]Professor and Canada Research Chair, GERAD & École Polytechnique de Montréal, Montréal, QC, Canada H3C 3A7, email: `anjos@stanfordalumni.org`. Research partially supported by the Natural Sciences and Engineering Research Council of Canada.

[§]Departamento de Matemática, Faculdade de Ciências e Tecnologia & CMA, Universidade Nova de Lisboa, Portugal, email: `mvcv@fct.unl.pt`. This work was partially supported by CMA/FCT/UNL, under the project PEst-OE/MAT/UI0297/2013.

of single clauses based on the contribution of each clause to the causes of unsatisfiability. Their classification varies from clauses that are necessary to prove unsatisfiability to unusable clauses. The highest degree of necessity corresponds to necessary clauses, where a clause is said to be necessary if every resolution refutation of the given formula must use this clause.

Unsatisfiability can also be expressed using optimization. This is known at least since the pioneering work of Williams [37] and Blair et al. [10] on the connections between inference in propositional logic and integer linear programming. 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 [15]. For certain classes of instances, including Horn formulas and their generalizations, the exactness of the linear programming relaxation has been established, see e.g. [11].

Semidefinite optimization, or 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. The Handbooks [38] and [6] provide a wide coverage of SDP theory, algorithms, software, and application areas in which SDP has had a major impact. One of the best-known results in SDP is due to Goemans and Williamson who proposed SDP-based polynomial-time approximation schemes for MAX-CUT and MAX-2-SAT [24]. Further research has deepened the connections between SDP and the SAT and MAX-SAT problems, see the recent survey chapter [5].

The SAT problem can be formulated as an SDP problem with a rank-one constraint, and removing the rank constraint yields a convex optimization problem that is an SDP relaxation of SAT, see e.g. [2]. This can be done in different ways. In [20, 21], de Klerk, van Maaren, and Warners introduced the Gap SDP relaxation and showed that it is exact for some well-known classes of SAT instances, in the sense that the Gap SDP is infeasible if and only if the SAT instance is unsatisfiable. Subsequent papers by Anjos [1, 2, 3, 4] and van Maaren et al. [35, 36] proposed several SDP relaxations for different versions of SAT, and some exactness results for particularly structublack SAT formulas. An exact SDP relaxation for general SAT is obtained by formulating the SAT instance as a binary optimization problem and then constructing the corresponding Lasserre SDP relaxation [29]. Lasserre's theory proves that this SDP relaxation is always exact. However, because the size of the relaxation is exponential in the number of Boolean variables in the instance, it is computationally impractical for all but very small instances of SAT.

A more direct connection between SAT and SDP was given in our recent paper [8] where we proved that the process of resolution in SAT is equivalent to a linear transformation between the feasible sets of SDP relaxations. This equivalence between resolution and SDP, called *SDP resolution*, makes it possible to write a direct proof of the exactness of Lasserre's SDP relaxation in the specific context of SAT without recourse to Lasserre's general theory. The exactness proof in [8] shows that the exact relaxation implicitly deduces whether the empty clause can be derived by a finite sequence of resolution steps starting from the SAT formula. It then follows that the SDP relaxation is infeasible precisely when such a sequence of steps exists.

This paper provides new results on the application of SDP in the context of unsatisfiability. Specifically we focus here is on the connection between SDP and minimal unsatisfiability. A CNF formula (formally defined in Section 1.1) is minimal unsatisfiable (MU) if it is unsatisfiable but if any clause is removed then the resulting formula is satisfiable. We establish a connection between SDP and minimal unsatisfiability by using a semidefinite least squares (SLS) problem to associate non-negative weights to the clauses in a formula. We then argue that these weights are a measure of the importance of each clause in rendering the formula unsatisfiable. In particular, we prove that this approach identifies two important cases: first, if an unsatisfiable formula is MU, then all of its clauses have the same weight; second, if a clause does not belong to any minimal unsatisfiable subformula (MUS) of an unsatisfiable formula, then its weight is zero.

The computation of these weights is in general a hard problem. To show this, let F denote the set of clauses in a CNF formula such that $|F| \geq 2$, and for $C \subseteq F$, let $w(C)$ be a assignment of non-negative weights to the clauses in C. Consider the following decision problem (CW):

$$\text{Given } F\text{, is } w(C) \text{ constant for } C \subseteq F?$$

We argue that (CW) is NP-hard by following the idea of the blackuction from SAT to minimal unsatisfiability in Lemma 2 of [34]. Specifically we build an unsatisfiable set of clauses G' by adding the clause Y to the clause-set G from the proof of Lemma 1 of [34], where G is MU if and only if F is unsatisfiable. Because Y has a clash with at least one clause of G, it follows that G' contains a clause that cannot be removed without destroying unsatisfiability. Thus, there is a polytime blackuction that from F constructs G' such that:

- if F is satisfiable then G' is MU, and hence $w(C)$ is constant (Theorem 5.2 in this paper).
- if F is unsatisfiable then G' contains a blackundant and a necessary clause, and hence $w(C)$ is not constant (Theorem 5.1 in this paper).

It follows that (CW) is NP-hard.

The motivation for this work is that, in spite of the hardness of the problem, the identification of MUSs remains a need in practice. The optimization problem (14) is a convex optimization problem that is in principle solvable in polynomial time except for the fact that the number of variables is exponential in the number of Boolean variables. Moreover, problem (14) has a structure that may be exploited in future to design a practical algorithm.

An additional contribution of this paper is an exploration of the use of SLS to test the infeasibility of an SDP problem. This is an extension of an earlier result of Dax [18] for linear optimization. We also explain the connection between the SLS problem and a semidefinite version of the well-known Farkas' Lemma.

Farkas' Lemma was used in the SAT context in [17] where a connection is made between a non-trivial solution of an homogenous system and CNF formulas that are tautologies. The problem of efficiently deleting clauses that do not contribute to any proof of unsatisfiability was studied in [27]. Using a Farkas' Lemma variant, classes of formulas where selecting a MUS is easy were investigated in [13]. These and other results on unsatisfiability can be found in the major source [14]. Practice-oriented papers to determine or approximate MUSs have been published by several authors, see e.g. [13, 25, 32].

The results of this article are related to the class UMU of finite unions of minimal unsatisfiable CNF formulas. This class first appeablack in [33] under the name "effective unsatisfiable set of clauses", while the class UMU was named and studied in [28] as "potentially necessary clauses". Trivially the union of all MUSs in some formula F is the largest UMU in F. Theorem 5.3 in this paper shows that clauses outside of the largest UMU have weight zero.

This paper is structublack as follows. Section 1.1 formally introduces SAT and SDP, and recalls the exact SDP formulation of SAT. In Section 2 we introduce the SLS problem for testing infeasibility of a general SDP problem, and in Section 3 we show how the SLS approach can be used to obtain SLS certificates of infeasibility for SAT. We then recall in Section 4 the concept of MUS and prove a new characterization of MUSs. In Section 5 we show the main results in this paper. Specifically we show that the SLS certificate of infeasiblity for an unsatisfiable formula yields a weight for each clause, and that these weights are a measure of the importance of each clause in rendering the formula unsatisfiable (Theorem 5.1). We further show that if the unsatisfiable formula is MU, then all of its clauses have the same weight (Theorem 5.2), and that if a clause does not belong to any MUS, then its weight is zero (Theorem 5.3). Section 6 presents three examples to illustrate our results, and Section 7 concludes the paper and proposes some directions for future research.

1.1 The Exact SDP Formulation of SAT

We consider the SAT problem for instances in conjunctive normal form (CNF). Such instances are specified by a set of Boolean variables $x_1, \ldots, x_n$ a propositional formula $F = \bigwedge\limits_{j=1}^{m} C_j$, with each clause C_j having the form $C_j = \bigvee\limits_{k \in B_j} x_k \vee \bigvee\limits_{k \in \bar{B}_j} \bar{x}_k$ where $B_j, \bar{B}_j \subseteq \{1, \ldots, n\}$, $B_j \cap \bar{B}_j = \emptyset$, and $\bar{x}_i$ denotes the negation of x_i. 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 F evaluates to TRUE?

We use the common description of the constraints of an SDP optimization problem (see e.g. [7]):

$$A_i \bullet X = b_i, \; i = 1, \ldots, m, \quad X \succeq 0, \tag{1}$$

where the matrices A_i and X are $n \times n$ real symmetric, $b \in \Re^m$ is a column vector, $X \succeq 0$ denotes that the matrix $X \in \mathcal{S}_+^n$, where $\mathcal{S}_+^n$ is the set of $n \times n$ real symmetric positive semidefinite matrices, and $M \bullet N$ denotes the inner product of two real symmetric matrices:

$$M \bullet N = \operatorname{trace}(M\, N) = \sum_{i=1}^{n} \sum_{j=1}^{n} M_{i,j} N_{i,j}$$

We may write the equality constraints of (1) as $\mathcal{A}(X) = b$, where the linear mapping $\mathcal{A}$ is defined as $\mathcal{A}(X) = (A_1 \bullet X, \ldots, A_m \bullet X)^T$. The adjoint of $\mathcal{A}$ is denoted $\mathcal{A}^T(u)$ with $u \in \Re^m$, and can be expressed as $\sum\limits_{j=1}^{m} u_j A_j$.

