Les Cahiers du GERAD ISSN: 0711–2440

On semidefinite least squares and minimum unsatisfiable subformulas

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

G–2014–33

May 2014

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

On semidefinite least squares and minimum unsatisfiable subformulas

Miguel F. Anjos[a]

Manuel V.C. Vieira[b]

[a] *GERAD, Canada Research Chair in Discrete Nonlinear Optimization in Engineering & Polytechnique Montréal, Montréal (Québec) Canada H3C 3A7*

[b] *Departamento de Matemática, Faculdade de Ciências e Tecnologia & CMA, Universidade Nova de Lisboa, Portugal*

anjos@stanfordalumni.org
mvcv@fct.unl.pt

May 2014

Les Cahiers du GERAD

G–2014–33

Abstract: This paper provides new results on the application of semidefinite optimization to satisfiability by studying the connection between semidefinite optimization and minimal unsatisfiable subformulas. We use a semidefinite least squares problem to assign weights to the clauses of a propositional formula. We then show that these weights are a measure of the importance of each clause in rendering the formula unsatisfiable. In particular, we show that this approach identifies two important cases: first, if an unsatisfiable formula is a minimal unsatisfiable subformula, 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. The results in this paper establish the basis for a semidefinite optimization-based algorithm to identify minimal unsatisfiable subformulas within a given formula. 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.

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 CMA/FCT/UNL, under the project PEst-OE/MAT/UI0297/2013.

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 [8]. 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 [13], and it is in general a challenging problem to detect that a SAT instance is unsatisfiable and to provide insight into its unsatisfiability.

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 [24] and Blair et al. [9] 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 [11]. For certain classes of instances, including Horn formulas and their generalizations, the exactness of the linear programming relaxation has been established, see e.g. [10].

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 [25] 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 revolutionary applications of SDP is due to Goemans and Williamson who proposed SDP-based polynomial-time approximation schemes for MAX-CUT and MAX-2-SAT [18]. Further research since this breakthrough has deepened the connections between SDP and the SAT and MAX-SAT problems, see the recent survey chapter [5].

The SDP approach is a useful tool in the study of unsatisfiability because it is possible to write an *exact explicit SDP relaxation* for every SAT instance. An SDP relaxation of a SAT instance is said to be exact if it solves the instance in the sense that the 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 somewell-known classes of SAT instances. Subsequent papers by Anjos [1, 2, 3, 4] and van Maaren et al. [22, 23] proposed several SDP relaxations for different versions of SAT, and some exactness results for particularly structured SAT formulas. The exact explicit 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 [21]. 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. Moreover, this line of argument gives no understanding of the deeper connections between SAT and SDP.

A more direct connection between SAT and SDP was given in our recent paper [7] 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 [7] 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 unsatisfiable subformulas. A minimal unsatisfiable subformula (MUS) is a subset of the clauses of a SAT formula with the property that it is unsatisfiable but if any clause in the subset is removed then the rest of the subset becomes satisfiable. We establish a connection between SDP and MUSes by using a semidefinite least squares (SLS) problem to associate weights to the clauses in a SAT 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 a MUS, then all of its clauses have the same weight; second, if a clause does not belong to any MUS, then its weight is zero. These results establish the basis for an SDP-based algorithm to identify MUSes in a general SAT formula; in particular such an algorithm would in principle be able to certifiably find the smallest MUS, i.e., the MUS with a globally minimum number of clauses.

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 [14] for linear optimization. We also explain the connection between the SLS problem and a semidefinite version of the well-known Farkas' Lemma.

This paper is structured 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 minimal unsatisfiable subformula (MUS) and prove a new characterization of MUSes. 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 minimal unsatisfiable, 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 variables $x_1, \dots, x_n$ 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 $I_j, \bar{I}_j \subseteq \{1, \dots, n\}$, $I_j \cap \bar{I}_j =$, 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 $x_1 \dots, x_n$ such that Φ evaluates to TRUE?

