Sums of Squares, Satisfiability and Maximum Satisfiability

Hans van Maaren
Linda van Norden

Technical University Delft
Faculty of Electrical Engineering, Mathematics and Computer Science
Department of Software Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
Email: H.vanMaaren@ewi.tudelft.nl, L.vanNorden@ewi.tudelft.nl

September 30, 2004

Abstract

Recently the Mathematical Programming community showed a renewed interest in Hilbert's Positivstellensatz. The reason for this is that global optimization of polynomials in $\mathbb{R}[x_1, \ldots, x_n]$ is $\mathcal{NP}$-hard, while the question whether a polynomial can be written as a sum of squares has tractable aspects. This is due to the fact that Semidefinite Programming can be used to decide in polynomial time (up to a prescribed precision) whether a polynomial can be rewritten as a sum of squares of linear combinations of monomials coming from a specified set. We investigate this approach in the context of Satisfiability. We are interested in the question whether the Satisfiability area can provide examples of polynomials which are non-negative but not sums of squares but also examine the potential of the theory for providing tests for unsatisfiability and providing MAXSAT upper bounds. We compare the SOS (Sums Of Squares) approach with existing upper bound techniques (for the MAX-2-SAT case) of Goemans and Williamson [7] and Feige and Goemans [6], which are based on Semidefinite Programming as well. We advocate that the SOS approach gives more accurate upper bounds.

1 Introduction

Hilbert's Positivstellensatz states that a non-negative polynomial in $\mathbb{R}[x_1, \ldots, x_n]$ is a SOS in case $n = 1$, or has degree $d = 2$, or $n = 2$ and $d = 4$. Despite these restrictive constraints explicit counter examples for the non-covered cases are rare although it has been proven that there must be many of them [2]. An interesting question therefore is whether random k-SAT threshold instances could provide such examples. The polynomials we will associate with these instances are very similar in appearance while those coming from unsatisfiable instances could provide counter examples, as will be clarified later on. On the other side [9] claims that for purposes of optimization, the replacement of the fact that a polynomial is non-negative by the assumption that it is a SOS gives very good results in practice. This claim could imply that we can develop an upper bound algorithm for MAXSAT using this SOS approach which gives tighter bounds than the existing ones.

Let us go into more detail now. Suppose a given polynomial $p(x_1, \ldots, x_n) \in \mathbb{R}[x_1, \ldots, x_n]$ has to be minimized over $\mathbb{R}^n$. This minimization can be written as the program

$$\begin{aligned} \max \alpha \\ \text{s.t. } p(x_1, \ldots, x_n) - \alpha \geq 0 \text{ on } \mathbb{R}^n \\ \alpha \in \mathbb{R} \end{aligned} \tag{1}$$

Clearly, the program

$$\begin{aligned} \max \alpha \\ \text{s.t. } p(x_1, \ldots, x_n) - \alpha \text{ is a SOS} \\ \alpha \in \mathbb{R} \end{aligned} \tag{2}$$

would result in a lower bound for problem (1).

There is a benefit in the approach above using the theory of 'Newton Polytopes'. The Newton polytope associated with a polynomial is constructed as follows: the exponent of a monomial $x_1^{a_1} \ldots x_n^{a_n}$ is identified with a lattice point $\bar{a} = (a_1, \ldots, a_n)$. The Newton polytope is the convex hull of all those lattice points associated with monomials appearing in the polynomial involved. Monomials useful for finding a SOS decomposition are those with an exponent $\bar{a}$ for which $2\bar{a}$ is in the Newton polytope. Thus adding more monomials would not enlarge the chance of success. Resuming, this means that for purposes of global optimization of a polynomial over $\mathbb{R}^n$ we have the advantage to know which monomials are possibly involved in the SOS decomposition (if existing) while we face the disadvantage that non-negative polynomials need not be decomposable as SOS.

