Sums of Squares, Satisfiability and Maximum Satisfiability

Hans van Maaren and Linda van Norden

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

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 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 and rounding techniques for the MAX-2-SAT case of Goemans and Williamson [8] and Feige and Goemans [6] and the MAX-3-SAT case of Karloff and Zwick [9], which are based on Semidefinite Programming as well. We show that the combination of the existing rounding techniques with the SOS based upper bound techniques yields polynomial time algorithms with a performance guarantee at least as good as the existing ones, but observably better in particular cases.

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 Blekherman [2] proved that there must be many of them. On the other side [10] claims that for purposes of optimization, the replacement of the fact that a polynomial is non-negative by the fact 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.

F. Bacchus and T. Walsh (Eds.): SAT 2005, LNCS 3569, pp. 294–308, 2005.
© Springer-Verlag Berlin Heidelberg 2005

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 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 associated with a polynomial 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. 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 a SOS decomposition (if existing) while we face the disadvantage that non-negative polynomials need not be decomposable as SOS.

Involving 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 $I_{\mathcal{B}}$ generated by the polynomials $x_1^2 - 1, \ldots, x_n^2 - 1$. However, 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

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

is easily seen to give the same outputs on $\{-1,1\}^n$ as $p(x_1,\ldots,x_n)$. Each $\frac{1+\sigma_j x_j}{2}$ is a square modulo $I_{\mathcal{B}}$ because

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

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

We now come to the point of explaining the SOS approach. 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 $M^T L^T L M = p$ 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

$$\begin{aligned} &M^T S M = p \\ &S \text{ positive semidefinite } \ (S \succeq 0) \end{aligned} \tag{5}$$

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. 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 $\{-1,1\}^n$. $F_\phi(\sigma)$ and $F_\phi^{\mathcal{B}}(\sigma)$ give the number of clauses violated by assignment σ. The following two examples illustrate the construction of F_ϕ and $F_\phi^{\mathcal{B}}$ and the outcome of the corresponding SDP's.

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

$$X \vee Y, \quad X \vee \neg Y, \quad \neg X \tag{6}$$

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

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

$$\begin{aligned} &\max \alpha \\ &\text{s.t. } F_\phi - \alpha = M^T S M \\ &\alpha \in \mathbb{R}, S \succeq 0 \end{aligned} \tag{7}$$

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 $F_\phi = \frac{1}{3} + \frac{3}{8}\left(x + \frac{1}{3}\right)^2 + \frac{1}{8}\left(xy - y\right)^2$. For this ϕ, $F_\phi^{\mathcal{B}} = \frac{1}{2}(1-x)\frac{1}{2}(1-y) + \frac{1}{2}(1-x)\frac{1}{2}(1+y) + \frac{1}{2}(1+x) = 1$. Clearly, $F_\phi^{\mathcal{B}} = 1$ means that any assignment will exactly violate one clause.

Example 2. Let ϕ be the following CNF formula

$$X \vee Y \vee Z, \quad X \vee Y \vee \neg Z, \quad \neg Y \vee \neg T, \quad \neg X, \quad T \tag{8}$$

$$F_\phi^{\mathcal{B}} = \frac{3}{2} + \frac{1}{4}x - \frac{1}{4}t + \frac{1}{4}xy + \frac{1}{4}yt \tag{9}$$

The Semidefinite Program (SDP)

$$\begin{aligned} &\max \alpha \\ &\text{s.t.} F_\phi^{\mathcal{B}} - \alpha \equiv (1, x, y, t) S (1, x, y, t)^T \text{ modulo } I_{\mathcal{B}} \\ &\alpha \in \mathbb{R}, S \succeq 0 \end{aligned} \tag{10}$$