We use the standard formulation for an SDP problem as follows:

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

where $C \bullet X := \operatorname{tr}(C\,X)$ and all matrices are $n \times n$ and symmetric. We may write the equality constraints of (1) as $\mathcal{A}(X) = b$, where $\mathcal{A}(X) = (A_1 \bullet X, \dots, A_m \bullet X)^T$.

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

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

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 $I \in \mathcal{P}(V)$. 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 $I, J \subseteq V$. Furthermore, the matrix Y satisfies:

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

We define the following set which gathers all these constraints:

$$\Omega_n = \{Y \succeq 0 | Y_{(I\Delta P),(P\Delta J)} = Y_{I,J},\ \forall \emptyset \subsetneq P \subseteq V \text{ and } y_\emptyset = 1\}. \tag{3}$$

An exact relaxation can be constructed using the set Ω_n as presented in the following definition.

Definition 1.1 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

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

Lasserre's Theorem [21, Theorem 3.2] implies that SDP(F) is feasible if and only if F is satisfiable.

2 A certificate of SDP infeasibility via least-squares

A certificate of infeasibility for SDP(F) proves that F is unsatisfiable. This follows by construction of the formulation SDP(F) whose feasible set contains all truth assignments of F. Clearly, if the feasible set of SDP(F) is empty, then F is unsatisfiable. In this section, we explore a characterization of infeasibility via a least squares problem.

Let x^S, with $S \subseteq V$ denote the 2^n truth assignments of a CNF formula,

$$x^S_1 = \begin{cases} 1 & \text{if} \quad 1 \in S \\ -1 & \text{if} \quad 1 \notin S, \end{cases}$$

and

$$x^S_k = \begin{cases} -x^S_{k-1} & \text{if} \quad k \in S \\ x^S_{k-1} & \text{if} \quad k \notin S \end{cases}$$

Remark 2.1 We have $x^S \neq x^R$ for all $S \neq R$. If p is the first element such that $p \in S$ and $p \notin R$, then $x^S_{p+1} = -x^S_p$, $x^R_{p+1} = x^R_p$ and $x^R_p = x^S_p$.

The vectors x^S, $S \subseteq V$ are the vertices of the cube $[-1,1]^n$. We can lift these vertices to the space $\mathbb{R}^{\mathcal{P}(V)}$ in the following way:

$$y^S_\emptyset = 1,\ y^S_{\{k\}} = x^S_k, \text{ and } y^S_J = \prod_{i\in J} x^S_i.$$

We can also lift the vectors $y^S \in \mathbb{R}^{\mathcal{P}(V)}$ to the space $\mathbf{S}^{2^n}$ by setting

$$Y^S = (y^S)(y^S)^T \quad \forall S \subseteq V.$$

The matrices Y^S are the vertices of the set Ω_n.

In the following, we prove that SDP(F) is either feasible or strongly infeasible. We refer the reader to [15] for more details about infeasibility of an SDP problem.

Definition 2.1 The SDP problem (1) is strongly infeasible if the feasible set is empty and there exists $\epsilon > 0$ such that for every $X \succeq 0$ there exists $i \in \{1, \ldots, m\}$ satisfying

$$|A_i \bullet X - b_i| > \epsilon.$$

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.,

$$Y = \sum_{S \subseteq V} \alpha_S Y^S.$$

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. $\square$

An immediate conclusion is that if SDP(F) is not feasible, then there exists a dual improving direction. Let A_i, $i = 1, \ldots, m$ be $n \times n$ real symmetric matrices and b a nonzero m-vector. Then Farkas' Lemma [20, Lemma 2.2.4] says:

Lemma 2.1 *Suppose that $\{\mathcal{A}(X) : X \succeq 0\}$ is closed. 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 a SLS problem. This is an extension of the theorem of Dax [14] for linear programming. The idea is that determining which of the two systems has a solution can be answered by considering the bounded least squares problem:

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

where $\| \; \|$ denotes the Euclidean norm. The minimum is attained if the set $\{\mathcal{A}(X) \,|X \succeq 0\}$ is closed, i.e., under the assumption of Lemma 2.1. For $X \in \mathbf{S}^n$, define the corresponding residual vector as $r(X) = b - \mathcal{A}(X)$. Under the assumption of Lemma 2.1, we have the following result.