Involving side constraints of the form we are interested in (the Boolean constraints of the form $x_1^2 - 1 = 0, \ldots, x_n^2 - 1 = 0$) the situation turns. It can be proven that polynomials which are non-negative on $\{-1, 1\}^n$ (please note that we use $\{-1, 1\}$- values for Boolean variables instead of the more commonly used $\{0, 1\}$-values, which is much more attractive when algebraic formalisms are involved) can always be written as a SOS modulo the ideal generated by the polynomials $x_1^2 - 1, \ldots, x_n^2 - 1$ (further denoted by $I_{\mathcal{B}}$) but in this case the 'Newton Polytope Property' is not valid, because higher degree monomials may cancel ones with lower degree, when performing calculations modulo $I_{\mathcal{B}}$. Hence, we have to consider possibly an exponential set of monomials in the SOS decomposition. To see this consider a polynomial $p(x_1, \ldots, x_n)$ which is non-negative on $\{-1, 1\}^n$. The expression

$$\sum_{\sigma \in \{-1,1\}^n} \frac{p(\sigma)}{2^n} (1 + \sigma_1 x_1) \ldots (1 + \sigma_n x_n) \tag{3}$$

is easily seen to give the same outputs on $\{-1, 1\}^n$ as $p(x_1, \ldots, x_n)$. Since each

$\frac{(1+\sigma_j x_j)}{2}$ is a square modulo $I_{\mathcal{B}}$ because

$$\left(\frac{1+\sigma_j x_j}{2}\right)^2 \equiv \frac{1+\sigma_j x_j}{2} \text{ modulo } I_{\mathcal{B}} \tag{4}$$

this expression is seen to be a SOS modulo $I_{\mathcal{B}}$. But at the same time it becomes evident that we might need an exponentially large basis of monomials in realizing this decomposition. Resuming, we see that if we want to optimize a polynomial over $\{-1,1\}^n$ we have the advantage that we know that a basis of monomials exists which will give us an exact answer, while we are facing the disadvantage that this basis could be unacceptably large.

We come to the point of explaining the SOS formalism. Let $M^T = (M_1, \ldots, M_k)$ be a row vector of monomials in variables $x_1, \ldots, x_n$ and $p(x_1, \ldots, x_n)$ a given polynomial in $\mathbb{R}[x_1, \ldots, x_n]$. The equation

$$M^T L^T L M = p \tag{5}$$

involving any matrix L of appropriate size would give an explicit decomposition of p as a SOS over the monomials used. Conversely, any SOS decomposition of p can be written in this way. This means that the Semidefinite Program

$$\begin{array}{c} M^T S M = p \\ S \text{ positive semidefinite } (S \succeq 0) \end{array} \tag{6}$$

gives a polynomial time decision method for the question whether p can be written as a SOS using $M_1, \ldots, M_k$ as a basis of monomials (up to prescribed precision: the method uses real numbers represented with a certain precision). The constraint $M^T S M = p$ in fact results in a set of linear constraints in the entries of the matrix S.

If we consider the Boolean side constraints we have a similar program. In this case however the equation $M^T S M = p$ needs to be satisfied only modulo $I_{\mathcal{B}}$. Also this constraint results in a set of linear constraints in the entries of the matrix S, but different from the ones above. This is caused by the above mentioned cancellation effects. We will give some examples later on.

First we shall associate polynomials to CNF formulae. With a literal X_i we associate the polynomial $\frac{1}{2}(1-x_i)$ and with $\neg X_j$ we associate $\frac{1}{2}(1+x_j)$. With a clause we associate the products of the polynomials associated with its literals. Note that for a given assignment $\sigma \in \{-1,1\}^n$ the polynomial associated with each clause outputs a zero or a one, depending of the fact whether σ satisfies the clause or not.

With a CNF-formula ϕ we associate two polynomials F_ϕ and $F_\phi^{\mathcal{B}}$. F_ϕ is the sum of squares of the polynomials associated with the clauses from ϕ . $F_\phi^{\mathcal{B}}$ is just the sum of those polynomials. Clearly, F_ϕ is non-negative on $\mathbb{R}^n$ and $F_\phi^{\mathcal{B}}$ is non-negative on $\{-1,1\}^n$.