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

$$\begin{aligned} &\max \alpha \\ &\text{s.t.} F_\phi^{\mathcal{B}} - \alpha \equiv (1, x, y, t, xy, xt, yt) S (1, x, y, t, xy, xt, yt)^T \text{ modulo } I_{\mathcal{B}} \\ &\alpha \in \mathbb{R}, S \succeq 0 \end{aligned} \tag{11}$$

gives output $\alpha = 1$. Note that the second SDP 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 $F_\phi^{\mathcal{B}}$. Let m be the number of clauses and n the number of variables 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. $\min_{\sigma \in \{-1,1\}^n} F_\phi(\sigma)$ and $\min_{\sigma \in \{-1,1\}^n} F_\phi^{\mathcal{B}}(\sigma)$ give rise to an exact MAXSAT-solution of ϕ: respectively $m - \min_{\sigma \in \{-1,1\}^n} F_\phi(\sigma)$ and $m - \min_{\sigma \in \{-1,1\}^n} F_\phi^{\mathcal{B}}(\sigma)$.

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

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

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*

$$\begin{aligned} & m - \max \alpha && (12)\\ & \textit{s.t. } F_\phi - \alpha \quad \textit{is a SOS} \\ & \alpha \in \mathbb{R} \\ & \textit{and} \\ & m - \max \alpha && (13)\\ & \textit{s.t. } F_\phi^{\mathcal{B}} - \alpha \equiv M^T S M \quad \textit{modulo } I_{\mathcal{B}} \\ & \alpha \in \mathbb{R}, S \succeq 0 \end{aligned}$$

give upper bounds for the MAXSAT-solution of ϕ.

Proof. Except for part 1.4 the reasonings behind the other parts are already discussed before or they are direct consequences of earlier statements. Here we prove Theorem 1.4: suppose F_ϕ takes its minimum in x and assume $x_1 = 1+\delta$ for some $\delta > 0$. This gives rise to contributions $(\frac{1}{2}(1+(1+\delta)))^2$ and $(\frac{1}{2}(1-(1+\delta)))^2$. Both contributions are smaller with $\delta = 0$ than with $\delta > 0$. The same argument can be applied for $x_1 = -1-\delta$. Thus $x \in [-1, 1]^n$. Furthermore, $F_\phi = 0$ only if in each polynomial associated to a clause at least one of the factors equals zero because F_ϕ is a sum of squares and hence non-negative. This can only be realized if in each polynomial associated to a clause at least one of the variables takes value 1 or -1 resulting in a partial satisfying assignment for ϕ.

Program (13) is the basis for the search for MAXSAT-upper bounds in this paper.

2 Upper Bounds for MAX-2-SAT

Although the SOS approach provides upper bounds for general MAXSAT-solutions, we restrict ourselves in this section to MAX-2-SAT. The reason is that we want to present a comparison with the results of the famous methods of Goemans and Williamson [8] and Feige and Goemans [6].

The first monomial basis we consider in the SOS approach (13) is $M^{GW} = (1, x_1, \ldots, x_n)$, where the x_i's come from the variables in ϕ.

Semidefinite Programming formulations come in pairs: the so-called primal and dual formulations, see for example [5] and [4]. Here we present a generic formulation of a primal semidefinite problem P, and its dual program D. In our context, D and P give the same optimal value.

Consider the program P

$$\min \operatorname{Tr}(CX) \tag{14}$$

$$\text{s.t. } \operatorname{diag}(X) = e \tag{15}$$

$$\begin{aligned}&\mathrm{Tr}(A_j X) \geq 1, \quad j = 1, \ldots, k \\ &X \succeq 0\end{aligned} \tag{16}$$

In the above formulation, C and X are symmetric square matrices of size, say $p \times p$. Tr means the trace of the matrix, i.e. the sum of the entries on the diagonal.

$$Tr(CX) = \sum_{i=1}^{p} \sum_{j=1}^{p} c_{ij} x_{ij}$$

diag$(X) = e$ means that the entries on the diagonal of matrix X are all ones. The A_j's are square symmetric matrices.