We use the fact that the SAT problem can be expressed in the form (1). First, we explain how each clause can be expressed as a linear constraint. Let TRUE be denoted by 1 and FALSE be denoted by -1, and express clause C_j as $C_j = \bigvee_{i \in I_j} s_i^j x_i$, where $I_j = B_j \cup \bar{B}_j$, x_i are the propositional variables, and the parameter s_i^j indicates whether x_i is negated or not in clause C_j, i.e., for $i \in I_j$, we let

$$s_i^j = \begin{cases} \;\;1, & \text{if } x_i \text{ is not negated in clause } C_j \\ -1, & \text{if } x_i \text{ is negated in clause } C_j \end{cases}$$

Then clause C_j is satisfied by a truth assignment (of ± 1) to the variables x_i if and only if

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

Expanding this product, we have

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

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

$$\sum_{J \subseteq I_j} (-1)^{|J|} s_J y_J = 0. \tag{2}$$

Therefore (2) gives a way to represent each clause by means of a constraint that is linear in the variables y_J. We illustrate this representation in the following example.

Example 1.1. Let $C_1 = \bar{x}_1 \vee x_2 \vee x_3$. We have $s_1^1 = -1, s_2^1 = 1, s_3^1 = 1$. Clause C_1 is satisfied if $s_i^1 x_i = 1$ for at least one i, i.e., if

$$(1 - s_1^1 x_1)(1 - s_2^1 x_2)(1 - s_3^1 x_3) = 0.$$

Because $(1 - s_1^1 x_1)(1 - s_2^1 x_2)(1 - s_3^1 x_3) = (1 + x_1)(1 - x_2)(1 - x_3)$, this condition is equivalent to

$$1 + x_1 - x_2 - x_3 - x_1x_2 - x_1x_3 + x_2x_3 + x_1x_2x_3 = 0.$$

Setting $y_{\{1\}} = x_1$, $y_{\{2\}} = x_2$, $y_{\{3\}} = x_3$, $y_{\{12\}} = x_1x_2$, $y_{\{13\}} = x_1x_3$, $y_{\{23\}} = x_2x_3$, $y_{\{123\}} = x_1x_2x_3$, we obtain the expression of clause C_1 in the form (2):

$$1 + y_{\{1\}} - y_{\{2\}} - y_{\{3\}} - y_{\{12\}} - y_{\{13\}} + y_{\{23\}} + y_{\{123\}} = 0.$$

Second, to conform to the form (1), we need to express the nonlinear relationship between the new formal variable y_J and the formal variables x_i, $i \in J$, using only linear constraints and a semidefinite constraint. Lasserre [29] proved that this can be done using semidefinite matrices of size $2^n \times 2^n$. To use this result, fix an ordering $\mathcal{O} = \{J_1, J_2, \ldots, J_{2^n}\}$ of the subsets of $\{1, \ldots, n\}$ and define the column vector containing the variables y_J according to $\mathcal{O}$:

$$y = (y_{J_1}, y_{J_2}, \ldots, y_{J_{2^n}})^T.$$

Now define the $2^n \times 2^n$ symmetric matrix Y as the rank-one matrix

$$Y = yy^T.$$

Clearly the resulting matrix Y is positive semidefinite (see e.g. [7]), and the elements of Y equal

$$Y_{I,J} = y_I y_J = \left(\prod_{i \in I} x_i\right)\left(\prod_{i \in J} x_i\right).$$

Note that $Y_{\emptyset,J} = y_J$, and hence the $\emptyset$ row (and column) of Y contains the y_J variables. Because $x_i = \pm 1$, if $i \in I \cap J$ then the resulting x_i^2 term in the definition of $Y_{I,J}$ equals 1 and can be omitted. Thus we have that $Y_{I,J} = y_{I \Delta J}$, where Δ denotes the symmetric difference of the sets I and J. Therefore, the diagonal elements $Y_{J,J}$ of Y equal 1. Furthermore, the element $Y_{I,J}$ is equal to each element of the form $Y_{(I\Delta P),(P\Delta J)}$ for every nonempty subset P of $\{1, \ldots, n\}$. The following example illustrates these properties.

Example 1.2. Considering the previous example, we note, for instance, that $\{23\}\Delta\{12\} = \{1,3\}$, and therefore

$$Y_{\{23\},\{12\}} = y_{\{23\}}y_{\{12\}} = x_2x_3x_1x_2 = x_1x_3 = y_{\{13\}}.$$

Another example is

$$Y_{\{23\},\{23\}} = y_{\{23\}}y_{\{23\}} = x_2x_3x_2x_3 = 1$$

because $\{23\}\Delta\{23\} = \emptyset$. Lastly, for $P = \{1\}$, we have

$$Y_{\{2\}\Delta\{1\},\{1,3\}\Delta\{1\}} = Y_{\{1,2\},\{3\}} = y_{\{1,2,3\}} = x_1x_2x_3 = y_{\{2\}\Delta\{1,3\}} = Y_{\{2\},\{1,3\}}.$$

The above discussion leads to the following definition of the set Ω_n:

Definition 1.1. For $n \in \mathbb{N}$ and the ordering $\mathcal{O}$ of the subsets of $\{1,\ldots,n\}$, define the set

$$\Omega_n = \{Y \in \mathbb{R}^{2^n \times 2^n} | Y = Y^T, Y \succeq 0, Y_{J,J} = 1, \text{ and } Y_{(I\Delta P),(P\Delta J)} = Y_{I,J}, \ \emptyset \subsetneq P \subseteq \{1,\ldots,n\}\}.$$

The entries of the matrices Y in Ω_n are thus linearizations of the products of variables x_i.

Third, we make a connection between the elements of Ω_n and the valuations of the Boolean variables x_i. We encode each of the truth assignments to the variables x_i in a vector x^S of length n as follows. For each $S \subseteq \{1,\ldots,n\}$, the entries of x^S are defined according to the following recursive rule (used in [23]):

$$x_1^S = \begin{cases} 1 & \text{if} \quad 1 \in S \\ -1 & \text{if} \quad 1 \notin S, \end{cases}$$

and for $k = 2,\ldots,n$:

$$x_k^S = \begin{cases} -x_{k-1}^S & \text{if} \quad k \in S \\ x_{k-1}^S & \text{if} \quad k \notin S \end{cases}$$

Example 1.3. If $n = 3$ and $S = \{1,2\}$, then

- $x_1^{\{1,2\}} = 1$ because $1 \in \{1,2\}$,
- $x_2^{\{1,2\}} = -x_1^{\{1,2\}} = -1$ because $2 \in \{1,2\}$, and
- $x_3^{\{1,2\}} = x_2^{\{1,2\}} = -1$ because $3 \notin \{1,2\}$.

Hence $x^{\{1,2\}} = (1,-1,-1)^T$.

Remark 1.1. For $R, S \subseteq \{1,\ldots,n\}, R \neq S$, we have $x^S \neq x^R$ because if p is the first element such that $p \in S$ and $p \notin R$, then $x_{p+1}^S = -x_p^S$, $x_{p+1}^R = x_p^R$ and $x_p^R = x_p^S$.

Note that the vectors x^S are the vertices of the cube $[-1,1]^n$. In accordance with the above, we write

$$y_\emptyset^S = 1, \ y_{\{k\}}^S = x_k^S, \text{ and } y_J^S = \prod_{i\in J} x_i^S.$$

and

$$Y^S = (y^S)(y^S)^T.$$

The matrices Y^S are the vertices of the set Ω_n, and hence their convex hull is contained in Ω_n. Moreover Ω_n is equal to this convex hull; this follows from applying [29, Theorem 3.2] to the max-cut problem, as exemplified in [29, Section 3.1].

Because Ω_n is equal to the convex hull of the rank-one matrices of the form Y^S, an exact formulation of SAT can be stated using the set Ω_n plus the linear constraints corresponding to the clauses:

Definition 1.2. Let F be a CNF formula on the variables $x_1, \ldots, x_n$ and with I_i the index set of variables of each clause $i = 1, \ldots, m$. We define the SDP representation of F as:

$$\begin{aligned} &\sum_{J \subseteq I_i} (-1)^{|J|} s_J^i Y_{\emptyset,J} = 0, \quad i = 1, \ldots, m \\ &Y \in \Omega_n. \end{aligned} \tag{SDP(F)}$$

The following example explicitly states an SDP formulation for a short CNF formula, and illustrates the structure of the elements of Ω_n.