Example 1 Let ϕ be the CNF-formula with the following three clauses

$$X \vee Y, \quad X \vee \neg Y, \quad \neg X \tag{7}$$

$$\begin{aligned} F_\phi &= \left(\frac{1}{2}(1-x)\frac{1}{2}(1-y)\right)^2 + \left(\frac{1}{2}(1-x)\frac{1}{2}(1+y)\right)^2 + \left(\frac{1}{2}(1+x)\right)^2 \\ &= \frac{3}{8} + \frac{1}{4}x + \frac{3}{8}x^2 + \frac{1}{8}y^2 + \frac{1}{4}xy^2 + \frac{1}{8}x^2y^2 \end{aligned} \tag{8}$$

In order to attempt to rewrite $F_\phi - \alpha$ as a SOS it suffices to work with the monomial basis $M^T = (1, x, y, xy)$. The program

$$\begin{aligned} &\max \alpha \\ \text{s.t. } &F_\phi - \alpha = M^T S M \\ &\alpha \in \mathbb{R}, S \succeq 0 \end{aligned} \tag{9}$$

gives an output $\alpha = \frac{1}{3}$, from which we may conclude that $2\frac{2}{3}$ is an upper bound for the MAXSAT-solution of ϕ. Notice that

$$F_\phi = \frac{1}{3} + \frac{3}{8}\left(x + \frac{1}{3}\right)^2 + \frac{1}{8}(xy - y)^2 \tag{10}$$

$$F_\phi^{\mathcal{B}} = \frac{1}{2}(1-x)\frac{1}{2}(1-y) + \frac{1}{2}(1-x)\frac{1}{2}(1+y) + \frac{1}{2}(1+x) = 1 \tag{11}$$

Clearly, $F_\phi^{\mathcal{B}} = 1$ means that any assignment will exactly violate one clause.

Example 2 Let ϕ be the following CNF-formula

$$X \vee Y \vee Z, \quad X \vee Y \vee \neg Z, \quad \neg Y \vee \neg T, \quad \neg X, \quad T \tag{12}$$

$$F_\phi^{\mathcal{B}} = \frac{3}{2} + \frac{1}{4}x - \frac{1}{4}t + \frac{1}{4}xy + \frac{1}{4}yt \tag{13}$$

1. The Semidefinite Program (SDP)

$$\begin{aligned} \max \quad & \alpha \\ \text{s.t.} \quad & F_\phi^{\mathcal{B}} - \alpha \equiv (1, x, y, t)^T S (1, x, y, t) \text{ modulo } I_{\mathcal{B}} \\ & \alpha \in \mathbb{R}, \quad S \succeq 0 \end{aligned} \tag{14}$$

   gives output $\alpha = 0.793$, from which we may conclude that 4.207 is an upper bound for the MAXSAT-solution of ϕ.

2. The SDP

$$\begin{aligned} \max \quad & \alpha \\ \text{s.t.} \quad & F_\phi^{\mathcal{B}} - \alpha \equiv (1, x, y, t, xy, xt, yt)^T S (1, x, y, t, xy, xt, yt) \text{ modulo } I_{\mathcal{B}} \\ & \alpha \in \mathbb{R}, S \succeq 0 \end{aligned} \tag{15}$$

   gives an output $\alpha = 1$. Note that the second approach gives a tighter upper bound, because more monomials are contained in the basis.

Next, we formulate some useful properties on the polynomials F_ϕ and $F_\phi^{\mathcal{B}}$. We omit proofs here. Let m be the number of clauses in ϕ.

Theorem 1. *1. For any assignment $\sigma \in \{-1, 1\}^n, F_\phi(\sigma) = F_\phi^{\mathcal{B}}(\sigma)$. Both give the number of clauses violated by σ.*

2. Both $\min_{\sigma \in \{-1,1\}^n} F_\phi(\sigma)$ and $\min_{\sigma \in \{-1,1\}^n} F_\phi^{\mathcal{B}}(\sigma)$ give rise to an exact MAXSAT-solution of ϕ.