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

$$X^* \succeq 0, \quad \mathcal{A}^T(r^*) \preceq 0, \quad \text{and} \quad X^* \bullet \mathcal{A}^T(r^*) = 0. \tag{4}$$

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

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

where $U_i = u_i u_i^T$ and $u_i \in \mathbb{R}^n$ is the unitary eigenvector associated to the eigenvalue $\lambda_i(X^*)$. Clearly, $\theta = 0$ solves the problem

$$\begin{array}{ll} \min & f_i(\theta) \\ \text{s.t.} & \lambda_i(X^*) + \theta \geq 0. \end{array} \tag{5}$$

Therefore, since $f_i'(0) = -2\mathcal{A}(U_i)^T r^*$,

$$\begin{aligned} \lambda_i(X^*) > 0 &\implies \mathcal{A}(U_i)^T r^* = 0, \\ \lambda_i(X^*) = 0 &\implies -\mathcal{A}(U_i)^T r^* \geq 0. \end{aligned}$$

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 $U \in \mathbf{S}^n_+$. 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 $Z \in \mathbf{S}^n_+$. Hence $\mathcal{A}^T(r^*) \preceq 0$.

Conversely, we assume that (4) holds and let $Z \succeq 0$. Let $U \in \mathbf{S}^n$ 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 (4) 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) *Let* X^* *solve* (LSQ) *and let* $r^* = b - \mathcal{A}(X^*)$ *denote the corresponding residual vector. If* $r^* = 0$*, then* X^* *solves (i) of Lemma 2.1. Otherwise,* r^* *solves (ii) and* $b^T r^* = \|r^*\|$.

Clearly $r^* \neq 0$ is a certificate of primal infeasibility.

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*

$$\begin{array}{ll} \max & b^T u \\ \text{s.t.} & \mathcal{A}^T(u) \preceq 0 \\ & \|u\| = 1. \end{array} \tag{6}$$

Proof. Let $u \in \mathbb{R}^n$ be a feasible point of (6). 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

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

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 $\mathcal{A}_c(X) = b_c,\ X \in \Omega_n$, where $\mathcal{A}_c(X) = b_c$ are the m equality constraints representing the clauses. Since Ω_n is the convex hull of a finite number of points, we have that is closed. Thus, the SDP formulation for a propositional formula F satisfies the constraint qualification for Farkas' Lemma.

Motivated by the discussion in Section 2, we consider the SLS problem

$$\begin{array}{ll} \min & \|b_c - \mathcal{A}_c(X)\|^2 \\ \text{s.t.} & X \in \Omega_n. \end{array}$$

This can be written as

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

where the constraints $A_s(X) = b_s$ enforce the matrix structure of X, i.e., $\mathcal{A}_s(X) = b_s$ represents all the linear constraints in the representation (3) of Ω_n.

Because satisfiability is determined by the clauses, we focus on the residuals of the constraints representing clauses. Problem (LSQa) therefore requires that the constraints defining the structure of Ω_n be satisfied, and seeks the matrix X that minimizes the infeasibility of the clauses constraints. The SAT instance is satisfiable if and only if the optimal value of (LSQa) is zero. Equivalently, the next proposition formally states that SDP(F) is infeasible if and only if $r_c^* = b_c - \mathcal{A}_c(X_s^*) \neq 0$, where X_s^* solves the problem (LSQa).