The program P has the following dual program D

$$\begin{aligned}&\max \sum_{i=1}^{p} \gamma_i + \sum_{j=1}^{k} y_j \\ &\text{s.t.Diag}(\gamma) + \sum_{j=1}^{k} y_j A_j + U = C \\ &U \succeq 0, y_j \geq 0\end{aligned} \tag{17}$$

in which the y_j's are the dual variables corresponding to constraints (16) and the γ_i's the dual variables corresponding to constraints (15). U is a symmetric square matrix and Diag(γ) is the square matrix with the γ_i's on the diagonal and all off-diagonal entries equal to zero.

2.1 Comparison of SOS Approach and 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

$$\begin{aligned}&\min F_\phi^{\mathcal{B}}(x) \\ &x \in \{-1, 1\}^n\end{aligned} \tag{18}$$

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 (18) 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 in $F_\phi^{\mathcal{B}}$ quadratic as well. For example, $3x_i$ is replaced by $3(v_i \bullet v_{n+1})$. Let $\hat{F}_\phi^{\mathcal{B}}$ be the polynomial constructed from $F_\phi^{\mathcal{B}}$ in this way. The problem Goemans and Williamson solve is

$$\begin{aligned}&\min \hat{F}_\phi^{\mathcal{B}}(v_1, \ldots, v_n, v_{n+1}) \\ &s.t. \ \|v_i\| = 1, v_i \in \mathbb{R}^{n+1}\end{aligned} \tag{19}$$

To transform (19) to a semidefinite program, $v_i \bullet v_j$ is replaced by t_{ij}. Let b_{ij} be the coefficient of $M_i^{GW} M_j^{GW}$ in the polynomial $F_\phi^{\mathcal{B}}$. Let $f_{ij} = f_{ji} = \frac{1}{2} b_{ij}$ and $f_{ii} = 0$ for each i. Let $M(F)$ be the symmetric matrix with entries f_{ij} and T is a symmetric matrix of the same size. Furthermore, let c_0 be the constant term in $F_\phi^{\mathcal{B}}$. To be precise, c_0 equals $\frac{1}{2}$ times the number of 1-literal clauses plus $\frac{1}{4}$ times the number of 2-literal clauses in ϕ.

Consequently, (19) is equivalent to the following semidefinite program

$$\begin{aligned} & c_0 + \min \sum_{i=1}^{n+1} \sum_{j=1}^{n+1} f_{ij} t_{ij} \\ & \text{s.t. } \; t_{ii} = 1, T \succeq 0 \end{aligned} \tag{20}$$

or in matrix notation,

$$\begin{aligned} & c_0 + \min \operatorname{Tr}(M(F)T) \\ & \text{s.t. } \; \operatorname{diag}(T) = e, T \succeq 0 \end{aligned} \tag{21}$$

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. The next theorem proves that the SOS-approach with monomial basis M^{GW} gives the same upper bound for MAX-2-SAT as program (21).

Theorem 2. *The SOS-approach with monomial basis M^{GW} gives the same upper bound as the upper bound obtained by the algorithm by Goemans and Williamson.*

Proof. In the Goemans-Williamson SDP (21) we only have the constraints of type (15), and not of type (16). This implies that we have to deal only with the γ_j-variables. The size of the variable matrix T is $n+1$, with n the number of variables in the CNF formula.

The dual problem of the GW-semidefinite program (21) is

$$\begin{aligned} & c_0 + \max \sum_{i=1}^{n+1} \gamma_i \\ & \text{s.t. } \; \operatorname{Diag}(\gamma) + U = M(F) \\ & U \succeq 0, \gamma_i \; \text{ free} \end{aligned} \tag{22}$$

We start with the program

$$\begin{aligned} & \max \alpha \\ & \text{s.t. } F_\phi^{\mathcal{B}} - \alpha \equiv M^T S M \; \text{ modulo } \; I_{\mathcal{B}} \\ & S \succeq 0, \alpha \in \mathbb{R} \end{aligned} \tag{23}$$

with monomial basis $M = M^{GW} = (M_1^{GW}, \ldots, M_{n+1}^{GW}) = (1, x_1, \ldots, x_n)$ and prove that it is equal to (22). Let s_{ij} be the (i,j)-th element of matrix S.