3. $F_\phi^{\mathcal{B}} \equiv F_\phi$ modulo $I_{\mathcal{B}}$.

4. *F_ϕ attains it minimum over $\mathbb{R}^n$ somewhere at the hypercube $[-1,1]^n$ (a compact set), while it can be zero only in a $\sigma \in \{-1,1\}^n$.*

5. *ϕ is unsatisfiable if and only if there exists an $\epsilon > 0$ such that $F_\phi - \epsilon \geq 0$ on $\mathbb{R}^n$.*

6. *If there exists an $\epsilon > 0$ such that $F_\phi - \epsilon$ is a SOS, then ϕ is unsatisfiable.*

7. *If there exists a monomial basis M and an $\epsilon > 0$ such that $F_\phi^{\mathcal{B}} - \epsilon$ is a SOS based on M, modulo $I_{\mathcal{B}}$, then ϕ is unsatisfiable.*

8. *Let M be a monomial basis, then*

$$\begin{aligned} m - \max \alpha \\ s.t.\ \ F_\phi - \alpha \quad \textit{is a SOS} \\ \alpha \in \mathbb{R} \end{aligned} \tag{16}$$

and

$$\begin{aligned} m - \max \alpha \\ s.t.\ \ F_\phi^{\mathcal{B}} - \alpha \equiv M^T S M \ \ \textit{modulo}\ \ I_{\mathcal{B}} \\ \alpha \in \mathbb{R}, S \succeq 0 \end{aligned} \tag{17}$$

give upper bounds for the MAXSAT-solution of ϕ.

Theorems 1.5 and 1.8 are the basis for the search for counterexamples for the non-covered cases of Hilbert's Positivstellensatz. Clearly, a 2-SAT formula ϕ gives a polynomial F_ϕ with degree 4 and the SDP

$$\begin{aligned} \max \alpha \\ \text{s.t.}\ \ F_\phi - \alpha \ \text{ is a SOS} \\ \alpha \in \mathbb{R} \end{aligned} \tag{18}$$

must have $\alpha = 0$ for a satisfiable formula (F_ϕ is a SOS itself). For an unsatisfiable formula ϕ, α might be zero, in which case $F_\phi - \epsilon$, with ϵ sufficiently small, is a non-negative polynomial, but not a SOS. The optimal α might be positive, in which case ϕ does not provide us with a counterexample. We will report on some experiments in Section 2.

Theorem 1.8 is the basis for the search for MAXSAT-upper bounds. Conclusions will be described in Section 3.

2 Counterexamples with d=4?

We generated a set of 2-SAT instances with 10 and 25 variables with a density varying between 1.0 and 2.0. For all 2200 sample instances, the SDP (18) gives solutions such that any α corresponding to a satisfiable instance of a particular size and density is smaller than any α corresponding to an unsatisfiable instance of the same size and density. From this we cannot conclude that all F_ϕ's corresponding to unsatisfiable ϕ's have the property that $F_\phi - \epsilon$ is a SOS for sufficiently small ϵ. The reason is that the solver used (Sedumi [10]) runs into numerical problems when accurate precision is requested: some values α

associated with satisfiable ϕ's are actually positive (in the fifth decimal or later), which is clearly not allowed. Moreover, some α's associated with unsatisfiable CNF-formulae ϕ are only larger in the sixth decimal compared to some α's associated with satisfiable CNF-formulae. Hence, although the sample set is in fact perfectly separated in satisfiable and unsatisfiable instances by the test, numerical imprecisions withhold us from a definite conclusion.

Based on the experiments there is no reason to suppose that polynomials of degree 4 coming from unsatisfiable 2-SAT formulae may provide us with the desired counterexamples. In the other hand, the numerical imprecision keeps us from the opposite conclusion as well. Only much more accurate optimization methods, or more accurate implementations of SDP-algorithms, could result in a more final conclusion.

3 Upper bounds for MAX-2-SAT

Although the SOS approach provides upper bounds for MAXSAT-solutions not only restricted to MAX-2-SAT, we restrict the experiments to the latter class for the moment. The reason is that we want to present a comparison with the results of the famous method of Goemans and Williamson [7] and Feige and Goemans [6] first.