Proposition 3.1 *Let X^* be a solution of (LSQ) and $r^* = b - \mathcal{A}(X^*)$. Let X_s^* be a solution of (LSQa) and $r_c^* = b_c - \mathcal{A}_c(X_s^*)$. We have $r^* = 0$ if and only if $r_c^* = 0$.*

Problem (LSQa) minimizes a strictly convex function over a convex set, and as discussed before, it has a strictly feasible solution. Therefore there is no duality gap, the KKT conditions are sufficient for optimality, and X_s^* and y_s^* are primal and dual optimal if and only if they satisfy

$$\begin{aligned} &\mathcal{A}_c^T(b_c - \mathcal{A}_c(X_s^*)) + \mathcal{A}_s^T(y_s^*) \preceq 0 \\ &\mathcal{A}_s(X_s^*) = b_s \\ &(\mathcal{A}_c^T(b_c - \mathcal{A}_c(X_s^*)) + \mathcal{A}_s^T(y_s^*)) \bullet X_s^* = 0 \\ &X_s^* \succeq 0. \end{aligned}$$

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

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

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

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 A characterization of minimal unsatisfiability

We now turn our attention to MUSes. Our interest is in exploring how the SDP approach can provide information about MUSes. The results in this section, while they do not involve SDP, are interesting by themselves, and furthermore Lemma 4.1 will be useful for the proofs in Section 5.2.

Definition 4.1 We say that F is a minimal unsatisfiable formula 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 sub-formula within it. This motivates the next definition.

Definition 4.2 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 minimal unsatisfiable formulas in terms of the sets $T(C)$.

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

$$T(C_i) \not\subseteq \bigcup_{\substack{k=1,\ldots,m \\ k \neq i}} T(C_k), \quad i = 1, \ldots, m \tag{7}$$

$$\bigcup_{k=1}^{m} T(C_k) = \mathcal{P}(V). \tag{8}$$

Proof. By (7), there exists $S \in T(C_i)$ such that x^S does not satisfy C_i but satisfies all other clauses. Together with condition (8) we have that F is unsatisfiable and $F \setminus \{C_i\}$ is satisfiable. The sufficient condition is proved. If F is a MUS, then for each $C \in F$ we have that F is unsatisfiable and $F \setminus \{C\}$ is satisfiable. Thus, there is x^S not satisfying C and satisfying all other clauses of F. Thus, condition (7) is proved. Condition (8) follows from the unsatisfiability of F. □

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*

$$T(C_i) \subseteq \bigcup_{\substack{k=1,\ldots,m \\ k \neq i}} T(C_k). \tag{9}$$

5 Semidefinite least squares residuals and minimal unsatisfiable subformulas

This section presents the main results of this paper. We first state and discuss the results in Section 5.1. The proofs follow in Section 5.2.

5.1 Main results

Let us define a weight for each clause/constraint using the residuals from the SLS problem.

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

$$w_i = w(C_i) = 2^{|I_i|} r_i.$$

The first result indicates how the weights can be used a measure of hardness of satisfying a 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*

$$w_k < w_i.$$

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*

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

Note that the converse of Theorem 5.2 does not hold in general. This is shown by Example 2 in Section 6.

The third result is that if a clause does not belong to any MUS, then its corresponding residual, and hence its weight, is zero.

Theorem 5.3 *Let F be an unsatisfiable CNF formula and C a clause in F. If $C \notin G$, for all minimal unsatisfiable formula $G \subseteq F$, then $r(C) = 0$.*

Example 1 in Section 6 illustrates Theorems 5.1 and 5.3.