We can reformulate program (23) as

$$\begin{aligned} &\max \alpha \\ &\text{s.t.} \sum_{i=1}^{n+1}\sum_{j=1}^{n+1} s_{ij} M_i^{GW} M_j^{GW} \equiv F_\phi^{\mathcal{B}} - \alpha \ \text{ modulo } \ I_{\mathcal{B}} \\ &S \succeq 0, \alpha \in \mathbb{R} \end{aligned} \tag{24}$$

Consider the constraint in program (24) for the coefficient of the constant. On the left hand side we have $\sum_{i=1}^{n+1} s_{ii}$, on the right hand side we have $c_0 - \alpha$. This results in the equality

$$\alpha = c_0 - \sum_{i=1}^{n+1} s_{ii} \tag{25}$$

We transform (24) to matrix notation. In the matrix formulation (26), on both left and right hand sides we have a matrix with on position (i, j), $i \neq j$, the coefficient of $M_i^{GW} M_j^{GW}$ using symmetry. Substituting (25) and using matrix notation we can reformulate (24) as

$$\begin{aligned} &c_0 + \max \sum_{i=1}^{n+1} -s_{ii} \\ &\text{s.t. } \ S - \text{diag}(S) = M(F) \\ &S \succeq 0 \end{aligned} \tag{26}$$

Identifying γ_i with matrix entries $-s_{ii}$ and U with S, it is immediate that (26) is equivalent to (22).

Hence, we proved that (23) with monomial basis M^{GW} equals (22). It can be concluded that the Goemans-Williamson SDP and the SDP of the SOS-approach with monomial basis M^{GW} are dual problems providing the same upper bound to the MAX-2-SAT solution.

Still, there is something more to say about these two different approaches. (20) has $\frac{1}{2}(n+1)(n+2)$ variables t_{ij} (not $(n+1)^2$ because T is symmetric). In the SOS approach (13) with monomial basis M^{GW}, 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 (20) the dimension grows quadratically.

In our experiments we tried several semidefinite programming solvers like Sedumi [11], DSDP [1], CSDP [3] and SDPA [7], but none of them could fully benefit from this factHowever, we found that CSDP performed best on SDP's of the form (13).

2.2 Adding Valid Inequalities vs Adding Monomials

Feige and Goemans [6] propose to add so-called valid inequalities to (21) in order to improve the quality of the relaxation. A valid inequality is an inequality that is satisfied by any optimal solution os the original (unrelaxed) problem but may be violated by the optimal solution of the relaxation. Valid inequalities improve the quality of the relaxation because they exclude a part of its feasible region that cannot contain the optimal solution of the original problem. Triangle inequalities are among the most frequently used valid inequalities. Two types of these 'triangle inequalities' are considered. The first is the inequality

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

Note that in (21) t_{ij} has replaced $x_i x_j$ and $t_{i,n+1}$ replaces x_i from $F_\phi^\mathcal{B}$. In fact, they consider the homogeneous form $1 + t_{i,n+1} + t_{j,n+1} + t_{i,j} \geq 0$ which is added to (21). One might add all these inequalities, also the ones obtained 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. It can be shown that $1+x_i+x_j+x_i x_j$ cannot be recognized as a SOS based on $M = (1, x_i, x_j)$. However, if we add $x_i x_j$ to the monomial basis we have

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

A similar argument can be given for the three inequalities with x_i and/or x_j replaced by $-x_i$ and/or $-x_j$. Hence, the effect of adding the valid inequality (27) and the three similar inequalities is captured in the SOS approach by adding the monomial $x_i x_j$ to the basis. Below we prove a theorem from which follows that adding the monomial $x_i x_j$ results in upper bounds that are at least as tight as the upper bound of the Feige-Goemans program with the four triangle inequalities of the form (27). The experiments in section 2.4 support this fact.