Example 1.4. Let

$$F = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_2 \vee x_3)$$

be a CNF formula. The SDP formulation is

$$1 - y_{\{1\}} - y_{\{2\}} + y_{\{12\}} = 0 \tag{3}$$

$$1 + y_{\{1\}} - y_{\{2\}} - y_{\{3\}} - y_{\{12\}} - y_{\{13\}} + y_{\{23\}} - y_{\{123\}} = 0 \tag{4}$$

$$Y \in \Omega_3, \tag{5}$$

where $Y \in \Omega_3$ is equivalent to Y having the following structure:

$$Y = \begin{pmatrix} 1 & y_{\{1\}} & y_{\{2\}} & y_{\{3\}} & y_{\{12\}} & y_{\{13\}} & y_{\{23\}} & y_{\{123\}} \\ y_{\{1\}} & 1 & y_{12} & y_{\{13\}} & y_{\{2\}} & y_{\{3\}} & y_{\{123\}} & y_{\{23\}} \\ y_{\{2\}} & y_{\{12\}} & 1 & y_{\{23\}} & y_{\{1\}} & y_{\{123\}} & y_{\{3\}} & y_{\{13\}} \\ y_{\{3\}} & y_{\{13\}} & y_{\{23\}} & 1 & y_{\{123\}} & y_{\{1\}} & y_{\{2\}} & y_{\{12\}} \\ y_{\{12\}} & y_{\{2\}} & y_{\{1\}} & y_{\{123\}} & 1 & y_{\{23\}} & y_{\{13\}} & y_{\{3\}} \\ y_{\{13\}} & y_{\{3\}} & y_{\{123\}} & y_{\{1\}} & y_{\{23\}} & 1 & y_{\{12\}} & y_{\{2\}} \\ y_{\{23\}} & y_{\{123\}} & y_{\{3\}} & y_{\{2\}} & y_{\{13\}} & y_{\{12\}} & 1 & y_{\{1\}} \\ y_{\{123\}} & y_{\{23\}} & y_{\{13\}} & y_{\{12\}} & y_{\{3\}} & y_{\{2\}} & y_{\{1\}} & 1 \end{pmatrix}.$$

Here, equations (3) and (4) represent the clauses $x_1 \vee x_2$ and $\bar{x}_1 \vee x_2 \vee x_3$, respectively.

It may happen that SDP(F) has no solution, in which case we say that it is *infeasible* (see also Definition 2.1 below). Lasserre's result implies that SDP(F) is infeasible if and only if F is unsatisfiable [8].

In the next section, we explore a characterization of infeasibility for SDP(F) via a least squares problem.

2 A Certificate of SDP Infeasibility via Least-Squares

Unlike in linear programming, it is possible for a set of SDP constraints to be weakly infeasible, as per the following definition [19, Definitions 2.4 and 2.5]:

Definition 2.1. The set of constraints (1) always satisfies one of the following three properties:

- it is *feasible* if there exists $X \succeq 0$ such that $A_i \bullet X = b_i, \quad i = 1, \ldots, m$;
- it is *weakly infeasible* if it is not feasible and for each $\epsilon > 0$ there exists $X \succeq 0$ such that

$$|A_i \bullet X - b_i| \leq \epsilon, \quad i = 1, \ldots, m.$$

- it is *strongly infeasible* if it is not feasible and there exists $\epsilon > 0$ such that for all $X \succeq 0$ there exists $i \in \{1, \ldots, m\}$ such that
$$|A_i \bullet X - b_i| > \epsilon.$$

In the following, we prove that SDP(F) cannot be weakly infeasible. This is desirable because it confirms that SDP(F) can make a clear distinction between satisfiability and unsatisfiability of F.

Theorem 2.1. *The formulation SDP(F) is either feasible or strongly infeasible.*

Proof. Clearly, SDP(F) is either feasible or infeasible. Let us assume that SDP(F) is infeasible. We prove that, for each $Y \in \Omega_n$, there is at least one equation such that its violation by Y is not less than a positive quantity ϵ. Let q_i denote the ith constraint in SDP(F) formulation. For any point Y^S with $S \subseteq V$ we have that

$$q_i(Y^S) = \sum_{J \subseteq I_i} (-1)^{|J|} s_J y_J^S = \prod_{j \in I_i} (1 - s_j x_j).$$

Since F is unsatisfiable, for every truth assignment x^S there is at least one clause which evaluates false. Thus, for any point Y^S not satisfying $q_i(Y^S) = 0$ we have

$$q_i(Y^S) = \prod_{j \in I_i} (1 - s_j x_j^S) = 2^{|I_i|}.$$

Any $Y \in \Omega_n$ is a convex combination of the extreme points $Y^S = y^S (y^S)^T$, $S \subseteq V$, i.e.,

$$Y = \sum_{S \subseteq V}^{n} \alpha_S Y^S, \quad \text{with} \ \ 0 \leq \alpha_S \leq 1, \ S \subseteq V \ \ \text{and} \ \sum_{S \subseteq V} \alpha_S = 1. \tag{6}$$

Then

$$q_i(Y) = \sum_{S \subseteq V} \alpha_S q_i(Y^S) \geq \alpha_S q_i(Y^S) \geq \frac{1}{2^n} 2^{|I_i|} = \frac{2^{|I_i|}}{2^n},$$

where we chose the constraint q_i not satisfied by Y^S such that its coefficient α_S is at least $\frac{1}{2^n}$. Setting $\epsilon = \frac{1}{2^n}$, the result is proved. □

Let A_i, $i = 1, \ldots, m$ be $n \times n$ real symmetric matrices and b a nonzero m-vector. Then we have the following variant of Farkas' Lemma [26, Lemma 2.2.4]:

Lemma 2.1. *Suppose that the set $\{\mathcal{A}(X) : X \succeq 0\}$ is closed and let $b \in \mathbb{R}^m$. Then exactly one of the following systems has a solution:*

(i) The primal system $A_i \bullet X = b_i$, $i = 1, \ldots, m$ and $X \succeq 0$.

(ii) The dual system $\sum_{i=1}^m u_i A_i \preceq 0$, $b^T u > 0$ with $u \in \mathbb{R}^m$.

In the following, we give a statement of Farkas' Lemma using an SLS problem. This is an extension of the theorem of Dax [18] for linear programming. The idea is that determining which of the two systems has a solution can be answeblack by considering the bounded least squares problem:

$$\begin{array}{ll} \min & \|b - \mathcal{A}(X)\|^2 \\ \text{s.t.} & X \succeq 0, \end{array} \tag{SLS}$$

where $\| \ \|$ denotes the Euclidean norm. The minimum is attained if the set $\{\mathcal{A}(X) \mid X \succeq 0\}$ is closed. For $X \in \mathcal{S}^n_+$, define the corresponding residual vector as $r(X) = b - \mathcal{A}(X)$. We have the following result.

Lemma 2.2. *Suppose that $\{\mathcal{A}(X) : X \succeq 0\}$ is closed. If $X^* \in \mathcal{S}^n_+$ and $r^* = r(X^*)$ is its residual vector, then X^* solves* (SLS) *if and only if X^* and r^* satisfy*

$$X^* \succeq 0, \quad \mathcal{A}^T(r^*) \preceq 0, \quad \text{and} \quad X^* \bullet \mathcal{A}^T(r^*) = 0. \tag{7}$$

Proof. Assume that X^* solves (SLS) and consider the following family of quadratic functions parametrized by θ:

$$f_i(\theta) = \|b - \mathcal{A}(X^* + \theta U_i)\|^2 = \|r^* - \theta \mathcal{A}(U_i)\|^2,$$

where $U_i = u_i u_i^T$ and $u_i \in \mathbb{R}^n$ is the normalized eigenvector associated to the eigenvalue $\lambda_i(X^*)$. Since X^* solves (SLS), then the minimum of the problem

$$\begin{array}{ll} \min & f_i(\theta) \\ \text{s.t.} & \lambda_i(X^*) + \theta \geq 0. \end{array} \tag{8}$$

is attained when $\theta = 0$. Observe that $f_i'(0) = -2\mathcal{A}(U_i)^T r^*$. We have two cases:

1. if $\lambda_i(X^*) > 0$, then $\theta = 0$ is a stationary point, implying $f_i'(0) = 2\mathcal{A}(U_i)^T r^* = 0$;

2. if $\lambda_i(X^*) = 0$, then $\theta = 0$ may not be a stationary point, implying $f_i'(0) = -2\mathcal{A}(U_i)^T r^* \geq 0$.

Thus

$$X^* \bullet \mathcal{A}^T(r^*) = \mathcal{A}(X^*)^T r^* = \mathcal{A}\left(\sum_{i=1}^n \lambda_i(X^*) U_i\right)^T r^* = \sum_{i=1}^n \lambda_i(X^*) \mathcal{A}(U_i)^T r^* = 0.$$

To prove the remaining condition, we similarly define

$$f_i(\theta) = \|r^* - \theta \mathcal{A}(U)\|^2,$$

for any extreme direction $U \in \mathcal{S}^n_+$. Again, $\theta = 0$ solves the problem

$$\begin{array}{ll} \min & f_i(\theta) \\ \text{s.t.} & \theta \geq 0, \end{array}$$

which implies $f_i'(0) = -2\mathcal{A}(U)^T r^* \geq 0$. Thus, $U \bullet \mathcal{A}^T(r^*) \leq 0$ gives that $Z \bullet \mathcal{A}^T(r^*) \leq 0$ for any $Z \in \mathcal{S}^n_+$. Hence $\mathcal{A}^T(r^*) \preceq 0$.