The first obvious monomial basis to consider is $M^T = (1, x_1, \ldots, x_n)$, where the x_i are the variables in ϕ. We refer to this choice as M_{GW} for reasons which will be clear later.

Semidefinite Programming Formulations come in pairs: the so-called primal and dual formulations. At this stage, we only refer to [5] and [4] for details and just mention that for our applications, these pairs return the same result, although the actual formulations differ considerably.

3.1 Comparison with Goemans-Williamson approach

The original Goemans-Williamson approach for obtaining an upper bound for a MAX-2-SAT problem starts with $F_\phi^{\mathcal{B}}$ too. The problem

$$\begin{array}{c} \min F_\phi^{\mathcal{B}}(x) \\ x \in \{-1, 1\}^n \end{array} \tag{19}$$

is relaxed by relaxing the Boolean arguments x_i. With each x_i, a vector $v_i \in \mathbb{R}^{n+1}$ is associated, with norm 1, and products $x_i x_j$ are interpreted as inproducts $v_i \bullet v_j$. In this way, they turn (19) into a Semidefinite Program, after making $F_\phi^{\mathcal{B}}$ homogenous by adding a dummy vector v_{n+1} in order to make the linear terms quadratic as well. For example, $3x_i$ is replaced by $3(v_i \bullet v_{n+1})$. Hence, Goemans and Williamson solve

$$\begin{array}{c} \min F_\phi^{\mathcal{B}}(v_1, \ldots, v_n, v_{n+1}) \\ \|v_i\| = 1, v_i \in \mathbb{R}^{n+1} \end{array} \tag{20}$$

which is the same as

$$\begin{array}{c} \min \sum_{i=1}^{n+1} \sum_{j=1}^{n+1} b_{ij} s_{ij} \\ s_{ii} = 1, S \succeq 0 \end{array} \tag{21}$$

In the above, s_{ij} replaced $v_i \bullet v_j$ and the b_{ij}'s correspond to the coefficients of the polynomial $F_\phi^{\mathcal{B}}$. The solution S is decomposed as $L^T L$ providing the interpretation of the vectors v_i.

Hence, while Goemans and Williamson relax the input arguments of $F_\phi^{\mathcal{B}}$, the SOS-method is a relaxation by replacing non-negativity by being a SOS. Surprisingly, the two resulting Semidefinite Programs can be shown dual to each other. Resuming, the upper bounds provided by the Goemans-Williamson-method coincide with the upper bounds obtained by the SOS approach using M_{GW} as a basis for the monomials.

Still, there is something more to say about these two different approaches. (21) has $\frac{1}{2}(n+1)(n+2)$ variables s_{ij} (not $(n+1)^2$ because S is symmetric). In the SOS approach,

$$\begin{aligned} &\max \alpha \\ \text{s.t. } F_\phi^{\mathcal{B}} \equiv M_{GW}^T S M_{GW} \ &\text{modulo } I_{\mathcal{B}} \\ &\alpha \in \mathbb{R}, S \succeq 0 \end{aligned} \tag{22}$$

it is not hard to see that in fact only the diagonal elements are essentially variable, because the off-diagonal elements are fixed. This means that the actual dimension of the SOS-program with monomial basis M_{GW} is linear in the number of variables, while in the Goemans-Williamson formulation (21) the dimension grows quadratically.

The solver Sedumi used in the experiments is not able to benefit from all sparseness structures in the data. Other existing solvers like DSDP [1], CDSDP [3] and SDPA [8] are under consideration with regard to this respect.