Feige and Goemans [6] further showed that adding for each triple of variables X_i, X_j and X_k to (21) the valid inequalities

$$\begin{aligned} 1 + t_{ij} + t_{ik} + t_{jk} \geq 0, \quad 1 - t_{ij} + t_{ik} - t_{jk} \geq 0 \\ 1 + t_{ij} - t_{ik} - t_{jk} \geq 0, \quad 1 - t_{ij} - t_{ik} + t_{jk} \geq 0 \end{aligned} \tag{28}$$

improves the tightness of the relaxation. Note that

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

Hence, the effect of adding the four inequalities (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 to the monomial basis results in upper bounds at least as tight compared to adding valid inequalities to the Goemans-Williamson SDP as shown in Theorem 3.

Finally, note that adding all inequalities of the form (28) amounts to adding $\mathcal{O}(n^3)$ inequalities, while in the SOS approach $\mathcal{O}(n^2)$ monomials of degree 2 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.

Theorem 3. *Adding monomials x_ix_j, x_ix_k and x_jx_k to the monomial basis in the SOS approach gives an upper bound at least as tight as the upper bound obtained by adding triangle inequalities of the type (28) to the Goemans-Williamson SDP.*

Proof. Without loss of generality we consider the triangle inequality

$$1 + x_1x_2 - x_1x_3 - x_2x_3 \geq 0 \tag{29}$$

In the notation of (20) this equation is $1 + t_{12} - t_{13} - t_{23} \geq 0$. In matrix notation this inequality is $Tr(AT) \geq 1$ with

$$A = \begin{pmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{pmatrix}$$

We consider the program

$$\begin{aligned} &\min \mathrm{Tr}(M(F)T) \\ &\text{s.t. } \mathrm{diag}(T) = e \\ &\mathrm{Tr}(AT) \geq 1 \\ &T \succeq 0 \end{aligned} \tag{30}$$

Assume that F is an homogenous polynomial of degree 2 in three variables x_1, x_2, x_3 only. This does not harm the general validity of this proof but makes the key steps more transparent. $M(F)$ is the coefficient matrix associated with the polynomial F. Let $F(x_1, x_2, x_3) = 2ax_1x_2 + 2bx_1x_3 + 2cx_2x_3$. Then,

$$M(F) = \begin{pmatrix} 0 & a & b \\ a & 0 & c \\ b & c & 0 \end{pmatrix}$$

The dual program of (30) is the following

$$\begin{aligned} &\max \gamma_1 + \gamma_2 + \gamma_3 + y \\ &\text{s.t. } yA + \mathrm{Diag}(\gamma) + U = M(F) \\ &U \succeq 0, y \geq 0 \end{aligned} \tag{31}$$

with $\mathrm{Diag}(\gamma)$ the 3×3-matrix with on its diagonal $\gamma_1, \gamma_2, \gamma_3$.

Program (31) can be reformulated as

$$\begin{aligned} &\max \gamma_1 + \gamma_2 + \gamma_3 + y \\ &\text{s.t. } M(F) - \mathrm{Diag}(\gamma) - yA \succeq 0 \\ &y \geq 0 \end{aligned} \tag{32}$$

Now suppose that $(\hat{\gamma_1}, \hat{\gamma_2}, \hat{\gamma_3}, \hat{y})$ is an optimal solution for (32). We will show that from this optimal solution a feasible solution for (33) can be constructed with $M = (1, x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3)$. In fact, we will even show that the monomial basis $M_1 = (1, x_1x_2, x_1x_3, x_2x_3)$ is already sufficient in this respect.

$$\begin{aligned} &\max \alpha \\ &\text{s.t. } MSM^T \equiv F - \alpha \quad \text{modulo } \; I_{\mathcal{B}} \end{aligned} \tag{33}$$

Program (33) with monomial basis M_1 can be reformulated as

$$\begin{aligned} &\max \left(-\sum_{i=1}^{4} s_{ii} \right) \\ &s_{12} + s_{21} + s_{34} + s_{43} = 2a \\ &s_{13} + s_{31} + s_{24} + s_{42} = 2b \\ &s_{23} + s_{32} + s_{14} + s_{41} = 2c \\ &S \succeq 0 \end{aligned} \tag{34}$$