Conversely, we assume that (7) holds and let $Z \succeq 0$. Let $U \in \mathcal{S}^n$ be defined by $U = Z - X^*$. Then

$$0 = X^* \bullet \mathcal{A}^T(r^*) = (Z - U) \bullet \mathcal{A}^T(r^*) = Z \bullet \mathcal{A}^T(r^*) - U \bullet \mathcal{A}^T(r^*),$$

which leads to

$$U \bullet \mathcal{A}^T(r^*) = Z \bullet \mathcal{A}^T(r^*) \leq 0.$$

Hence, the identity

$$\|b - \mathcal{A}(Z)\|^2 = \|b - \mathcal{A}(U) - \mathcal{A}(X^*)\|^2 = \|b - \mathcal{A}(X^*)\|^2 - \mathcal{A}(U)^T r^* + \|\mathcal{A}(U)\|^2$$

shows that

$$\|b - \mathcal{A}(Z)\|^2 \geq \|b - \mathcal{A}(X^*)\|^2.$$

□

Condition (7) gives

$$b^T r^* = (\mathcal{A}(X^*) + r^*)^T r^* = X^* \mathcal{A}^T(r^*) + (r^*)^T r^* = \|r^*\|^2$$

and the following variant of Farkas' lemma follows.

Theorem 2.2 (SLS form of Farkas' Lemma)**.** *Suppose that X^* solves* (SLS) *and that $r^* = b - \mathcal{A}(X^*)$ is the corresponding residual vector. Then*

(i) If $r^* = 0$, then $A_i \bullet X^* = b_i$, $i = 1, \dots, m$ and $X^* \succeq 0$ (i.e., X^* satisfies the primal system);

(ii) Otherwise $\sum_{i=1}^m r_i^* A_i \preceq 0$, $b^T r^* > 0$ (i.e., r^* satisfies the dual system), and $b^T r^* = \|r^*\|^2$.

Clearly $r^* \neq 0$ is a certificate that the primal system has no solution.

Corollary 2.1. *Let X^* and r^* be as in Theorem 2.2 and assume that $r^* \neq 0$. Then the vector $r^*/\|r^*\|$ solves the problem*

$$\begin{array}{ll} \max & b^T u \\ \text{s.t.} & \mathcal{A}^T(u) \preceq 0 \\ & \|u\| = 1. \end{array} \tag{9}$$

Proof. Let $u \in \mathbb{R}^n$ be a feasible point of (9). Then

$$X^* \bullet \mathcal{A}^T(u) \leq 0$$

and the Cauchy-Schwartz inequality gives

$$|(r^*)^T u| \leq \|r^*\| \|u\| = \|r^*\|.$$

Combining these relations we show that

$$b^T u = (\mathcal{A}(X^*) + r^*)^T u = X^* \bullet \mathcal{A}^T(u) + (r^*)^T u \leq (r^*)^T u \leq \|r^*\|.$$

Therefore, since $b^T(r^*/\|r^*\|) = \|r^*\|$, the result is proved. □

3 A Semidefinite Least Squares Formulation of SAT

The formulation SDP(F) can be expressed in the form $\mathcal{A}_c(Y) = b_c$, $\mathcal{A}_s(Y) = b_s$, $Y \succeq 0$, where $\mathcal{A}_c(Y) = b_c$ denotes the m equality constraints representing the clauses:

$$\sum_{J \subseteq I_i} (-1)^{|J|} s_J^i Y_{\emptyset,J} = 0, \quad i = 1, \dots, m,$$

and $\mathcal{A}_s(Y) = b_s$ denotes the equality constraints in the definition of Ω_n:

$$Y_{J,J} = 1, J \in \mathcal{O}, \text{ and } Y_{(I\Delta P),(P\Delta J)} = Y_{I,J}, \ \emptyset \subsetneq P \subseteq \{1,\dots,n\}, I, J \in \mathcal{O}.$$

Motivated by the discussion in Section 2, we consider the SLS problem associated with F:

$$\begin{array}{ll} \min & \|b_c - \mathcal{A}_c(Y)\|^2 \\ \text{s.t.} & \mathcal{A}_s(Y) = b_s \\ & Y \succeq 0, \end{array} \qquad (\text{SLS}_F)$$

Problem (SLS$_F$) requires that the constraints defining the structure of Ω_n be satisfied, and seeks the matrix Y that minimizes the infeasibility of the clause constraints. The SAT instance is satisfiable if and only if the optimal value of (SLS$_F$) is zero. Equivalently, SDP(F) is infeasible if and only if $r_c^* = b_c - \mathcal{A}_c(Y^*) \neq 0$, where Y^* is the optimal solution of problem (SLS$_F$).

Problem (SLS$_F$) minimizes a strictly convex function over a convex set containing at least one positive definite matrix, namely the identity matrix, therefore the Karush-Kuhn-Tucker (KKT) optimality conditions (see e.g. [12]) are necessary and sufficient for optimality, and Y^* and u_s^* are primal and dual optimal if and only if they satisfy

$$\begin{array}{l} \mathcal{A}_c^T(b_c - \mathcal{A}_c(Y^*)) + \mathcal{A}_s^T(u_s^*) \preceq 0 \\ \mathcal{A}_s(Y^*) = b_s \\ (\mathcal{A}_c^T(b_c - \mathcal{A}_c(Y^*)) + \mathcal{A}_s^T(u_s^*)) \bullet Y^* = 0 \\ Y^* \succeq 0. \end{array}$$

Lemma 3.1. *The vector* (r_c^*, u_s^*) *with* $r_c^* \neq 0$ *is an infeasibility certificate for SDP(F).*

Proof. We have to prove that (r_c^*, u_s^*) satisfies conditions (ii) of Lemma 2.1. From the first KKT condition we have that $\mathcal{A}^T(r_c^*, u_s^*) \preceq 0$. Moreover

$$\begin{array}{rcl} b^T(r_c^*; u_s^*) & = & b_c^T r_c^* + b_s^T u_s^* \\ & = & (r_c^* + \mathcal{A}_c(Y^*))^T r_c^* + \mathcal{A}_s(Y^*)^T u_s^* \\ & = & (r_c^*)^T r_c^* + Y^* \bullet \mathcal{A}_c^T(r_c^*) + Y^* \bullet \mathcal{A}_s^T(u_s^*) \\ & = & \|r_c^*\|^2 + \mathcal{A}^T(r_c^*, u_s^*) \bullet Y^* \\ & = & \|r_c^*\|^2 > 0, \end{array}$$

where the last equality follows by the third KKT condition. □

We call $r_c^* > 0$ an *SLS certificate of infeasibility.* Note that each component of r_c^* corresponds to a clause of the SAT instance.

4 Minimal Unsatisfiability

We now turn our attention to exploring how the SDP approach can provide information about minimal unsatisfiability. We state in this section some preliminaries to the main results in Section 5.

A classification of single clauses based on their contribution to the causes of unsatisfiability was proposed in [28]. The highest degree of necessity is given by "necessary clauses", where a clause $C \in F$ is called necessary if every resolution refutation of F must use C.

Definition 4.1. Let F be an unsatisfiable formula. A clause $C \in F$ is said to be *necessary* if and only if there exists a partial assignment satisfying $F \setminus C$.

The corresponding notion of blackundancy is that of clauses which are unnecessary.

Definition 4.2. Let F be an unsatisfiable formula. A clause $C \in F$ is said to be *unnecessary* if and only if $F \setminus C$ remains unsatisfiable. Equivalently, there exist resolution refutations of F that do not use C.

Next we formally introduce minimal unsatisfiability.

Definition 4.3. We say that F is minimal unsatisfiable (MU) if and only if F is unsatisfiable and $F \setminus C$ is satisfiable for any clause C in F.

Whenever we have an unsatisfiable formula, it is clear that it must contain at least one minimal unsatisfiable subformula within it. This motivates the next definition.

Definition 4.4. Let F be an unsatisfiable CNF formula. We say that $G \subseteq F$ is a minimal unsatisfiable sub-formula (MUS) of F if G is minimal unsatisfiable.

For each clause C_i we define the set of truth assignments for which C_i evaluates to false:

$$T_i = T(C_i) = \{S \subseteq V \mid x^S \text{ evaluates false clause } C_i\}.$$

Clearly $|T_i| = 2^{n-|I_i|}$, where I_i is the index set of variables appearing in clause C_i.

The following lemma characterizes (trivially) minimal unsatisfiabilty in terms of the sets $T(C)$.

Lemma 4.1. *The CNF formula F is minimal unsatisfiable if and only if*

$$T(C_i) \nsubseteq \bigcup_{\substack{k=1,\ldots,m \\ k\neq i}} T(C_k), \quad i = 1, \ldots, m \tag{10}$$

$$\bigcup_{k=1}^{m} T(C_k) = \mathcal{P}(V). \tag{11}$$

Corollary 4.1. *Let F be an unsatisfiable CNF formula and $C_i \in F$. Then $F \setminus C_i$ is unsatisfiable if and only if*

$$T(C_i) \subseteq \bigcup_{\substack{k=1,\ldots,m \\ k\neq i}} T(C_k). \tag{12}$$

5 Semidefinite Least Squares Residuals and Minimal Unsatisfiable Formulas

This section presents the main results of this paper.