4 Adding valid inequalities vs adding monomials

Feige and Goemans [6] propose to add so-called valid inequalities to (21) in order to improve the upper bounds. Two types of these 'triangle inequalities' are considered. The first is the inequality

$$1 + x_i + x_j + x_i x_j \geq 0 \tag{23}$$

In fact, they consider the homogeneous form $1 + x_i x_{n+1} + x_j x_{n+1} + x_i x_j \geq 0$. One might add all these inequalities, also the ones by replacing x_i and/or x_j by $-x_i$ or $-x_j$. Another possibility is to add only those inequalities where x_i and x_j appear together in the same clause. In this section, we examine how this compares with the SOS approach. First, it can be shown that $1+x_i+x_j+x_i x_j$ cannot be recognized as a SOS based on $M^T = (1, x_i, x_j)$. However, if we add $x_i x_j$ to the monomial basis we have

$$1 + x_i + x_j + x_i x_j \equiv \frac{1}{4}\left(1 + x_i + x_j + x_i x_j\right)^2 \ \text{ modulo } I_{\mathcal{B}} \tag{24}$$

Hence, the effect of adding the valid inequality (23) to (21) is captured in the SOS approach by adding the monomial $x_i x_j$ to the basis.

By a careful examination using primal and dual formulations, one can show that adding the monomial $x_i x_j$ results in tighter upper bounds and even captures the effect of adding the other three inequalities

$$1 - x_i - x_j + x_i x_j \geq 0 \tag{25}$$
$$1 - x_i + x_j - x_i x_j \geq 0 \tag{26}$$
$$1 + x_i - x_j - x_i x_j \geq 0 \tag{27}$$

from the Feige-Goemans context. The experiments in section 5 support this fact.

Finally, Feige and Goemans showed that adding the valid inequalities for a triple of variables

$$1 + x_i x_j + x_i x_k + x_j x_k \geq 0 \tag{28}$$
$$1 + x_i x_j - x_i x_k + x_j x_k \geq 0 \tag{29}$$
$$1 - x_i x_j + x_i x_k - x_j x_k \geq 0 \tag{30}$$
$$1 + x_i x_j - x_i x_k - x_j x_k \geq 0 \tag{31}$$

may improve the tightness of the relaxation.

Note that

$$1 + x_i x_j + x_i x_k + x_j x_k = \frac{1}{4}\left(1 + x_i x_j + x_i x_k + x_j x_k\right)^2 \text{ modulo } I_{\mathcal{B}} \tag{32}$$

Hence, the effect of adding (28) is captured by adding $x_i x_j$, $x_i x_k$ and $x_j x_k$ to the monomial basis. Also in this case, the effect of adding the monomials results in tighter upper bounds compared to adding valid inequalities.

Finally, note that adding all inequalities of the form (28) to (31) amounts to adding $\mathcal{O}(n^3)$ inequalities, while in the SOS approach only $\mathcal{O}(n^2)$ pairs need to be added. For the moment it is too early to decide whether existing SDP-solvers are suitable, or can be modified, to turn this effect into a computational benefit as well.

5 Preliminary experimental results

In this section we consider besides the Goemans-Williamson upper bound the next three variants of the Feige-Goemans method.

Variant FG+m: The valid inequalities added in this variant are only those coming directly from the clauses. For instance, if $X \vee \neg Y$ is a clause, we add the valid inequality $1 - x + y - xy \geq 0$.

Variant FG+4p: For each pair of variables X_i and X_j occurring in a same clause, the four inequalities (23) and (25) to (27) are added.

Variant FG+all: For *each* pair of variables the four inequalities (23) and (25) to (27) are added.

We compare the upper bounds resulting from these variants with the upper bound obtained from the semidefinite program (17) with monomial basis M_p consisting of the set $\{1, x_1, \ldots, x_n\}$ extended with all $x_i x_j$ for variables X_i and X_j appearing in a same clause. We call the corresponding upper bound SOS_p.