$1 + x_1x_2 - x_1x_3 - x_2x_3$ is a SOS modulo $I_{\mathcal{B}}$, because the following holds

$$1 + x_1x_2 - x_1x_3 - x_2x_3 = \frac{1}{4} M_1 \varDelta M_1^T \tag{35}$$

with $\varDelta$ the positive semidefinite matrix

$$\varDelta = \begin{pmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{pmatrix}$$

Let Z be the 4×4 matrix with the 3×3 matrix $M(F) - \text{Diag}(\hat{\gamma}) - \hat{y}A$ starting in the upper left corner and having zeros in fourth row and column. We can conclude that $Z + \frac{1}{2}\hat{y}\varDelta \succeq 0$ because $Z \succeq 0$, $\varDelta \succeq 0$ and $\hat{y} \geq 0$. The matrix $Z + \frac{1}{2}\hat{y}\varDelta$ satisfies the constraints in (34) and $-\sum_{i=1}^{4} s_{ii} = \hat{\gamma_1} + \hat{\gamma_2} + \hat{\gamma_3} + \hat{y}$.

Because the optimal solution of Feige-Goemans SDP with valid inequalities equals a feasible solution of the SDP approach with monomials of degree 2 in the monomial basis, it can be concluded that the SOS approach gives at least as tight upper bounds.

2.3 SOS-Approach on Feige-Goemans Example

In general the CNF formula of $dFGn$ in n variables is defined as

$$\begin{aligned} &x_1 \vee x_2, \; x_2 \vee x_3, \; x_3 \vee x_4, \; \ldots, x_n \vee x_1 \\ &\neg x_1 \vee \neg x_2, \; \neg x_2 \vee \neg x_3, \; \neg x_3 \vee \neg x_4, \; \ldots, \; \neg x_n \vee \neg x_1 \end{aligned} \tag{36}$$

Feige and Goemans [6] present $dFG5$ as worst-known case example with respect to the performance guarantee of their approach.

Note that we can satisfy $2n-1$ of the clauses if n is odd by setting the odd-numbered variables to true and the even-number variables to false. It is not possible to satisfy all clauses for odd n. In this section, we show that the SOS-approach with a monomial basis 1, all variables and all possible monomials of degree 2 solves (36) to optimality.

Theorem 4. *Let n be an odd number. The polynomial $F^{\mathcal{B}}_{\phi_n} - 1$ with*

$$F^{\mathcal{B}}_{\phi_n} = \frac{1}{2}\left(n + x_1x_2 + x_2x_3 + \ldots + x_{n-1}x_n + x_nx_1\right) \tag{37}$$

is a sum of squares if we choose as monomial basis $M = \{1, x_1, \ldots, x_n\}$ extended with the set of all possible monomials of degree 2.

Proof. As initial step we start with $dFG3$. The polynomial $F^{\mathcal{B}}_{\phi_3}$ is

$$F^{\mathcal{B}}_{\phi_3} = \frac{1}{2}\left(3 + x_1x_2 + x_2x_3 + x_3x_1\right) \tag{38}$$

Notice that $F^{\mathcal{B}}_{\phi_3} - 1$ is a sum of squares modulo $I_{\mathcal{B}}$, because

$$\frac{1}{2}\left(\frac{1}{2}\left(1 + x_1x_2 + x_2x_3 + x_3x_1\right)\right)^2 \equiv \frac{1}{2}\left(1 + x_1x_2 + x_2x_3 + x_3x_1\right)$$

We use this fact to prove by induction that $F^{\mathcal{B}}_{\phi_n} - 1$ is also a sum of squares relative to the monomial basis considered. Assume that the polynomial $F^{\mathcal{B}}_{\phi_{n-2}} - 1$ related to $dFG(n-2)$ is a sum of squares modulo $I_{\mathcal{B}}$.