5.1 Characterization of the Solutions of (SLS$_F$)

Using the expression (6) for $Y \in \Omega_n$, we have

$$q_j(Y) = \sum_{S \subseteq V} \alpha_S q_j(Y^S) = \sum_{S \in T_j} \alpha_S 2^{|I_j|} = 2^{|I_j|} \sum_{S \in T_j} \alpha_S, \tag{13}$$

because

$$q_j(Y^S) = \begin{cases} 2^{|I_j|} & \text{if} \quad Y^S \text{ does not satisfy constraint } j \\ 0 & \text{if} \quad Y^S \text{ satisfies constraint } j. \end{cases}$$

Substituting this expression into the objective function of (SLS$_F$), we have:

$$\|b_c - A_c(Y)\|^2 = \sum_{j=1}^{m} q_j^2(Y) = \sum_{j=1}^{m} \left(2^{|I_j|} \sum_{S \in T_j} \alpha_S \right)^2 = \sum_{j=1}^{m} 4^{|I_j|} \left(\sum_{S \in T_j} \alpha_S \right)^2.$$

Therefore, minimizing $\|b - A(Y)\|^2$ over Ω_n is equivalent to

$$\begin{aligned} \min & \sum_{j=1}^{m} 4^{|I_j|} \left(\sum_{S \in T_j} \alpha_S \right)^2 \\ \text{s.t.} & \sum_{S \subseteq V} \alpha_S = 1 \\ & \alpha_S \geq 0, \ \forall S \subseteq V. \end{aligned} \tag{14}$$

The KKT conditions for (14) are:

$$2 \sum_{j \in M_S} \sum_{J \in T_j} 4^{|I_j|} \alpha_J - \lambda_S - z = 0, \quad S \subseteq V, \tag{15}$$

$$\lambda_S \alpha_S = 0, \quad S \subseteq V, \tag{16}$$

$$\sum_{S \subseteq V} \alpha_S = 1, \tag{17}$$

$$\alpha_S, \lambda_S \geq 0, \quad S \subseteq V, \tag{18}$$

where $M_S = \{j \mid S \in T_j\}$ is the set of clauses falsified by x^S. Moreover, we have the following simple property: $j \in M_S$ if and only if $S \in T_j$.

5.2 Technical lemmas

We present here technical lemmas, with sufficient conditions, that allow us to know if the values of α_S, as a solution of (15)-(18), should be set to zero or not.

The following lemma shows that if x^S falsifies clause C_k and C_i, and there is x^R such that only falsifies clause C_i, then $\alpha_S = 0$.

Lemma 5.1. *Let F be an unsatisfiable CNF formula such that there is $S \in T_k \cap T_i$ with $k \neq i$ and $R \in T_i$ such that $R \notin T_j$ for all $j \neq i$. Let $\alpha = (\alpha_J)_{J \subseteq V}$ be a solution of (14). Then we have $\alpha_S = 0$.*

Proof. We consider the following conditions

$$2\sum_{J\in T_k} 4^{|I_k|}\alpha_J + 2\sum_{J\in T_i} 4^{|I_i|}\alpha_J + 2\sum_{j\in M_S\setminus\{i,k\}}\sum_{J\in T_j} 4^{|I_j|}\alpha_J - \lambda_S - z = 0$$

and for some $R \in T_i$ such that $R \notin T_j$ for all $j \neq i$,

$$2\sum_{J\in T_i} 4^{|I_i|}\alpha_J - \lambda_R - z = 0.$$

Combining these equations we obtain

$$2\sum_{J\in T_k} 4^{|I_k|}\alpha_J + 2\sum_{j\in M_S\setminus\{i,k\}}\sum_{J\in T_j} 4^{|I_j|}\alpha_J - \lambda_S = -\lambda_R. \tag{19}$$

If $\lambda_S \neq 0$, then $\alpha_S = 0$. If $\lambda_S = 0$, then equation (19) is satisfied only if $\lambda_R = 0$ and $\alpha_S = 0$. The result is proved. □

The following lemma is a general version of the previous one. From condition $R \in T_i$ such that $R \notin T_j$ for all $j \neq i$, we have that $M_R = \{i\}$. Since $\{i,k\} \subseteq M_S$, we have $M_R \subset M_S$. This is the sufficient condition of the following lemma.

Lemma 5.2. *Let F be an unsatisfiable CNF formula such that there are $R, S \subseteq V$ satisfying $M_R \subset M_S$. Then $\alpha_S = 0$.*

Proof. We have to use the KKT condition (15) respecting R, S. Thus,

$$2\sum_{j\in M_S}\sum_{J\in T_j} 4^{|I_j|}\alpha_J - \lambda_S = 2\sum_{j\in M_S}\sum_{J\in T_j} 4^{|I_j|}\alpha_J - \lambda_R$$

Since $M_R \subset M_S$, we get

$$2\sum_{j\in M_S\setminus M_R}\sum_{J\in T_j} 4^{|I_j|}\alpha_J\lambda_S = \lambda_S - \lambda_R. \tag{20}$$

We have the following two cases:

- if $\lambda_S \neq 0$, then by complementary $\alpha_S = 0$;
- if $\lambda_S = 0$, then, since the left side of (20) is non-negative and the left side is non-positive, $\alpha_S = 0$.

The result is proved. □

The following lemma says that if x^S only falsifies clause C_k then the correspondent coefficient α_S can not be set to zero.

Lemma 5.3. *Let F be an unsatisfiable CNF formula such that there is $S \in T_k$ with $S \notin T_j$ for all $j \neq k$. Then $\alpha_S \neq 0$.*

Proof. Since $\sum_{J\subseteq V} \alpha_J = 1$ there is $R \in T_i$ such that $\alpha_R \neq 0$. If $R = S$, then there is nothing to prove. Thus, we assume that $R \notin T_k$ (if $R \in T_k$, then by Lemma 5.1 $\alpha_R = 0$). Let us consider the KKT condition (15) with respect to S and R.

$$2 \sum_{J\in T_k} 4^{|I_k|}\alpha_J - \lambda_S = z$$
$$2 \sum_{j\in M_R} \sum_{J\in T_j} 4^{|I_j|}\alpha_J - \lambda_R = z,$$

respectively. Combining these equations we get that

$$2 \sum_{J\in T_k} 4^{|I_k|}\alpha_J - \lambda_S = 2 \sum_{j\in M_R} \sum_{J\in T_j} 4^{|I_j|}\alpha_J - \lambda_R.$$

If $\alpha_J = 0$ for all $J \in T_k$, then

$$2 \sum_{j\in M_R} \sum_{J\in T_j} 4^{|I_j|}\alpha_J = \lambda_R - \lambda_S.$$

Since $\alpha_R \neq 0$, together with $\lambda_R\alpha_R = 0$ implies that $\lambda_R = 0$. Thus we obtain

$$2 \sum_{j\in M_R} \sum_{J\in T_j} 4^{|I_j|}\alpha_J = -\lambda_S,$$

which is impossible. Note that the left-hand side is positive ($\alpha_R \neq 0$) and the right-hand side is non-positive. Therefore, there is $S' \in T_k$ such that $\alpha_{S'} \neq 0$. Let us assume $S' \neq S$. We consider the S' KKT condition,

$$2 \sum_{j\in M_{S'}} \sum_{J\in T_j} 4^{|I_j|}\alpha_J - \lambda_{S'} = z.$$

Thus, we get that

$$2 \sum_{J\in T_k} 4^{|I_k|}\alpha_J - \lambda_S = 2 \sum_{j\in M_{S'}} \sum_{J\in T_j} 4^{|I_j|}\alpha_J - \lambda_{S'}.$$

Simplifying it,

$$-\lambda_S = 2 \sum_{j\in M_{S'}\setminus\{k\}} \sum_{J\in T_j} 4^{|I_j|}\alpha_J - \lambda_{S'}. \tag{21}$$

We consider two cases:

(i) if $\lambda_{S'} \neq 0$ then $\alpha_{S'} = 0$;

(ii) if $\lambda_{S'} = 0$ then, from (21), $\lambda_S = 0$ and $\alpha_{S'} = 0$.

Hence, in any case, $\alpha_{S'} = 0$, which is a contradiction. We conclude that $S = S'$. □

If there is a clause C_i under the assumptions of the previous lemma we deduce that $r_i \neq 0$.

Lemma 5.4. *If F is minimal unsatisfiable, then for each $C_k \in F$, there is $S \in T_k$ such that $\alpha_S \neq 0$. Moreover, $S \notin T_j$ for $j \neq k$.*

Proof. Since F is minimal unsatisfiable, by Lemma 4.1 there is $S \in T_k$ such that $S \notin T_j$ with $j \neq k$. The result follows, applying Lemma 5.3. □

5.3 Clause Weights and their Properties

Let us define a weight for each clause/constraint using the residuals from the SLS problem. We recall from (13) that

$$r_j = q_j(Y) = \sum_{S \subseteq V} \alpha_S q_j(Y^S) = \sum_{S \in T_j} \alpha_S 2^{|I_j|} = 2^{|I_j|} \sum_{S \in T_j} \alpha_S,$$

where I_j is the set of indices of the Boolean variables in clause C_j.

Definition 5.1. For each clause C_j we define the corresponding *weight* $w(C_j)$:

$$w_j = w(C_j) = 2^{|I_j|} r_j.$$

The first result indicates how the weights can be used as a measure of hardness of satisfying a clause. In other words, a necessary clause has weight strictly greater than any unnecessary clause.