In our initial experiments we used a set of 900 randomly generated instances with 10 variables. For each of the densities 1.0, 1.5, 2.0, ..., 5.0, 100 instances are considered. In the table below we give for each method the average upper bound over this set of 100 instances for each density. In the table below, the first column indicates the density, the second the upper bounds by SOS_p, the third column gives the Goemans-Williamson upper bound, the fourth gives the upper bound of variant FG+m, the next the upper bound by variant FG+4p and the last column gives upper bound of variant FG+all.

d	SOS_p	GW	FG+m	FG+4p	FG+all
1.0	9.9600	9.9857	9.9658	9.9624	9.9624
1.5	14.8400	14.9494	14.8632	14.8538	14.8538
2.0	19.4516	19.7485	19.5058	19.4781	19.4762
2.5	23.9700	24.3513	24.0288	24.0000	23.9985
3.0	28.2500	28.7421	28.3429	28.3014	28.2975
3.5	32.5500	33.0229	32.6318	32.5845	32.5823
4.0	36.7802	37.2518	36.8689	36.8202	36.8182
4.5	40.8411	41.4043	40.9562	40.8923	40.8889
5.0	45.0209	45.6111	45.1188	45.0745	45.0711

From Table 1 we see that SOS_p gives the tightest upper bounds. This is not only true on average but in fact for each individual instance.

Table 2 gives the same type of results for instances with 25 variables.

d	SOS_p	GW	FG+m	FG+4p	FG+all
1	25.000000	25.0000	25.0000	25.0000	25.0000
1.5	37.657019	37.9924	37.7385	37.7025	37.7011
2	48.805404	49.8262	48.9994	48.9044	48.8960
2.5	60.604647	62.0014	60.9626	60.7375	60.7186
3	71.206614	72.6691	71.4971	71.3461	71.3244
3.5	82.344978	83.8737	82.7146	82.5560	82.5224
4	92.649132	94.2478	93.0320	92.8607	92.8240
5	113.838240	115.5524	114.2348	114.0486	114.0039

6 An algebraic classification of satisfiability

We close our paper with a classification based on Theorem 1. Let ϕ be a CNF-formula. As we have seen in Section 1, we have

Theorem 2. *ϕ is unsatisfiable if and only if $F_\phi^{\mathcal{B}} - \epsilon$ is a SOS modulo $I_{\mathcal{B}}$ for some $\epsilon > 0$.*

Let I_ϕ be the ideal generated by $F_\phi^{\mathcal{B}}$ and $I_{\mathcal{B}}$. We can formulate the rather elegant theorem whose computational implications are not transparent yet.

Theorem 3. *ϕ is unsatisfiable if and only if -1 is a SOS in the ring $\mathbb{R}[x_1, \ldots, x_n]$ modulo I_ϕ.*

References

[1] S.J. Benson and Y.Ye. DSDP4 - a software package implementing the dual-scaling algorithm for semidefinite programming. Tech. Report ANL/MCS-TM-255, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL,, 2002.

[2] G. Blekherman. There are significantly more nonnegative polynomials than sums of squares. http://sitemaker.umich.edu/gblekher/research, 2004.

[3] B. Borchers. CSDP : A C library for semidefinite programming. Technical report, Socorro, NM, USA, 1997.

[4] E. de Klerk. *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications*, volume 65 of *Applied Optimization Series*. Kluwer Academic Publishers, 2002.

[5] E. de Klerk and J.P. Warners. Semidefinite programming relaxations for max 2-sat and 3-sat: Computational perspectives. In Combinatorial and Global Optimization, P.M. Pardalos, A. Migdalas, and R.E. Burkard (eds.), Series on Applied Optimization, Volume 14, World Scientific Publishers, 2002.

[6] U. Feige and M. X. Goemans. Approximating the value of two prover proof systems, with applications to MAX2SAT and MAXDICUT. In *Proceedings of theThirdIsrael Symposium on Theory of Computing and Systems*, pages 182–189, 1995.

[7] M. X. Goemans and D. P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. Assoc. Comput. Mach.*, 42:1115–1145, 1995.

[8] K. Nakata K. Fujisawa, M. Kojima and M. Yamashita. SDPA(Semidefinite Programming Algorithm) : User's manual. Research Reports on Information Sciences, Ser. B : Operations Research B308, Dept. of Information Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152, Japan, 2002.

[9] B. Sturmfels P. A. Parrilo. Minimizing polynomial functions. In Algorithmic and quantitative real algebraic geometry , DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 60, pp. 83–99, AMS.

[10] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999. Special issue on Interior Point Methods.