The polynomial $F^{\mathcal{B}}_{\phi_n}$ equals

$$F^{\mathcal{B}}_{\phi_n} = F^{\mathcal{B}}_{\phi_{n-2}} + \frac{1}{2}(2 + x_{n-2}x_{n-1} + x_{n-1}x_n + x_nx_1 - x_{n-2}x_1) \tag{39}$$

We assumed that $F^{\mathcal{B}}_{\phi_{n-2}} - 1$ is a sum of squares. Let $T_1(x) = 1 - x_1x_{n-2} + x_{n-2}x_{n-1} + x_1x_{n-1}$ and $T_2(x) = 1 - x_1x_{n-1} + x_{n-1}x_n + x_nx_1$. Note that $F^{\mathcal{B}}_{\phi_n} = F^{\mathcal{B}}_{\phi_{n-2}} + \frac{1}{2}(T_1(x) + T_2(x))$ and for $i = 1$ and $i = 2$

$$\frac{1}{2}\left(\frac{1}{2}T_i(x)\right)^2 \equiv \frac{1}{2}T_i(x) \text{ modulo } I_{\mathcal{B}}$$

This proves that $F^{\mathcal{B}}_{\phi_n} - 1$ is also a sum of squares.

From this theorem we can conclude that the SOS-approach with monomial basis 1, the variables and all possible monomials of degree 2 identifies $F^{\mathcal{B}}_{\phi_n} - 1$ as a sum of squares. Hence, the minimum of $F^{\mathcal{B}}_{\phi_n}$ is at least 1. We conclude that our SOS approach solves (36) to optimality.

2.4 Experimental Results

In this section we consider besides the Goemans-Williamson upper bound the next four 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 of the type (27) are added.
Variant FG_{ap}**:** For *each* pair of variables the four inequalities of the type (27) are added.
Variant FG_{pt}**:** All inequalities of variant FG_{ap} are added and additionally for each triple of variables the four inequalities of type (28) added.

We compare the upper bounds resulting from these variants with the upper bound obtained from the semidefinite program (13) 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. SOS_{ap} is the variant with monomial basis $\{1, x_1, \ldots, x_n\}$ extended with all $x_i x_j$ for each pair of variables X_i and X_j. We will restrict ourselves to small-scale problems in this section, because solving the SDP's of SOS_p takes a lot of time with the SDP-solvers currently available.

In our initial experiments we used a set of 900 randomly generated instances with 10 variables and different densities. For each of the densities 1.0, 1.5, 2.0, . . ., 5.0, 100 instances are considered. The 'bound ratio' R is defined as the optimal MAXSAT solution divided by the upper bound found. In Table 1 we give for each method the average R over the set of unsatisfiable instances out of the 100 generated instances for each density. In Table 1, the first column indicates the density, the second the upper bounds by SOS_p, the third column gives the results of SOS_{ap}, the fourth gives the Goemans-Williamson upper bound, the fifth gives the upper bound of variant FG_m, the next the upper bound by variant FG_{4p}, then the upper bound of FG_{ap} and the last column gives upper bound of variant FG_{pt}.

From Table 1 we see that the upper bounds obtained by SOS_{ap} are at least as tight as the other ones. This is not only true on average but in fact for each

Table 1. 10 variables, MAX-2-SAT

d	SOS_p	SOS_{ap}	GW	FG_m	FG_{4p}	FG_{ap}	FG_{pt}
1.0	1	1	0.933480	0.984195	0.993151	0.993151	1
1.5	1	1	0.953544	0.989779	0.993869	0.993869	1
2.0	0.99984	1	0.969460	0.994136	0.997043	0.997242	1
2.5	1	1	0.979549	0.996760	0.998317	0.998420	1
3.0	1	1	0.981904	0.996485	0.998044	0.998188	1
3.5	1	1	0.985519	0.997435	0.998904	0.998975	1
4.0	0.999995	1	0.987250	0.997538	0.998873	0.998930	0.999964
4.5	0.999973	0.999973	0.986327	0.997120	0.998692	0.998776	0.999936
5.0	0.999979	0.999979	0.987015	0.997779	0.998764	0.998841	0.999971