5.2 Proofs

Every $X \in \Omega_n$ can be expressed as

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

We also have

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

because

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

Let X^* be a minimizer of (LSQa). Then the components of the residual vector r have the form

$$r_j = q_j(X^*) = 2^{|I_j|} \sum_{S \in T_j} \alpha_S. \tag{10}$$

Since

$$\|b - A(X)\|^2 = \sum_{j=1}^{m} q_j^2(X)$$

and

$$q_j^2(X) = \left(2^{|I_j|} \sum_{S \in T_j} \alpha_S\right)^2,$$

it follows that

$$\|b - A(X)\|^2 = \sum_{j=1}^{m} \left(2^{|I_j|} \sum_{S \in T_j} \alpha_S\right)^2 = \sum_{j=1}^{m} 4^{|I_j|} \left(\sum_{S \in T_j} \alpha_S\right)^2.$$

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

$$\begin{aligned} \min & \sum_{j=1}^{m} \left(\sum_{S \in T_j} 2^{|I_j|} \alpha_S\right)^2 \\ \text{s.t.} & \sum_{S \subseteq V} \alpha_S = 1 \\ & \alpha_S \geq 0, \ \forall S \subseteq V. \end{aligned} \tag{11}$$

The KKT conditions of (11) are sufficient for optimality:

$$2 \sum_{j \in M_S} \sum_{J \in T_j} 4^{|I_j|} \alpha_J - \lambda_S - z = 0 \quad \forall S \subseteq V, \tag{12}$$

$$\lambda_S \alpha_S = 0 \quad \forall S \subseteq V, \tag{13}$$

$$\sum_{S \subseteq V} \alpha_S = 1, \tag{14}$$

$$\alpha_S, \lambda_S \geq 0 \ \forall S \subseteq V, \tag{15}$$

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

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 (11). 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

$$2 \sum_{J \in T_k} 4^{|I_k|} \alpha_J + 2 \sum_{j \in M_S \setminus \{i,k\}} \sum_{J \in T_j} 4^{|I_j|} \alpha_J - \lambda_S = -\lambda_R. \tag{16}$$

If $\lambda_S \neq 0$, then $\alpha_S = 0$. If $\lambda_S = 0$, then equation (16) is satisfied only if $\lambda_R = 0$ and $\alpha_S = 0$. The result is proved. $\square$

The following lemma is a general version of the previous one.

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

Proof. We have to use the KKT condition (12) 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

$$2 \sum_{j \in M_S \setminus M_R} \sum_{J \in T_j} 4^{|I_j|} \alpha_J \lambda_S = \lambda_S - \lambda_R. \tag{17}$$

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 (17) is non-negative and the left side is non-positive, $\alpha_S = 0$.

The result is proved. □

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 (12) 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,

$$-\lambda_S = 2 \sum_{j \in M_{S'} \setminus \{k\}} \sum_{J \in T_j} 4^{|I_j|} \alpha_J - \lambda_{S'}. \tag{18}$$

We consider two cases:

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

(ii) if $\lambda_{S'} = 0$ then, from (18), $\lambda_S = 0$ and $\alpha_{S'} = 0$.

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

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

Lemma 5.4 *Let F be an unsatisfiable CNF formula. 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. $\square$

Theorem 5.4 *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. Then*

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

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. $\square$

Theorem 5.1 follows directly from Theorem 5.4 by the observation that the components of the residual have the form $r_k = \sum_{S \in T_k} 2^{|I_k|} \alpha_S$ and $r_i = \sum_{S \in T_i} 2^{|I_i|} \alpha_S$.

It remains only to give the proof of Theorem 5.2.

Proof of Theorem 5.2. 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, (12), 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

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

The result is proved. □

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 $r(C) = 0$.*

Proof. The proposition follows by Lemma 5.2. □

Finally, Theorem 5.3 follows by Propositions 5.2 and 5.1.

6 Examples

In this section, we present some examples to illustrate our results.

Example 6.1 shows that the converse of Theorem 5.2 does not hold in general. Example 6.2 illustrates Theorems 5.1 and 5.3. Example 6.3 is the application of our theoretical approach to an instance from [19]. The example shows that our approach provides the same information about the instance as the algorithm HYCAM proposed in [19] for finding MUSes.

Example 6.1 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.