Theorem 5.1. *Let F be an unsatisfiable CNF formula and C_i, C_k two clauses of F such that $F \setminus C_i$ is satisfiable and $F \setminus C_k$ is unsatisfiable. We have*

$$w_k < w_i.$$

Proof. There is at least a point Y^R such that $R \in T_i$ and $R \notin T_j$ for $j \neq i$. For $R \in T_i$ and $S \in T_k$ we consider the following KKT conditions

$$2 \sum_{j \in M_S} \sum_{J \in T_j} 4^{|I_j|} \alpha_J - \lambda_S = z$$
$$2 \sum_{j \in M_R} \sum_{J \in T_j} 4^{|I_j|} \alpha_J - \lambda_R = z.$$

From here, we obtain

$$2 \sum_{j \in M_S} \sum_{J \in T_j} 4^{|I_j|} \alpha_J - \lambda_S = 2 \sum_{j \in M_R} \sum_{J \in T_j} 4^{|I_j|} \alpha_J - \lambda_R,$$

which is equivalent to

$$2 \sum_{J \in T_k} 4^{|I_k|} \alpha_J + 2 \sum_{j \in M_S \setminus \{k\}} \sum_{J \in T_j} 4^{|I_j|} \alpha_J - \lambda_S = 2 \sum_{J \in T_i} 4^{|I_i|} \alpha_J - \lambda_R,$$

because $R \notin T_j$ for all $j \neq i$. By Lemma 5.3 $\alpha_R \neq 0$, which implies that $\lambda_R = 0$. Thus,

$$2 \sum_{J \in T_k} 4^{|I_k|} \alpha_J + 2 \sum_{j \in M_S \setminus \{k\}} \sum_{J \in T_j} 4^{|I_j|} \alpha_J - \lambda_S = 2 \sum_{J \in T_i} 4^{|I_i|} \alpha_J.$$

If $\alpha_J = 0$ for all $J \in T_k$ clearly the result follows, because $\sum_{J \in T_k} 4^{|I_k|} \alpha_J = 0$. If there is $S \in T_k$ such that $\alpha_S \neq 0$, then $\lambda_S = 0$. Hence, we obtain

$$2 \sum_{J \in T_k} 4^{|I_k|} \alpha_J + 2 \sum_{j \in M_S \setminus \{k\}} \sum_{J \in T_j} 4^{|I_j|} \alpha_J = 2 \sum_{J \in T_i} 4^{|I_i|} \alpha_J,$$

which implies that

$$2 \sum_{J \in T_k} 4^{|I_k|} \alpha_J < 2 \sum_{J \in T_i} 4^{|I_i|} \alpha_J.$$

The result follows from the definition of the weights in Definition 5.1. $\square$

We illustrate by example the theorem above.

Example 5.1. Consider be the following SAT instance F with 3 variables and 7 clauses:

$$F = \begin{cases} c_1 & : & x_1 \vee x_2 \\ c_2 & : & \bar{x}_2 \vee x_3 \\ c_3 & : & \bar{x}_1 \vee x_2 \\ c_4 & : & \bar{x}_2 \vee \bar{x}_3 \\ c_5 & : & x_1 \vee \bar{x}_2 \\ c_6 & : & \bar{x}_1 \vee \bar{x}_2 \\ c_7 & : & x_2 \vee x_3 \end{cases}$$

The possible truth assignments are shown in the following table:

J	x_1	x_2	x_3	M_J
1	1	1	1	{4,6}
2	1	1	-1	{2,6}
3	1	-1	1	{3}
4	1	-1	-1	{3,7}
5	-1	1	1	{4,5}
6	-1	1	-1	{2,5}
7	-1	-1	1	{1}
8	-1	-1	-1	{1,7}

Let us index the sets $J \subseteq V$ by $1, 2, \ldots, 2^3$. We have

$$T_1 = \{7, 8\},\ T_2 = \{2, 6\},\ T_3 = \{3, 4\},\ T_4 = \{1, 5\},\ T_5 = \{5, 6\},\ T_6 = \{1, 2\}, T_7 = \{4, 8\}.$$

The KKT conditions for SDP(F) are

$$\begin{cases} 32(\alpha_1 + \alpha_5) + 32(\alpha_1 + \alpha_2) - \lambda_1 = z \\ 32(\alpha_1 + \alpha_2) + 32(\alpha_2 + \alpha_6) - \lambda_2 = z \\ 32(\alpha_3 + \alpha_4) - \lambda_3 = z \\ 32(\alpha_3 + \alpha_4) + 32(\alpha_4 + \alpha_8) - \lambda_4 = z \\ 32(\alpha_1 + \alpha_5) + 32(\alpha_5 + \alpha_6) - \lambda_5 = z \\ 32(\alpha_2 + \alpha_6) + 32(\alpha_5 + \alpha_6) - \lambda_6 = z \\ 32(\alpha_7 + \alpha_8) - \lambda_7 = z \\ 32(\alpha_7 + \alpha_8) + 32(\alpha_4 + \alpha_8) - \lambda_8 = z \\ \alpha_i \lambda_i = 0, \quad i = 1, \ldots, 8 \\ \displaystyle\sum_{i=1}^{8} \alpha_i = 1. \end{cases}$$

and a possible solution for the KKT system is given by

$$
\begin{aligned}
&\alpha_1 = \alpha_2 = \alpha_5 = \alpha_6 = \frac{1}{12}, \\
&\alpha_3 = \alpha_7 = \frac{1}{3}, \\
&\alpha_4, \alpha_8 = 0, \\
&\lambda_i = 0, \quad i = 1, \dots, 8, \\
&z = \frac{32}{3}.
\end{aligned}
$$

Note that this solution is not unique; for example, $\alpha_1 = \alpha_4 = \alpha_6 = \alpha_8 = 0$, $\alpha_2 = \alpha_5 = \frac{1}{6}$ and $\alpha_3 = \alpha_7 = \frac{1}{3}$ is also a solution. However, the residual components are the same for different solutions of the KKT system. The residual components are

$$
\begin{aligned}
r_1 &= 4(\alpha_7 + \alpha_8) = 4\left(\frac{1}{3} + 0\right) = \frac{4}{3} \\
r_2 &= 4(\alpha_2 + \alpha_6) = 4\left(\frac{1}{12} + \frac{1}{12}\right) = \frac{2}{3} \\
r_3 &= 4(\alpha_3 + \alpha_4) = 4\left(\frac{1}{3} + 0\right) = \frac{4}{3} \\
r_4 &= 4(\alpha_1 + \alpha_5) = 4\left(\frac{1}{12} + \frac{1}{12}\right) = \frac{2}{3} \\
r_5 &= 4(\alpha_5 + \alpha_6) = 4\left(\frac{1}{12} + \frac{1}{12}\right) = \frac{2}{3} \\
r_6 &= 4(\alpha_1 + \alpha_2) = 4\left(\frac{1}{12} + \frac{1}{12}\right) = \frac{2}{3} \\
r_7 &= 4(\alpha_4 + \alpha_8) = 0.
\end{aligned}
$$

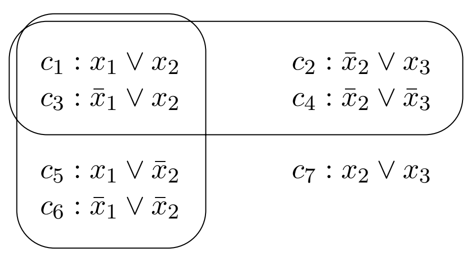

Figure 1: MUSs of F

In Figure 5.1 we identify MUSs of F. If we remove clause c_1 or c_3 we obtain a satisfiable sub-formula. However, if we remove one of the other clauses, the sub-formula obtained is still unsatisfiable. Note that $w_1, w_3 > w_i$ with $i \neq 1, 3$. This follows Theorem 5.1. Moreover, c_7 does not belong to any MUS of F and we have $w(C_7) = 0$, accordingly with Theorem 5.3.

Each set $\{T_1, T_2, T_3, T_4\}$ and $\{T_1, T_3, T_5, T_6\}$ defines a MUS of F (see Lemma 4.1). In this sense each MUS defines a covering of the truth assignments.

The second result is that for a minimal unsatisfiable formula, all the clauses have the same weight.

Theorem 5.2. *Let F be an unsatisfiable CNF formula. If F is minimal unsatisfiable, then*

$$w_i = w_k, \quad \forall i, k \in \{1, \dots, m\}.$$

Proof. Let C_i, C_k two clauses in F. Let $R \in T_i$ and $R \notin T_j$ for all $j \neq i$ and $S \in T_k$ and $S \notin T_j$ for all $j \neq i$. From KKT conditions, (15), we obtain

$$\sum_{j \in M_S} \sum_{J \in T_j} 4^{|I_j|} \alpha_J - \lambda_S = \sum_{j \in M_R} \sum_{J \in T_j} 4^{|I_j|} \alpha_J - \lambda_R,$$

which is equivalent to

$$\sum_{J \in T_k} 4^{|I_k|} \alpha_J + \sum_{j \in M_S \setminus \{k\}} \sum_{J \in T_j} 4^{|I_j|} \alpha_J - \lambda_S = \sum_{J \in T_i} 4^{|I_i|} \alpha_J + \sum_{j \in M_R \setminus \{i\}} \sum_{J \in T_j} 4^{|I_j|} \alpha_J - \lambda_R.$$

Since $R \notin T_j$ for $j \neq i$ and $S \notin T_j$ for $j \neq k$, this gives

$$\sum_{J \in T_k} 4^{|I_k|} \alpha_J - \lambda_S = \sum_{J \in T_i} 4^{|I_i|} \alpha_J - \lambda_R.$$

Applying Lemma 5.4, $\alpha_S, \alpha_R \neq 0$, which implies $\lambda_S = \lambda_R = 0$. Thus

$$\sum_{J \in T_k} 4^{|I_k|} \alpha_J = \sum_{J \in T_i} 4^{|I_i|} \alpha_J.$$

Hence, $w(C_i) = w(C_k)$ and the result is proved. $\square$

Note that the converse of Theorem 5.2 does not hold in general. This is shown by the following example.