Table 2. 25 variables, MAX-2-SAT

d	SOS_p	GW	FG_{ap}
1.5	0.999403	0.955241	0.995686
2.0	0.999607	0.968157	0.997279
2.5	0.999577	0.976133	0.997639
3.0	0.999906	0.979769	0.998238
3.5	0.999691	0.981444	0.997520
4.0	0.999908	0.982812	0.997890
4.5	0.999882	0.983297	0.997721
5.0	0.999989	0.984816	0.998196

individual instance. SOS_p is almost always, except for one instance, at least as good as the best Feige-Goemans variant FG_{pt}. In these experiments with MAX-2-SAT instances with 10 variables, the SDP's of each variant are solved by Sedumi [11]. Table 2 gives the same type of results for instances with 25 variables but only for the upper bounds that are most relevant and computationally not too expensive to obtain. For the instances with 25 variables GW and FG_{ap} are solved by Sedumi. The SDP's of SOS_p are solved by CSDP [3], because this solver is faster, more accurate and uses less memory when solving these SDP's.

2.5 Note on the Complexity of Different Approaches

The complexity of short step semidefinite optimization algorithms (like for example Sedumi) is $\mathcal{O}((2V^2+C)\sqrt{V})$ if V is the size of the semidefinite variable-matrix in the SDP and C the number of constraints. We will compare the computational complexity of the different upper bound variants in this section. Let ϕ be a CNF formula with n variables and m clauses.

For each of the variants considered the V's and C's involved lead to the following complexities

$$CP_{GW} = \mathcal{O}(n^2\sqrt{n}) \tag{40}$$

$$CP_{FG_m} = \mathcal{O}((n^2+m)\sqrt{n}) \tag{41}$$

$$CP_{FG_{4p}} = \mathcal{O}((n^2+m)\sqrt{n}) \tag{42}$$

$$CP_{FG_{ap}} = \mathcal{O}\left(n^2\sqrt{n}\right) \tag{43}$$

$$CP_{FG_{pt}} = \mathcal{O}(n^3\sqrt{n}) \tag{44}$$

$$CP_{SOSp} = \mathcal{O}\left((n+m)^2\sqrt{n+m}\right) \tag{45}$$

$$CP_{SOSap} = \mathcal{O}(n^5) \tag{46}$$

3 Conclusions and Future Research

In this paper, we presented a new approach for computing upper bounds for MAX-SAT. We show theoretically and experimentally that it gives for MAX-2-SAT at least as tight upper bounds as the approaches by Goemans and Williamson [8] and Feige and Goemans [6].

In a next paper, we conclude that a combination of the original rounding procedures of Goemans and Williamson and of Feige and Goemans, which provide the lower bounds they use to obtain the performance ratio guarantee of their methods, with the SOS based upper bound techniques we proposed, leads to polynomial time algorithms for MAX-2-SAT having performance ratio guarantee at least as good, but observably better in particular cases. A similar conclusion can be drawn with respect to the Karloff and Zwick results for MAX 3-SAT.

Future research should mainly concentrate on developing SDP software which is specifically designed for dealing with the type of problems emerging from the SOS approach. They possess a very special structure which could be explored.

References

[1] S.J. Benson and Y. Ye. DSDP5 User guide - The dual-scaling algorithm for semidefinite programming. Technical Report ANL/MCS-TM-255, Argonne National Laboratory, 2005.

[2] G. Blekherman. There are significantly more nonnegative polynomials than sums of squares. submitted to Israel Journal of Mathematics, 2004.

[3] B. Borchers. CSDP : A C library for semidefinite programming. Technical report, New Mexico Tech, 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 theThird Israel Symposium on Theory of Computing and Systems*, pages 182–189, 1995.

[7] K. Fujisawa, M. Kojima, K. Nakata, 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.

[8] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the Assoc. Comput. Mach.*, 42(6):1115–1145, 1995.

[9] H. Karloff and U. Zwick. A 7/8-approximation algorithm for MAX 3SAT? In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science, Miami Beach, FL, USA.* IEEE Press, 1997.

[10] P.A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming Ser. B*, 96(2):293–320, 2003.

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