Below, we list the possible truth assignments:

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}

For simplicity of the notation we indexed the sets $J \subseteq V$ to numbers $1, 2, \ldots, 2^3$. We have

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

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

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

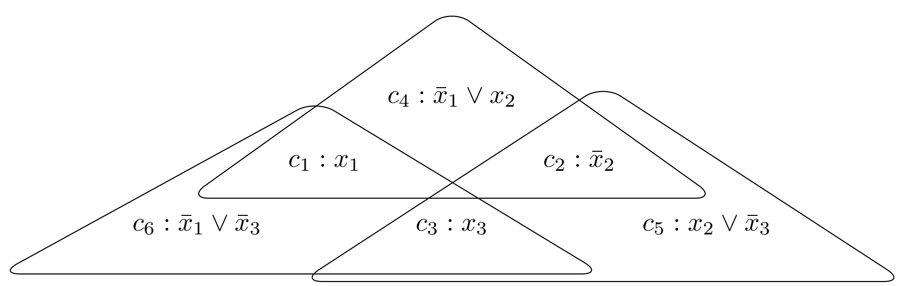

Figure 1: MUSes of F

Example 6.2 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

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 $1, 2, \dots, 2^3$. 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

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

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

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

In Figure 2 we identify MUSes 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 $r_7 = 0$, 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.

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

Figure 2: MUSes of F

Example 6.3 This example is taken from [19].

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

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, using the KKT conditions, and without computing the values of the remaining α_i's, we deduce that

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

With our theoretical approach, we identified the same pattern for MUSes that is identified by algorithm HYCAM [19], as it can be seen in the next figure.

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 these weights are a measure of the importance of each clause in rendering the formula unsatisfiable, and we identified the values of the weights in two important cases, namely when a formula is a MUS, and when a clause does not belong to any MUS. These results in this paper establish the basis for an semidefinite optimization-based algorithm to identify minimal unsatisfiable subformulas within a given formula. The design of such an algorithm will likely also use techniques for handling infeasiblity in optimization, see e.g. [12], and the semidefinite approach has the advantage that it

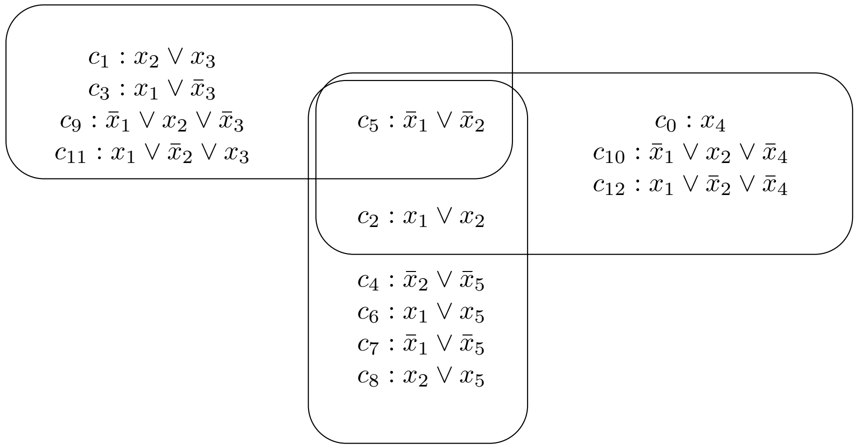

Figure 3: MUSes of F

can in principle provide the means to certify that a MUS is in fact the smallest MUS, i.e., the MUS with a globally minimum number of clauses.

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.

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, 2012.

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

[8] 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.

[9] 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.

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

[11] 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.

[12] J.W Chinneck. Feasibility and Infeasibility in Optimization:: Algorithms and Computational Methods, volume 118. Springer, 2007.

[13] S.A. Cook. The complexity of theorem-proving procedures. In Proc. of 3rd Annual ACM Symp. on Theory of Computing, pages 151–158, 1971.

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

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

[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.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.

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

[20] C. Helmberg. Semidefinite programming for combinatorial optimization. Technical Report ZR-00-34, Konrad-Zuse-Zentrum für Informationstechnik Berlin.

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

[22] 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.

[23] 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.

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

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