Example 5.2. Let F be the following SAT instance:

$$F = \begin{cases} c_1 & : & x_1 \\ c_2 & : & \bar{x}_2 \\ c_3 & : & x_3 \\ c_4 & : & \bar{x}_1 \vee x_2 \\ c_5 & : & x_2 \vee \bar{x}_3 \\ c_6 & : & \bar{x}_1 \vee \bar{x}_3. \end{cases}$$

In this example, we illustrate that the converse of Theorem 5.2 is not valid in general, namely that having all equal weights for the clauses does not imply that the CNF formula is minimal unsatisfiable.

We list below the possible truth assignments:
For simplicity of the notation we indexed the sets $J \subseteq V$ to numbers $1, 2, \dots, 2^3$. We have

$$T_1 = \{5, 6, 7, 8\},\ T_2 = \{1, 2, 5, 6\},\ T_3 = \{2, 4, 6, 8\},\ T_4 = \{3, 4\},\ T_5 = \{3, 7\},\ T_6 = \{1, 3\}.$$

J	x_1	x_2	x_3	M_J
1	1	1	1	{2,6}
2	1	1	-1	{2,3}
3	1	-1	1	{4,5,6}
4	1	-1	-1	{3,4}
5	-1	1	1	{1,2}
6	-1	1	-1	{1,2,3}
7	-1	-1	1	{1,5}
8	-1	-1	-1	{1,3}

The KKT conditions for the instance SDP(F) are

$$\begin{cases} 8(\alpha_1+\alpha_2+\alpha_5+\alpha_6)+32(\alpha_1+\alpha_3)-\lambda_1 = z \\ 8(\alpha_1+\alpha_2+\alpha_5+\alpha_6)+8(\alpha_2+\alpha_4+\alpha_6+\alpha_8)-\lambda_2 = z \\ 32(\alpha_3+\alpha_4)+32(\alpha_3+\alpha_7)+32(\alpha_1+\alpha_3)-\lambda_3 = z \\ 8(\alpha_2+\alpha_4+\alpha_6+\alpha_8)+32(\alpha_3+\alpha_4)-\lambda_4 = z \\ 8(\alpha_5+\alpha_6+\alpha_7+\alpha_8)+8(\alpha_1+\alpha_2+\alpha_5+\alpha_6)-\lambda_5 = z \\ 8(\alpha_5+\alpha_6+\alpha_7+\alpha_8)+8(\alpha_1+\alpha_2+\alpha_5+\alpha_6)+8(\alpha_2+\alpha_4+\alpha_6+\alpha_8)-\lambda_6 = z \\ 8(\alpha_5+\alpha_6+\alpha_7+\alpha_8)+32(\alpha_3+\alpha_7)-\lambda_7 = z \\ 8(\alpha_5+\alpha_6+\alpha_7+\alpha_8)+32(\alpha_3+\alpha_7)-\lambda_8 = z \\ \alpha_i\lambda_i = 0, \quad i = 1,\ldots,8 \\ \displaystyle\sum_{i=1}^{8}\alpha_i = 1. \end{cases}$$

We have as a possible solution for the KKT system,

$$\begin{aligned} &\alpha_1 = \alpha_4 = \alpha_7 = \frac{2}{15}, \\ &\alpha_2 = \alpha_5 = \alpha_8 = \frac{3}{15}, \\ &\alpha_3 = \alpha_6 = 0, \\ &\lambda_3 = \lambda_6 = \frac{64}{15}, \quad \lambda_i = 0, \quad i \neq 3,6, \\ &z = \frac{128}{15}. \end{aligned}$$

The residual components are

$$\begin{aligned}
r_1 &= 2(\alpha_5+\alpha_6+\alpha_7+\alpha_8) = 2\left(\frac{3}{15}+0+\frac{2}{15}+\frac{3}{15}\right) = \frac{16}{15}\\
r_2 &= 2(\alpha_1+\alpha_2+\alpha_5+\alpha_6) = 2\left(\frac{2}{15}+\frac{3}{15}+\frac{3}{15}+0\right) = \frac{16}{15}\\
r_3 &= 2(\alpha_2+\alpha_4+\alpha_6+\alpha_8) = 2\left(\frac{3}{15}+\frac{2}{15}+0+\frac{3}{15}\right) = \frac{16}{15}\\
r_4 &= 4(\alpha_3+\alpha_4) = 4\left(0+\frac{2}{15}\right) = \frac{8}{15}\\
r_5 &= 4(\alpha_3+\alpha_7) = 4\left(0+\frac{2}{15}\right) = \frac{8}{15}\\
r_6 &= 4(\alpha_1+\alpha_3) = 4\left(\frac{2}{15}+0\right) = \frac{8}{15}.
\end{aligned}$$

Thus, weights of each clause are

$$w_1 = w_2 = w_3 = w_4 = w_5 = w_6 = \frac{32}{15}.$$

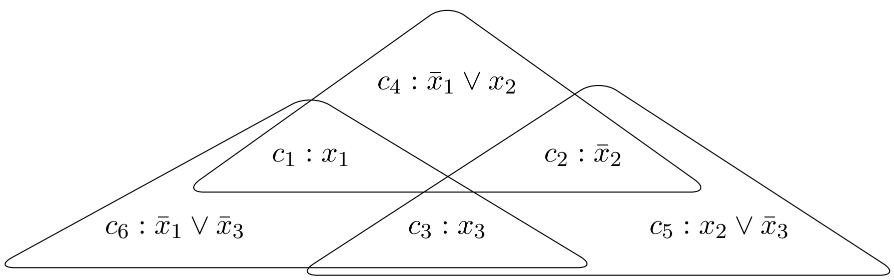

Figure 2: MUSs of F

The third result is that if a clause does not belong to any MUS, then its corresponding residual, and hence its weight, is zero. To prove this, we first prove the following propositions.

Proposition 5.1. *For any minimal unsatisfiable sub-formula G of F, $C_k \notin G$ if and only if for any $J \in T(C_k)$ there is $S \notin T(C_k)$ such that $M_S \subset M_J$.*

Proof. We assume that for any $J \in T_k$ there is $S \notin T_k$ such that $M_S \subset M_J$. Any clause falsified by x^S is also falsified by x^J. Thus, to form a MUS of F, we have to choose some clause C_j with $j \in M_S$. But this clause is also falsified by x^J, therefore C_k is never chosen to form a MUS.

Now, we prove the opposite direction. Assume that there is $J \in T_k$ for all $S \notin T_k$ such that $M_S \not\subseteq M_J$. By construction $M_S \neq M_J$, because $k \in M_J$ but $k \notin M_S$. Thus, we can form MUS of F, containing C_k, in the following way:

(i) for each $S \in \mathcal{P}(V) \setminus T_k$ we choose a clause indexed by $j_S \in M_S$ such that $j_S \notin M_J$;

(ii) if j_S have been chosen before, we move to the next point x^S, with $S \in \mathcal{P}(V) \setminus T_k$;

(iii) let G' be the set of chosen clauses. Clearly, at least the point x^J with $J \in T_k$ satisfies G';

(iv) let $G = G' \cup \{C\}$;

(v) G is unsatisfiable and if it is not minimal, we just can remove clauses from G' until G is minimal unsatisfiable.

The result is proved. □

Proposition 5.2. *If for any $J \in T(C)$ there is $S \notin T(C)$ such that $M_S \subset M_J$ then $w(C) = 0$.*

Proof. The proposition follows by Lemma 5.2. □

Theorem 5.3. *Let F be an unsatisfiable CNF formula and C a clause in F. If C is not in the largest UMU of F, then $w(C) = 0$.*

Proof. It follows by Propositions 5.2 and 5.1. □

Example 5.1 illustrates the result of Theorem 5.3. Note that the question of whether the converse of Theorem 5.3 holds remains open.

6 Application to an Example from [25]

We conclude with the application of our results to an instance from [25].

The propositional formula is

$$F = \left\{ \begin{array}{rcl} c_0 & : & x_4 \\ c_1 & : & x_2 \vee x_3 \\ c_2 & : & x_1 \vee x_2 \\ c_3 & : & x_1 \vee \bar{x}_3 \\ c_4 & : & \bar{x}_2 \vee \bar{x}_5 \\ c_5 & : & \bar{x}_1 \vee \bar{x}_2 \\ c_6 & : & x_1 \vee x_5 \\ c_7 & : & \bar{x}_1 \vee \bar{x}_5 \\ c_8 & : & x_2 \vee x_5 \\ c_9 & : & \bar{x}_1 \vee x_2 \vee \bar{x}_3 \\ c_{10} & : & \bar{x}_1 \vee x_2 \vee \bar{x}_4 \\ c_{11} & : & x_1 \vee \bar{x}_2 \vee x_3 \\ c_{12} & : & x_1 \vee \bar{x}_2 \vee \bar{x}_4 \end{array} \right.$$

Using Lemma 5.2 we obtain

$$\alpha_i = 0, \ \forall i \in \Lambda_0 = \{1, 3, 4, 5, 7, 8, 26, 27, 28, 30, 31, 32\}.$$

To guarantee that complementary condition is satisfied we assume that $\lambda_i = 0$ for $i \notin \Lambda_0$. With this assumption, and using the KKT conditions, we deduce that

$$\begin{aligned}
w_4 &= w_6 = w_7 = w_8, \\
w_0 &= w_{10} = w_{12}, \\
w_1 &= w_3 = w_9 = w_{11}, \\
w_2 &= w_0 + w_6, \\
w_5 &= w_2 + w_3.
\end{aligned}$$

Note that our approach generates a set of equations on the weights that identifies the same pattern for MUSs as identified by algorithm HYCAM [25] and depicted in Figure 6. Specifically, the equations show that:

- for each area of the diagram in Figure 3 that does not represent an intersection, the weights of the clauses in that area will be equal. Specifically for the example: the weights of c_4, c_6, c_7 and c_8 are equal, as are the weights of c_0, c_{10} and c_{12}, and those of c_1, c_3, c_9 and c_{11}.
- for the areas that represent intersections, the weight of the clause in the intersection is equal to the sum of weights obtained by taking one clause from the non-intersecting part of each of the areas that form the intersection. Specifically for the example:
  - clause c_2 is in the intersection of two MUSs, and one of our equations states that $w_2 = w_0 + w_6$, i.e., the weight of c_2 is equal to the sum of the weight of c_0 and the weight of c_6; but as observed earlier, c_0 could be replaced by c_{10} or c_{12}, and c_6 could be replaced by c_4, c_7 or c_8.
  - clause c_5 is in the intersection of the three MUSs, and $w_5 = w_2 + w_3$ so that its weight is equal to the weight of c_3 (or any one of c_1, c_9, c_{11}) plus the weight of c_2, which in turn is the sum of two weights as observed already. The conclusion is that the weight of c_5 is equal to the sum of three weights, one from each of the areas forming the intersection where c_5 lies.

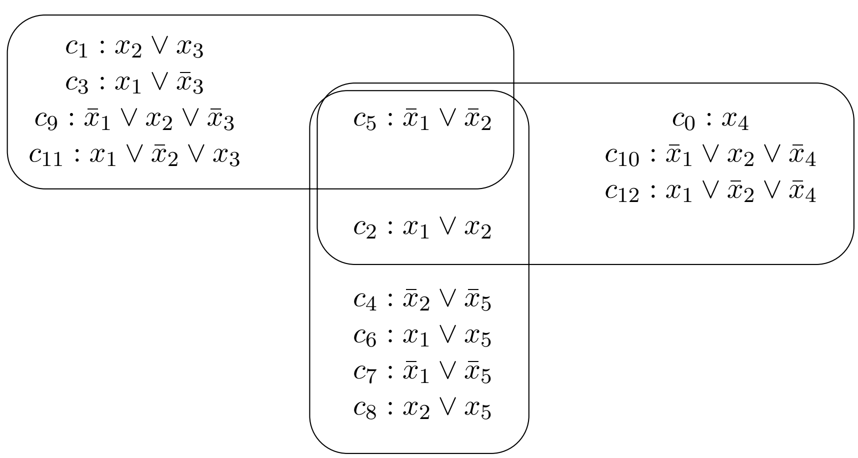

Figure 3: MUSs of F

7 Conclusion and Future Research

In this paper we showed how a semidefinite least squares problem can be used to associate weights to the clauses in a propositional formula. We then showed that the weight of a necessary clause is strictly greater than the weight of any unnecessary clause. We also showed that if a formula is minimal unsatisfiable, then all of its clauses have the same weight, and that if a clause does not belong to any minimal unsatisfiable formula, then its weight is zero. As mentioned in the Introduction, the optimization problem (14) is a convex optimization problem with a special structure that may be exploited in the design of a computational algorithm. This is a promising direction for future research.

Another contribution of this paper is the consideration of how the infeasibility of a semidefinite optimization problem can be tested using a semidefinite least squares problem, and a discussion of the connection between the SLS problem and Farkas' Lemma for semidefnite optimization. Future research in this direction could look into the potential application of the SLS approach as described in Section 3 for SAT to other combinatorial problems where the understanding of why solutions of a certain type may or may not exist.

Acknowledgements

The authors thank the two anonymous reviewers for their thorough and insightful critiques and suggestions that helped us in improving this paper.

References

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

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

[3] 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.

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

[5] M.F. Anjos. Progress in semidefinite optimization techniques for satisfiability. In A. Ridha Mahjoub, editor, *Progress in Combinatorial Optimization*, pages 489–520. Wiley, Oxford, 2011.

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

[8] M.F. Anjos and M.V.C. Vieira. Semidefinite resolution and exactness of semidefinite relaxations for satisfiability. *Discrete Applied Mathematics*, 161:2812–2826, 2013.

[9] 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.

[10] 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.

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

[12] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, 2004.

[13] R. Bruni. On exact selection of minimally unsatisfiable subformulae. *Ann. Math. Artif. Intell.*, 43(1):35–50, 2005.

[14] H.K. Büning and O. Kullmann. Minimal unsatisfiability and autarkies. In *Handbook of Satisfiability*, pages 339–401. 2009.

[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] G. Davydov and I. Davydova. Tautologies and positive solvability of linear homogeneous systems. *Ann. Pure Appl. Logic*, 57(1):27–43, 1992.

[18] A. Dax. Classroom note: An elementary proof of farkas' lemma. *SIAM review*, 39(3):503–507, 1997.

[19] E. de Klerk. *Aspects of Semidefinite Programming*, volume 65 of *Applied Optimization*. Kluwer Academic Publishers, Dordrecht, 2002.

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

[21] 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.

[22] M.J. García de la Banda, P.J. Stuckey, and J. Wazny. Finding all minimal unsatisfiable subsets. In: *Proceedings of the 5th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, 27-29 August 2003, Uppsala, Sweden*, pages 32–43. ACM, 2003.

[23] A. Deza, E. Nematollahi, and T. Terlaky. How good are interior point methods? Klee-Minty cubes tighten iteration-complexity bounds. *Math. Program.*, 113(1, Ser. A):1–14, 2008.

[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] E. Grégoire, B. Mazure, and C. Piette. Using local search to find MSSes and MUSes. *Eur. J. Oper. Res.*, 199(3):640–646, 2009.

[26] C. Helmberg. *Semidefinite Programming for Combinatorial Optimization.* Habilitationsschrift, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2000. `http://opus4.kobv.de/opus4-zib/files/602/ZR-00-34.pdf`.

[27] O. Kullmann. Lean clause-sets: generalizations of minimally unsatisfiable clause-sets. *Discrete Applied Mathematics*, 130(2):209–249, 2003.

[28] O. Kullmann, I. Lynce, and J. Marques-Silva. Categorisation of clauses in conjunctive normal forms: Minimally unsatisfiable sub-clause-sets and the lean kernel. In: *Theory and Applications of Satisfiability Testing - SAT 2006*, pages 22–35.

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

[30] M.H. Liffiton, M.N. Mneimneh, I. Lynce, Z.S. Andraus, J. Marques-Silva, and K.A. Sakallah. A branch and bound algorithm for extracting smallest minimal unsatisfiable subformulas. *Constraints*, 14(4):415–442, 2009.

[31] M.H. Liffiton and K.A. Sakallah. On finding all minimally unsatisfiable subformulas. In: F. Bacchus and T. Walsh (eds), *Theory and Applications of Satisfiability Testing, 8th International Conference, SAT 2005, St. Andrews, UK, June 19-23, 2005, Proceedings*, volume 3569 of *Lecture Notes in Computer Science*, pages 173–186. Springer, 2005.

[32] M.H. Liffiton and K.A. Sakallah. Algorithms for computing minimal unsatisfiable subsets of constraints. *J. Autom. Reasoning*, 40(1):1–33, 2008.

[33] S. Ma and D. Liang. A polynomial-time algorithm for blackucing the number of variables in MAX SAT problem. *Science in China (Series E)*, 40(3):301–311, June 1997.

[34] C.H. Papadimitriou and D. Wolfe. The complexity of facets resolved. *Journal of Computer and System Sciences*, 37(1):2–13, August 1988.

[35] 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, Berlin, 2005.

[36] H. van Maaren, L. van Norden, and M. J. H. Heule. Sums of squares based approximation algorithms for MAX-SAT. *Discrete Appl. Math.*, 156(10):1754–1779, 2008.

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

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